Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Machine-Learning-Based Forward Modeling of a p-Type Gate-All-Around Junctionless Nanowire Field-Effect Transistor Using Quantum-Corrected Technology Computer-Aided Design

Sushovan Das
Bangladesh University of Engineering and Technology https://orcid.org/0009-0001-0922-4129
Archishman Sarkar
Bangladesh University of Engineering and Technology https://orcid.org/0009-0001-4278-0469
Vivek Chowdhury
Bangladesh University of Engineering and Technology https://orcid.org/0009-0006-6180-8398
Md Mohsinur Rahman Adnan
mmradnan@eee.buet.ac.bd
Bangladesh University of Engineering and Technology https://orcid.org/0000-0002-3333-1514

Research Article

Keywords: gate-all-around, junctionless field-effect transistor, nanowire, neural network, p-type, forward modeling, dilated convolution, technology computer-aided design

Posted Date: November 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11735626/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Machine-Learning-Based Forward Modeling of a p-Type Gate-All-Around Junctionless Nanowire Field-Effect Transistor Using Quantum-Corrected Technology Computer-Aided Design

Sushovan Das[1], *Archishman Sarkar*[1], *Vivek Chowdhury*[1], *and Md Mohsinur Rahman Adnan*[1,*]

[1]Department of EEE, Bangladesh University of Engineering and Technology, Dhaka 1205, Bangladesh
E-mail: mmradnan@eee.buet.ac.bd

Abstract

Quantum confinement becomes significant once a transistor channel is thinner than about 10 nm, so technology computer-aided design (TCAD) models of such devices must include it. Nearly all machine-learning surrogates reported for gate-all-around junctionless field-effect transistors (GAA-JLFETs) cover n-type devices only, although complementary logic also requires p-type devices. In this work, we calibrated a quantum-corrected drift-diffusion TCAD model against measured data from an n-type GAA-JLFET and then adapted it to a p-type device. Six design parameters were sampled by Latin hypercube sampling, and the transfer characteristic of each design was simulated at drain voltages of −0.1 and −1 V. Runs were retained on the basis of simulation reliability rather than device performance, and the noise floor was set from current samples with a physically impossible sign. The resulting 217 devices were used to train a dilated residual one-dimensional convolutional network that predicts the full transfer curve directly from the design parameters. On 26 held-out test devices, the network reached a median R^2 of 0.9994 and a worst-case R^2 of 0.9959, and it predicted the ON current with a mean error of 1.6%. Each prediction requires a single forward pass instead of a TCAD solve, which removes the simulation bottleneck from the design of p-type GAA-JLFETs.

Keywords: gate-all-around; junctionless field-effect transistor; nanowire; neural network; p-type; forward modeling; dilated convolution; technology computer-aided design

1. Introduction

As metal-oxide-semiconductor field-effect transistors (MOSFETs) are scaled down, short-channel effects such as drain-induced barrier lowering (DIBL), threshold-voltage roll-off and subthreshold-slope degradation become stronger and make the device harder to switch off [1]. Below a gate length of about 10 nm, a second problem appears: the source and drain require an abrupt doping profile next to the channel, and forming it requires short, high-temperature anneals that are difficult to integrate into the rest of the process [2]. The industry has answered with a series of new architectures: planar MOSFETs gave way to silicon-on-insulator devices, then to double- and triple-gate FinFETs, and now to gate-all-around (GAA) nanowires and nanosheets [3, 4]. As the generation advances, the channel is more wrapped by the gate. As a result, the gate gets more control ,and it limits how far the drain field reaches. GAA takes this to its limit. In a cylindrical nanowire, a smaller radius directly strengthens gate control, and the subthreshold slope stays close to ideal with low drain leakage even at very short gate lengths.

The abrupt-doping problem is eliminated by the junctinless (JL) transistor in a distinct way. A conventional MOSFET has its source and drain doped opposite to the channel, so a p–n junction forms on each side. A JL device has none. Its source, channel and drain share one doping type and concentration, and it behaves as a gated resistor, in which the gate widens or narrows a depletion region inside a uniformly

doped body [5, 6]. Combining the two ideas gives the GAA-JLFET, which keeps the strong gate control of GAA without needing abrupt junctions and is therefore a strong candidate for sub-10-nm CMOS.

A GAA-JLFET still has to be simulated extensively before it is fabricated. TCAD is the standard tool for this [7, 8]. It solves the carrier-transport and Poisson equations on a mesh of the device structure, which makes it accurate but slow. One sweep takes seconds to minutes. A substantial fraction of runs also fail to converge or return unphysical output, so each usable result costs more than the run time alone suggests [9]. Studies that need thousands of runs, such as design-space exploration, variability analysis or compact-model calibration, are limited by the solver, and switching on quantum-confinement or advanced transport models slows it further.

Machine learning can remove this bottleneck. It takes milliseconds of time to evaluate a trained network, and because it is differentiable, it can sit inside an optimization loop or a circuit simulator without calling TCAD again [10, 11]. Neural networks have already replicated the behavior of GAA and junctionless devices at a fraction of the TCAD cost [12, 13, 14, 9].

The same approach has been applied to other device types, including parameter extraction for standard compact models [15, 16], deep-learning tools for FinFETs and other advanced-node devices [17, 18, 19, 20], variability prediction for ultra-small vertical and GAA devices [21], and general TCAD–AI frameworks [10]. Machine learning has also been used in reverse for FinFETs, returning the design parameters that meet a target performance orders of magnitude faster than manual TCAD iteration [22].

To the best of our knowledge, however, no machine-learning surrogate has been reported for a *p-type* GAA junctionless transistor; nearly every forward-model study of this device family considers n-type devices only [12, 13, 23]. This gap matters for two reasons. First, CMOS logic requires both n- and p-type transistors, so an n-type-only method cannot by itself support a logic gate or a standard-cell library [24]. Second, a p-type JLFET requires a low-work-function gate ($\phi_M \leq 4.1$ eV) rather than the high-work-function gate of an n-type JLFET [2], and holes are transported through silicon differently from electrons. A p-type transfer curve therefore cannot be obtained by mirroring an n-type model, whether by hand or by a network extrapolating outside its training range, and a forward model built specifically for the p-type GAA-JLFET is needed.

This paper presents such a model. We first calibrate a quantum-corrected TCAD model against measured data from a fabricated n-type GAA-JLFET. With the physical models confirmed, we adapt the same model to p-type operation and retain the quantum correction, since the channel is thin enough to require it. We then sweep the main design parameters and simulate the transfer characteristic of each design at two drain voltages. Runs are retained only on the basis of simulation reliability, not device performance, so the training set includes poor designs as well as good ones. These data train a compact neural network that predicts the transfer curve directly from the design parameters.

This work makes four contributions: (i) a quantum-corrected TCAD model of a GAA-JLFET calibrated against measured n-type data; (ii) its extension to a p-type GAA-JLFET with quantum confinement, which, to the best of our knowledge, has not been reported before; (iii) a data-cleaning procedure that judges each run only on simulation reliability and sets the noise floor from the data itself, using current samples with a physically impossible sign; and (iv) a trained network that returns the transfer characteristic of a p-type device from its design parameters in a single forward pass, without a TCAD run.

The rest of the paper is organized as follows. Section 2 describes the device physics, the n-type calibration, the p-type model, the dataset and its cleaning, and the network and its training. Section 3 presents and discusses the results, and Section 4 concludes the paper.

2. Methodology

Figure 1 summarizes the workflow: the quantum-corrected TCAD model, the data cleaning, and the trained forward network.

2.1. Junctionless GAA Nanowire Operation

In a conventional MOSFET, the source and drain are doped opposite to the channel, forming a junction on each side. A JLFET has no such junction: it is a single piece of silicon with the same doping from

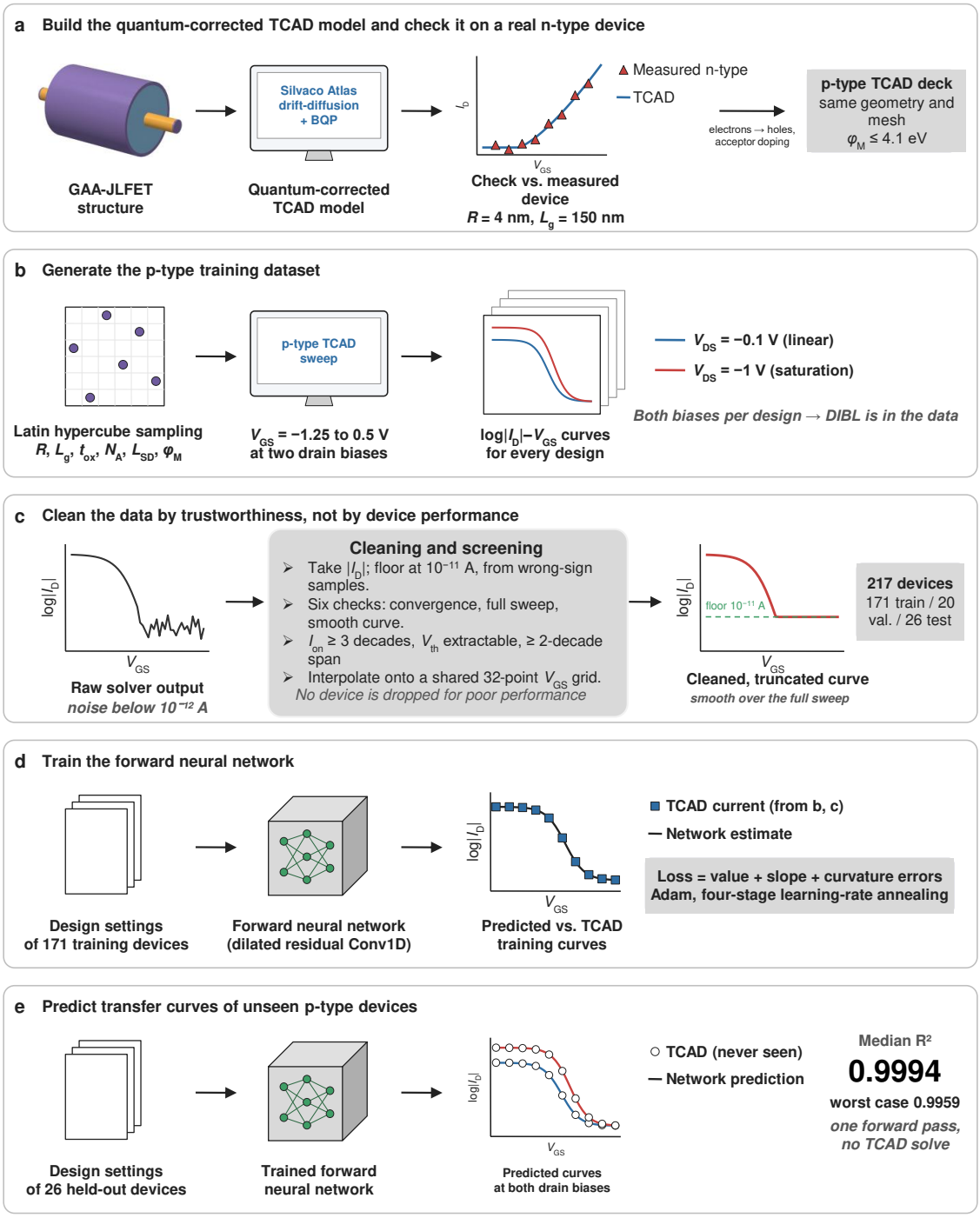


Figure 1: **Overview of the method.** (a) Quantum-corrected technology computer-aided design (TCAD) model of a gate-all-around junctionless field-effect transistor (GAA-JLFET), built in Silvaco Atlas with the Bohm quantum potential (BQP). The model was calibrated against a measured n-type device and then switched to p-type. (b) Latin hypercube sampling of the six design parameters. Each design was simulated at $V_{DS} = -0.1$ V and -1 V. (c) Raw solver output carries numerical noise in the OFF-state (left); cleaning and screening (middle) clamp each curve at the 10^{-11} A noise floor (right). In total, 217 devices passed. (d) A dilated residual convolutional network was trained on 171 of them. (e) The trained network predicted the curves of 26 unseen devices in one forward pass, with a median R^2 of 0.9994.

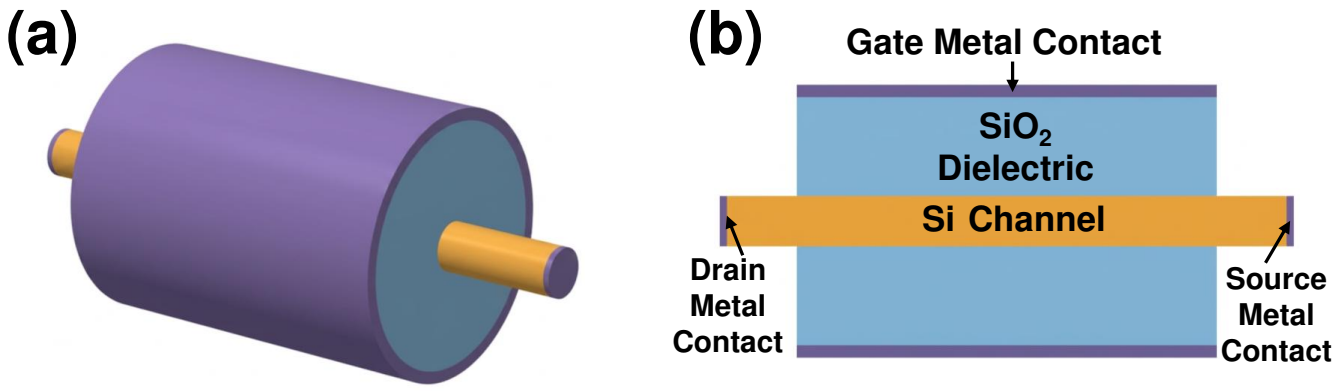


Figure 2: **Gate-all-around junctionless FET structure:** **(a)** three-dimensional view of the GAA-JLFET; **(b)** two-dimensional cross-section.

source to drain [5, 2]. The gate wraps around the nanowire and forms a cylindrical capacitor with the channel (Figure 2a,b). Its work function is chosen so that at $V_{GS} = 0$ V the gate fully depletes the nanowire cross-section [2]. Full depletion at zero bias requires a high-work-function gate for an n-type nanowire ($\phi_M \geq 5.1$ eV) and a low-work-function gate for a p-type nanowire ($\phi_M \leq 4.1$ eV) [2]. The OFF state is thus reached by depleting the uniformly doped channel through the gate; if the channel is not fully depleted, the remaining carriers carry leakage current. Because a JLFET has no source/body or drain/body p–n junction in the current path, conventional junction-engineering techniques such as halo or pocket implants do not apply directly.

The threshold voltage V_{th} of a JLFET is the gate voltage at which the channel becomes fully depleted and the last conducting path disappears. This differs from the conventional MOSFET definition, which refers to the formation of an inversion layer. For a cylindrical nanowire of radius R and doping N_D,

$$V_{th} = V_{FB} - \frac{\pi q N_D R^2}{C_{\text{eff}}} \tag{1}$$

where V_{FB} is the flat-band voltage, q is the elementary charge, N_D is the uniform donor concentration in the channel, R is the nanowire radius, and C_{eff} is the effective gate capacitance per unit length [25].

For a p-type device, N_D is replaced by the acceptor concentration N_A and the sign of the depletion-charge term is reversed. This sets the JLFET apart from a fully depleted MOSFET in two ways. Shrinking R for better gate control also moves V_{th}, because the depletion charge in Equation (1) scales with R^2. The doping is also constrained from both sides: it must be high enough for a low ON-state resistance, yet low enough for the channel to deplete fully at zero bias. As the channel doping is much lower than its source/drain doping, a mosfet doesn't face this trade-off.

Above threshold, the gate voltage shrinks the depletion region, and current flows mainly through the neutral core of the nanowire rather than through a thin surface inversion layer as in a conventional MOSFET [2]. Bulk conduction makes the device much less sensitive to surface roughness and interface charge, which is why junctionless devices often show a higher effective mobility than equally scaled MOSFETs. At higher gate voltage, a surface accumulation layer also begins to conduct; this regime is usually avoided, because it reintroduces interface scattering and makes the transconductance less linear.

2.2. TCAD Model Calibration and p-Type Device Simulation

We first reproduced an n-type GAA-JLFET from the literature in Silvaco Atlas [8]. The reference device [26] is a cylindrical silicon nanowire, 5 nm in radius (R) with a 160-nm gate length (L_g). It is doped n-type at $N_D = 6.7 \times 10^{18}$ cm^{-3}, has a 4-nm oxide (t_{ox}), and uses a p^{++} polysilicon gate with a work function ϕ_M of 4.85 eV. With a diameter near 10 nm, the nanowire confines electrons strongly enough that the

model has to account for quantum effects [2], so we added the Bohm quantum potential (BQP) model to the drift-diffusion equations. BQP is a semi-classical correction. Carriers still move by drift and diffusion, but an extra potential, set by the local carrier density, pushes them away from the Si/oxide interface. This reproduces the qualitative trend of a Schrödinger–Poisson solution without solving the Schrödinger equation. The simulation also included Shockley–Read–Hall recombination, the Lombardi (CVT) mobility model and bandgap narrowing. The nanowire was meshed with radial, axial and angular spacing fine enough to resolve the depletion and neutral regions, and the mesh and bias steps were refined automatically whenever the solver failed to converge.

We compared the simulated transfer characteristics (Figure 3a,b) with measured I_{DS}–V_{GS} data of a 10-nm-diameter silicon nanowire junctionless transistor at drain voltages of 0.1 and 1 V, reported by Singh et al. [26] (their Figure 2). The Atlas structure was built directly from the dimensions and doping reported in that work: nanowire diameter, oxide thickness, channel doping and gate work function. Table 1 compares the figures of merit of the measured and simulated devices. At both biases, the simulated threshold voltage was within 20 mV and the subthreshold swing within 2 mV/dec of the measurement; the ON current was 22–36% below the measured value, and the OFF current was within a factor of four.This n-type comparison is the only connection between the later results and observed hardware because the p-type dataset of Section 2.3 is exclusively derived from simulation.

Table 1: Figures of merit of the TCAD model and the measured device [26] at both drain biases.

FoM	V_{DS} (V)	Experimental	TCAD
V_{th} (V)	0.1	0.390	0.370
SS (mV/dec)	0.1	61.1	60.0
I_{on} (A)	0.1	2.63×10^{-6}	2.06×10^{-6}
I_{off} (A)	0.1	4.97×10^{-13}	2.78×10^{-13}
I_{on}/I_{off}	0.1	5.28×10^{6}	7.40×10^{6}
V_{th} (V)	1.0	0.344	0.357
SS (mV/dec)	1.0	62.2	60.2
I_{on} (A)	1.0	1.03×10^{-5}	6.56×10^{-6}
I_{off} (A)	1.0	1.18×10^{-12}	3.34×10^{-13}
I_{on}/I_{off}	1.0	8.73×10^{6}	1.96×10^{7}

After calibrating the model against n-type data, we adapted it to a p-type device. The p-type device kept the GAA nanowire geometry of the calibrated n-type device, but its channel was doped p-type and its gate was given a low work function ($\phi_M \leq 4.1$ eV, Section 2.1) so that the channel still fully depletes at $V_{GS} = 0$ V. We reused the calibrated model and replaced the electron parameters with hole parameters: hole mobility, hole effective mass and density of states, and acceptor doping. The geometry, transport models, mesh and temperature ($T = 300$ K) do not depend on carrier type and were left unchanged. This is the standard way to adapt a calibrated n-type TCAD deck to p-type, but it has a limitation: the hole mobility and the default BQP hole parameters come from published values for bulk or thin-film silicon, not from a measured p-type GAA-JLFET. The surrogate results in Section 3 therefore show how well the network reproduces a TCAD model calibrated on n-type hardware, not on p-type hardware.

Tables 2 and 3 list the swept design parameters. The gate-length and doping ranges follow the targets of the current technology roadmap for sub-10-nm gate-all-around logic [27], and the gate work function spans the candidate metal-gate electrodes reported for scaled CMOS gate stacks [28, 29]. We built a separate quantum-corrected simulation and network for the p-type JLFET instead of reusing the n-type ones, because a p-type curve cannot be obtained by flipping the sign of the matching n-type curve: the subthreshold slope, the sensitivity of V_{th} to nanowire radius, and the degradation of mobility with vertical field all differ between the two carrier types, even for the same geometry.

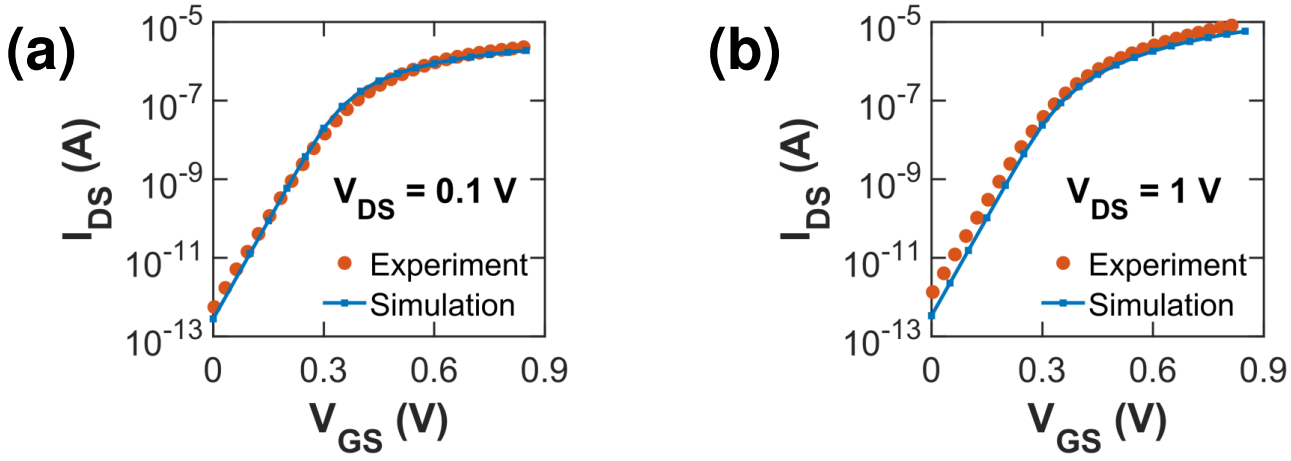


Figure 3: **Calibration of the TCAD model:** quantum-corrected TCAD model compared with the measured n-type device [26]. I_{DS}–V_{GS} characteristics at **(a)** $V_{DS} = 0.1$ V and **(b)** $V_{DS} = 1$ V.

Table 2: Design parameters and their ranges.

Parameter	Range
Channel radius, R	3.5–10 nm
Gate length, L_g	20–80 nm
Oxide thickness, t_{ox}	4–20 nm
Channel doping, N_A	10^{18}–3×10^{19} cm^{-3}
S/D extension length, L_{SD}	10–40 nm
Drain voltage, V_{DS}	-0.1 V, -1 V

2.3. Design Space, Short-Channel Effects, and Simulation Setup

GAA geometry suppresses short-channel effects far more than planar devices or even FinFETs, but DIBL still matters, especially at short gate lengths. DIBL measures how much V_{th} shifts when the drain voltage increases, DIBL $= -(V_{th,\text{high}} - V_{th,\text{low}})/(V_{DS,\text{high}} - V_{DS,\text{low}})$, and reflects how far the drain field reaches into the depletion region near the source and lowers the barrier to conduction.

We sampled the six design parameters in Tables 2 and 3 (nanowire radius R, gate length L_g, oxide thickness t_{ox}, channel doping N_A, source/drain extension length L_{SD} and gate work function ϕ_M) with Latin hypercube sampling [30], which spreads the samples evenly across all six dimensions and needs far fewer runs than a full grid for similar coverage. Fewer runs also means less solver time lost to the runs that fail to converge (Section 2.4). We sampled N_A on a logarithmic scale and the other parameters linearly, and saved the random seed for reproducibility. For each design, V_{GS} was swept from -1.25 to 0.5 V at two drain voltages, $V_{DS} = -0.1$ V and -1 V. The two biases were chosen far apart: at -0.1 V the device stays

Table 3: Gate materials and their work functions.

Gate material	Work function (eV)
Hf	3.9
Zr	4.05
Ta	4.25
Al	4.28
W	4.55
Mo/TiN	4.6

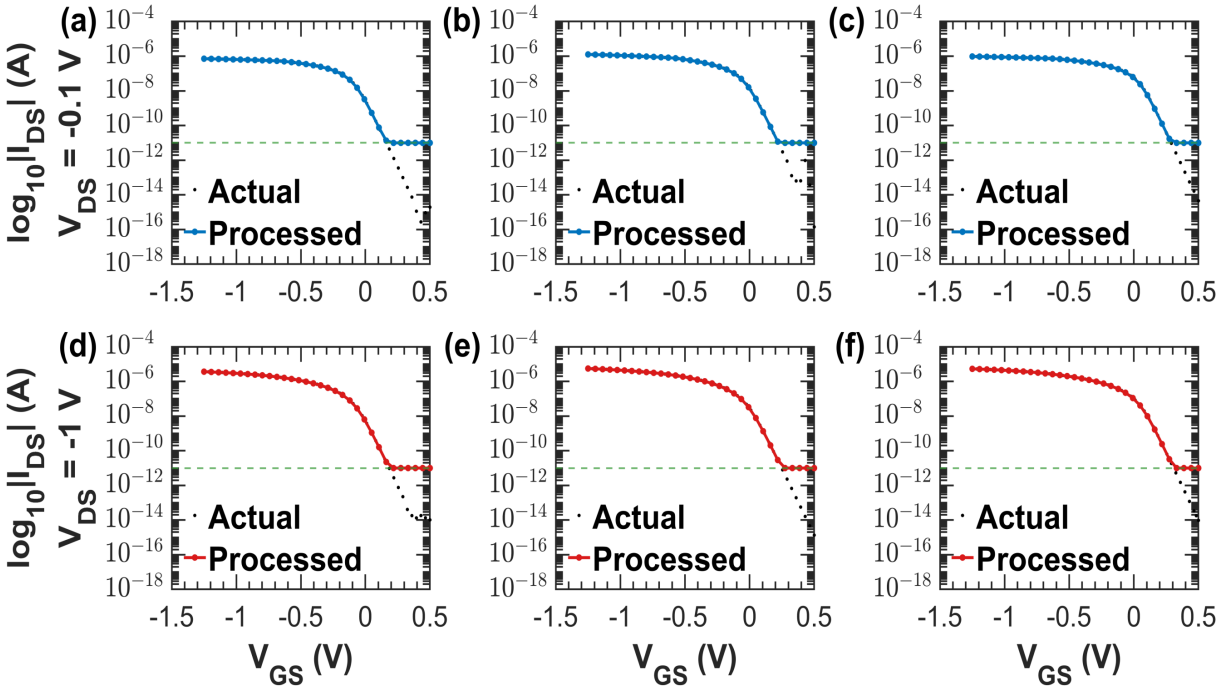


Figure 4: **Data cleaning and the noise floor derived from the data.** $\log_{10}|I_{DS}|$ versus V_{GS} for three sample devices, one per column, at **(a)**–**(c)** $V_{DS} = -0.1$ V and **(d)**–**(f)** $V_{DS} = -1$ V. Black dotted lines are the raw Silvaco output, which falls to 10^{-16}–10^{-18} A deep in the OFF-state because of solver noise. Solid lines are the processed curves, clamped at $I_{\text{floor}} = 10^{-11}$ A (dashed green line, Equation (2)), about 2.2 times the largest wrong-sign sample of 4.47×10^{-12} A. Above the floor, the raw and processed curves are identical.

in the linear region through the ON-state, whereas at -1 V it is in saturation over most of the sweep. The network therefore has to reproduce both curves and the difference between them, which is harder than fitting either curve alone; for this reason, the loss function in Section 2.6 also penalizes errors in the slope of the curve.

2.4. Dataset Cleaning and Feature Engineering

Not every simulated device converged. A device was counted only if both drain-bias curves were available, so a small improvement in single-run convergence gave a much larger gain at the device level. The raw output went through three steps: two cleaning steps, followed by reliability checks that decided which converged runs entered the dataset.

Current sign. A p-channel device biased at negative V_{DS} carries a negative drain current throughout the sweep, so from this point we worked with the magnitude $|I_{DS}|$; once the noise floor below is applied, the sign carries no additional information.

Noise floor from the data. Deep in the OFF-state, the solver returns currents many orders of magnitude below any physical value. These are numerical noise, not device behavior. We did not pick the floor by eye; we took it from the data. I_{DS} must be negative everywhere, so a sample with a positive value cannot be physical, and its magnitude directly measures the solver's own noise. Across both drain biases, the largest such sample had a magnitude of 4.47×10^{-12} A. We set the floor to $I_{\text{floor}} = 10^{-11}$ A, about 2.2 times this bound, and applied it as

$$|I_{DS}| \leftarrow \max\big(|I_{DS}|,\, I_{\text{floor}}\big). \tag{2}$$

With the floor applied, every cleaned curve varied smoothly in $\log_{10}|I_{DS}|$ at both $V_{DS} = -0.1$ V (Figure 4a–c) and $V_{DS} = -1$ V (Figure 4d–f). We clamped the noisy samples instead of setting them to

zero, because the network works on $\log_{10}|I_{DS}|$ and the logarithm of zero is undefined. The same trick carries over to other carrier types and simulators. Any quantity that must keep a known sign gives its own noise estimate through the samples that break that sign.

Reliability screening. Each converged run was judged on how far the simulation could be trusted, whatever the resulting transistor looked like. A device with a poor subthreshold swing, for example, is exactly the kind of variation the model has to learn. Filtering by performance would have removed such devices and left the surrogate without any poor designs to learn from. The screening rules therefore checked only reliability: (i) convergence at both biases; (ii) a complete sweep over the full V_{GS} range; (iii) $\log_{10}|I_{DS}|$ varying smoothly after the floor; (iv) an ON current at least three decades above the noise floor; (v) a threshold voltage extractable by the constant-current method within the sweep; and (vi) a cleaned log-current span of at least two decades. In total, 217 devices passed and were used for the rest of this work. The subthreshold swing, ON/OFF currents, transconductance and threshold voltage of each device were still recorded as metadata, not used as filters, so that they remain available for later analysis.

From the screened devices, we built the features used for training. Because the solver adapts its own voltage steps, the devices did not share a common voltage grid; the raw number of points per device was never below 51 and varied from device to device. We sorted each device by V_{GS} and interpolated it onto a common grid of $N_{V_{GS}} = 32$ points, using linear interpolation for $|I_{DS}|$ and interpolation on a logarithmic scale for $\log_{10}|I_{DS}|$, since linear interpolation across several decades would smear the subthreshold region. The network in Section 2.5 operates on this 32-point grid. Because the transconductance and log-derivative columns returned by the solver are unreliable near the noise floor, we discarded them and recomputed both by central differences on the cleaned grid.

Each device was thus represented by $|I_{DS}|$, $\log_{10}|I_{DS}|$, $\partial|I_{DS}|/\partial V_{GS}$ and $\partial \log_{10}|I_{DS}|/\partial V_{GS}$ at both drain biases. The network learns to predict the first two quantities at each bias. On the other hand, the derivative columns were retained for a future inverse-design model. The design-parameter vector of each device is

$$\mathbf{y} = [R,\, t_{ox},\, L_g,\, \log_{10} N_A,\, \phi_M,\, L_{SD}]^{\top} \tag{3}$$

with the doping on a logarithmic scale, since it spans about one and a half decades in the dataset and a linear scale would compress the low-doping end into a narrow part of the input range. Both the parameter vector and the curves were scaled to $[-1, 1]$ by min–max scaling, matching the tanh output of the network, and the scaling constants were saved with the trained weights so that predictions can be converted back to physical units. The 217 devices were split with a fixed random shuffle into 171 training, 20 validation and 26 test devices. The validation set was used for early stopping and learning-rate scheduling; the test set was not used until Section 3.

2.5. Neural Network Architecture

The network maps the six-element scaled design vector to the scaled output curves on the 32-point V_{GS} grid of Section 2.4 (Figure 5, Table 4). A dense layer expands the input to 1024 units, which are reshaped into a sequence of 32 steps with 32 features each, one step per grid point, so that the following layers can shape the curve along V_{GS}. Earlier work on similar problems has used recurrent networks, but one is not needed here. The dense layer mixes all six inputs before the reshape, so every step of the sequence already carries information about all six parameters. What remains is to shape the curve along the V_{GS} axis.

A pointwise (1×1) convolution then projects the 32-feature sequence onto 64 channels, ahead of the residual stack. It is applied once and is separate from the 1×1 convolution inside each residual block. Table 4 therefore lists the projection as its own row and counts the block-internal convolutions in the residual-block row.

The residual stack uses dilated one-dimensional convolution blocks [31, 32].The threshold transition is a global feature that depends on all six inputs simultaneously, whereas a simple short-filter convolution only detects a few nearby grid points. With dilation rates of $\{1, 2, 4, 8\}$ and filter size $k = 3$, each output position sees $1 + 2(1 + 2 + 4 + 8) = 31$ of the 32 grid points after only four layers. Doubling the dilation at

each layer makes the receptive field grow exponentially with depth instead of linearly. Each block applies a dilated convolution, a GELU activation [33], a 1×1 convolution, a residual (skip) connection and layer normalization [34]. All of them keep a width of 64 filters, so the skip connection adds directly and the stack trains more easily. A final 1×1 convolution with a tanh activation produces four output channels at each of the 32 grid points. makes the stack easier to train. A final 1×1 convolution with a tanh activation produces four output channels at each of the 32 grid points.

The network predicts $|I_{DS}|$ and $\log_{10}|I_{DS}|$ in separate channels and does not derive one from the other. A bounded tanh output cannot hold a quantity spanning many decades with the same accuracy everywhere. The logarithmic output keeps the OFF-state accurate, while the linear output keeps the ON-state accurate, where most of the drive current and practical relevance lie.

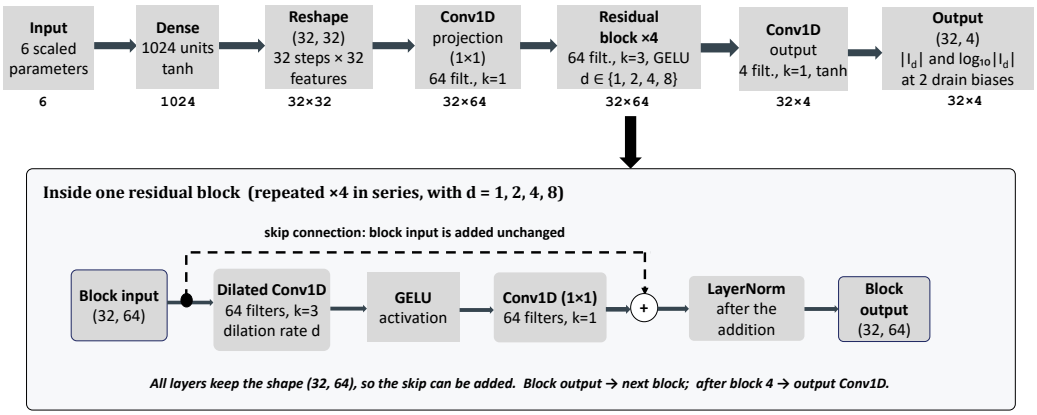


Figure 5: Dilated residual one-dimensional convolutional surrogate for forward prediction of GAA-JLFET transfer characteristics.

Table 4: Architecture of the forward network. $N_{V_{GS}} = 32$ is the number of points on the common gate-voltage grid (Section 2.4). The two Conv1D rows are a one-time channel projection before the residual stack and a pointwise output layer after it; the 1×1 convolution inside each residual block is included in the residual-block row. Total trainable parameters: 76,100.

Layer	Setting	Output shape
Input	scaled design parameters	6
Dense	1024 units, tanh	1024
Reshape	–	$(32, 32)$
Conv1D (channel projection)	64 filters, $k = 1$	$(32, 64)$
Residual block $\times 4$	$k = 3$, $d \in \{1, 2, 4, 8\}$, GELU, 64 filters	$(32, 64)$
Conv1D (output)	4 filters, $k = 1$, tanh	$(32, 4)$

2.6. Loss Function and Training Procedure

A loss that matches current values point by point is not sufficient: two curves can be close in value yet differ in slope or curvature, and the slope sets the transconductance and the sharpness of the subthreshold-to-ON transition. The local slope is also what a future inverse-design use of the network relies on. Following the convention for sparse transistor data [35], we therefore trained with a composite loss that also penalizes errors in the first and second differences along the gate-voltage axis,

$$\mathcal{L} = \mathcal{L}_I + \mathcal{L}_{\Delta I} + \mathcal{L}_{\Delta^2 I} \tag{4}$$

where $\mathcal{L}_I$ is the mean-squared error between the true and predicted scaled curves, and $\mathcal{L}_{\Delta I}$ and $\mathcal{L}_{\Delta^2 I}$ are the mean-squared errors of their first and second differences with respect to gate voltage. The scaled tensor keeps the linear and logarithmic currents in separate channels, so all three loss terms penalize errors in linear and log space at the same time. Additionally, the two drain biases are distinct channels. The network therefore has to reproduce the difference between them, such as the DIBL-induced shift, and cannot fit each bias on its own. We trained with Adam [36], starting at a learning rate of 10^{-3} and lowering it in four stages. Each stage ran until the validation loss (Equation 4) had not improved for 50 epochs. As a result, the best weights were restored, and the learning rate was multiplied by 0.35 for the next stage. The final model used the weights with the lowest validation loss over all four stages, and Table 5 lists the schedule recorded during training.

Table 5: Learning-rate annealing schedule and the best validation loss reached in each of the four stages.

Stage	Learning rate	Epochs	Best validation loss
1	1.00×10^{-3}	466	1.488×10^{-3}
2	3.50×10^{-4}	421	8.69×10^{-4}
3	1.22×10^{-4}	400	7.47×10^{-4}
4	4.29×10^{-5}	125	7.39×10^{-4}

The lowest validation loss, 7.39×10^{-4}, was reached in the final stage (Figure 6). The staged schedule matters in practice, because a single fixed learning rate is either too small to reach a good solution or too large to resolve the fine curvature that the $\mathcal{L}_{\Delta^2 I}$ term targets once the overall curve shape has been learned. All weights started from the Glorot uniform initialization [37], the default for the dense and convolutional layers used here.

The model was selected on validation loss alone; no held-out accuracy score entered that choice. We then reported the accuracy of the selected model with the per-device coefficient of determination,

$$R^2 = 1 - \frac{\sum_i (y_i - \widehat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{5}$$

averaged over both current channels and both drain biases of each device. R^2 was computed only on the 26 held-out test devices in Section 3 ,and it had no contribution on selecting the weights.The network was trained in TensorFlow/Keras [38] with all random seeds fixed and deterministic operations enabled for reproducibility; nevertheless, fixing the seeds alone is insufficient because many default kernels choose non-deterministic algorithms for performance.

3. Results and Discussion

3.1. Accuracy on Held-Out Devices

We measured accuracy by comparing the TCAD and predicted curves with the R^2 score of Equation 5. A single R^2 across several curves is not well defined, so we computed one score for each output (linear and log current at both drain biases) and averaged the four scores for each device. Every quantity was converted back to physical units first, which means the score is in real amperes and not on the internal $[-1, 1]$ scale. Because the linear and log channels are averaged, a device scores well only if both its ON-state and its subthreshold region are right. Getting one right at the expense of the other lowers the score.

The 26 held-out test devices were never used for training or model selection. On these, the network reached a median R^2 of 0.9994 (Figure 7b), a 5th-percentile value of 0.9979 and a worst-case value of 0.9959 (Figure 7a), as listed in Table 6. The median and the 5th percentile differed by only 0.0015, and even the worst device exceeded $R^2 = 0.99$. For the best-case device (Figure 7c), the prediction was almost indistinguishable from TCAD at both drain biases. There is no long low tail, which suggests the network learned trends across the whole design space and did not memorize a narrow part of it. Testing on design points farther outside the sampled ranges would be needed to confirm this.

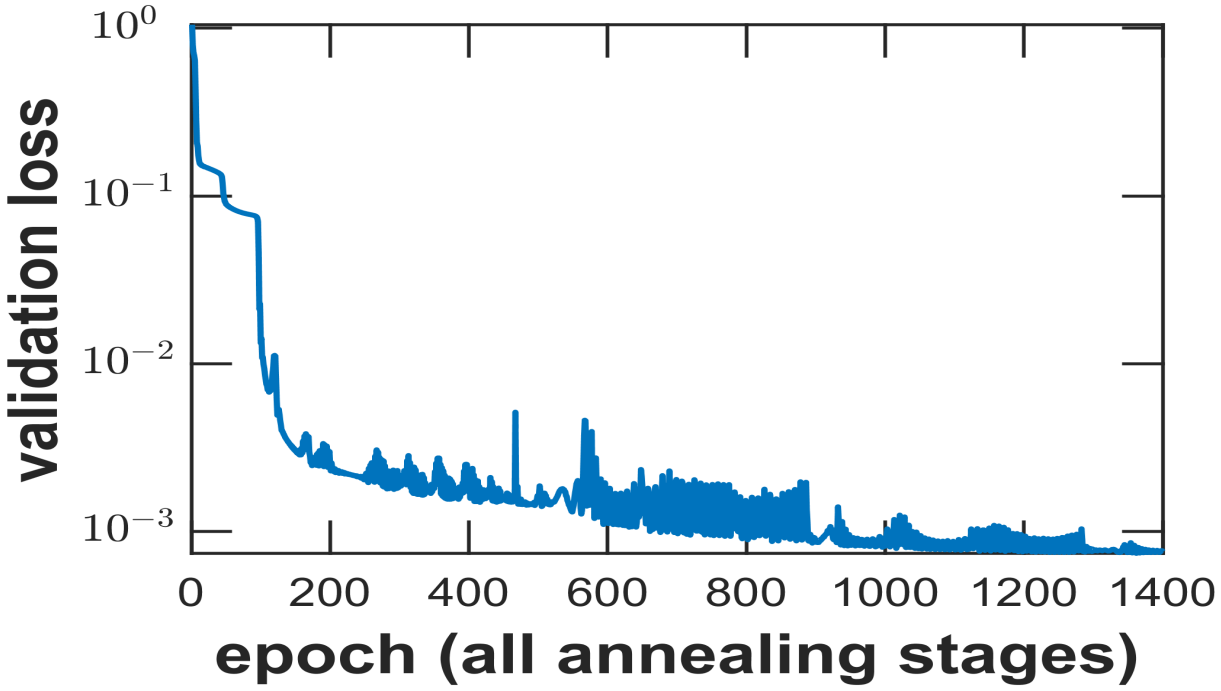


Figure 6: **Validation loss during training.** Validation loss (Equation (4)) versus epoch over the four learning-rate annealing stages (Table 5), which ended at epochs 466, 887, 1287 and 1412. The learning rate was reduced by a factor of 0.35 at the start of each new stage. The lowest loss, 7.39×10^{-4}, was reached in stage 4, and these weights were used in the final model.

Table 6: Accuracy of the forward network on the held-out test set.

Quantity	Value
Decoder type	dilated Conv1D
Trainable weights	76,100
Test devices	26
Median R^2	0.9994
5th-percentile R^2	0.9979
Worst-case R^2	0.9959

Figure 7 shows where the remaining error lies. In the worst device (run 120, panel a), the clearest gap appeared in the steep subthreshold transition at $V_{DS} = -1$ V, where the predicted current fell off at a slightly less positive V_{GS} than the TCAD current, which is equivalent to a small shift in V_{th}. Though the noise floor matched closely, smaller gaps were visible near the ON-state knee at both biases. The predicted and TCAD curves overlapped fully for the median and best case devices (Figure 7b and 7c). As a result, the residual error is concentrated where the curve is steepest, which is consistent with V_{th} and $g_{m,max}$ exhibiting more errors than I_{on} (Section 3.2).

3.2. Comparison of Predicted and TCAD Figures of Merit

A high average R^2 can still hide errors in individual figures of merit. The constant-current V_{th} depends on the curve in one narrow window, for instance. If the network's errors cluster there instead of spreading across the sweep, V_{th} can shift slightly while the average R^2 stays high. Table 7 compares the figures of merit extracted from the predicted and TCAD curves.

I_{on} has a mean error of 1.6% and $R^2 = 0.99$ across the test set ,and became the highest accurate figure of merit. It is read from the linear-current output at the point of largest current, where the curve

Table 7: Figures of merit extracted from the predicted and TCAD curves of the 26 held-out test devices at both drain biases.

FoM	Mean % error	Median % error	R^2
I_{on}	1.6	1.2	0.99
V_{th}	29.0 (7–9 mV)	4.5	0.99
$g_{m,max}$	5.3	4.1	0.93

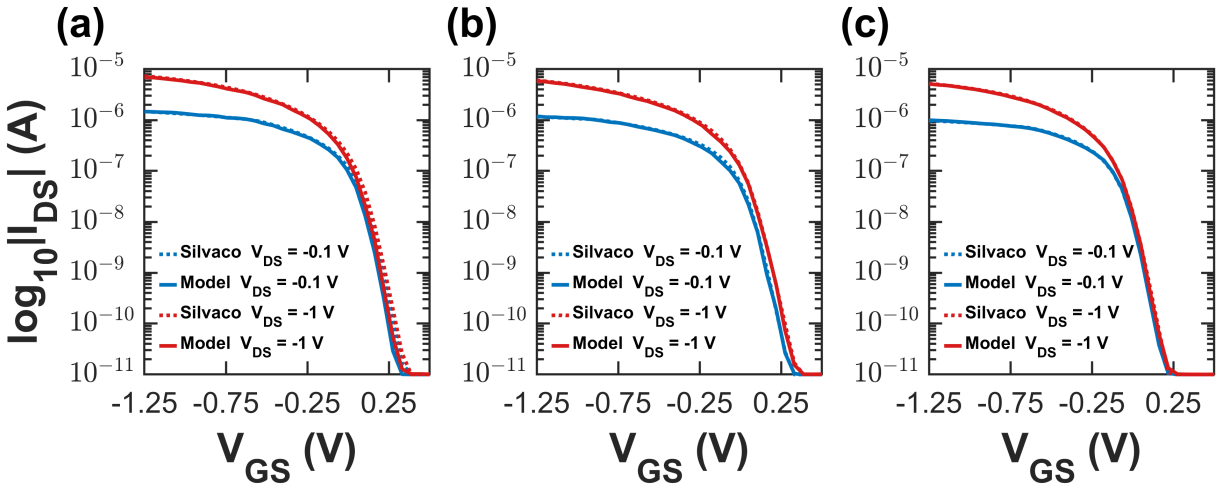


Figure 7: **Predicted versus technology computer-aided design (TCAD) transfer curves for held-out test devices.** $\log_{10}|I_{DS}|$ versus V_{GS} for the **(a)** worst-case ($R^2 = 0.9959$), **(b)** median ($R^2 = 0.9994$) and **(c)** best-case test devices. Dotted lines are Silvaco TCAD; solid lines are the network prediction. Blue is $V_{DS} = -0.1$ V and red is $V_{DS} = -1$ V. In (a), most of the mismatch lies near threshold at $V_{DS} = -1$ V.

is nearly flat with respect to V_{GS}, so small deviations in the predicted curve change I_{on} very little. V_{th} had a mean absolute error of 7–9 mV, well below the 56-mV spacing of the V_{GS} grid, and $R^2 = 0.99$. Its mean percentage error (29.0%) appears large only because several test devices have V_{th} close to 0 V, where the same small absolute shift becomes a large fraction of the reference value; the median percentage error (4.5%) is more representative. $g_{m,max}$, being a derivative of the predicted curve, is more sensitive to local prediction noise than I_{on}, and it showed both a higher mean error (5.3%) and a lower R^2 (0.93).

The surrogate model is not only better in terms of accuracy but also computationally faster and cheaper than the TCAD solver.A prediction is just a single forward pass after training rather than a meshed drift-diffusion solve. Design-space exploration and compact-model calibration are therefore no longer limited by simulation time.

4. Conclusion

We developed a quantum-corrected TCAD model of a p-type GAA-JLFET, calibrated against a fabricated n-type device, and trained a compact forward neural network on the cleaned I_{DS}–V_{GS} dataset to predict the transfer characteristic directly from six design parameters. On 26 held-out devices, the network reached a median R^2 of 0.9994 and a worst-case R^2 of 0.9959, and it predicted I_{on} with a mean error of 1.6% and V_{th} with a mean absolute error of 7–9 mV. Two preprocessing steps were most important: a noise floor derived from current samples with a physically impossible sign, and logarithmic interpolation onto a common voltage grid. Selecting data by simulation reliability rather than device performance was equally important. Each prediction takes a single forward pass instead of a TCAD solve, which removes the slow trial-and-error step from the design of p-type GAA-JLFETs. Because the derivative columns are retained in the dataset (Section 2.4), the model can serve as the forward step of a future inverse-design tool that

starts from a target figure of merit, such as a desired subthreshold swing or threshold voltage, and returns the design parameters that achieve it. A further step is to measure a fabricated p-type device, so that the carrier-type adaptation of Section 2.2 can be verified directly rather than assumed.

Acknowledgements

The authors thank the Bangladesh University of Engineering and Technology (BUET) for providing the necessary resources.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] J.-P. Colinge. *FinFETs and Other Multi-gate Transistors*. Jan. 2008. ISBN: 978-0-387-71751-7. DOI: 10.1007/978-0-387-71752-4.

[2] S. Sahay and M. J. Kumar. *Junctionless Field-Effect Transistors: Design, Modeling and Simulation Shubham Sahay and Mamidala Jagadesh Kumar*. Feb. 2019. ISBN: 978-1-119-52353-6. DOI: 10.1002/9781119523543.

[3] N. Loubet, T. Hook, P. Montanini, C.-W. Yeung, S. Kanakasabapathy, M. Guillom, T. Yamashita, J. Zhang, X. Miao, J. Wang, et al. "Stacked Nanosheet Gate-All-Around Transistor to Enable Scaling Beyond FinFET". In: *Proc. Symp. VLSI Technol.* June 2017, T230–T231. DOI: 10.23919/VLSIT.2017.7998183.

[4] I. Ferain, C. A. Colinge, and J.-P. Colinge. "Multigate Transistors as the Future of Classical Metal–Oxide–Semiconductor Field-Effect Transistors". In: *Nature* 479.7373 (2011), pp. 310–316. DOI: 10.1038/nature10676.

[5] J.-P. Colinge, C.-W. Lee, A. Afzalian, N. D. Akhavan, R. Yan, I. Ferain, P. Razavi, B. O'Neill, A. Blake, M. White, A.-M. Kelleher, B. McCarthy, and R. Murphy. "Nanowire Transistors Without Junctions". In: *Nature Nanotechnology* 5.3 (Mar. 2010), pp. 225–229. DOI: 10.1038/nnano.2010.15.

[6] C.-W. Lee, A. Afzalian, N. D. Akhavan, R. Yan, I. Ferain, and J.-P. Colinge. "Junctionless Multigate Field-Effect Transistor". In: *Applied Physics Letters* 94.5 (2009), p. 053511. DOI: 10.1063/1.3079411.

[7] S. Selberherr. *Analysis and Simulation of Semiconductor Devices*. Vienna, Austria: Springer-Verlag, 1984. DOI: 10.1007/978-3-7091-8752-4.

[8] *ATLAS User's Manual: Device Simulation Software*. Silvaco, Inc. Santa Clara, CA, USA, 2020.

[9] S.-C. Han, J. Choi, and S.-M. Hong. "Acceleration of Semiconductor Device Simulation With Approximate Solutions Predicted by Trained Neural Networks". In: *IEEE Transactions on Electron Devices* 68 (May 2021), pp. 5483–5489. DOI: 10.1109/TED.2021.3075192.

[10] C. Jeong, S. Myung, I. Huh, B. Choi, J. Kim, H. Jang, H. Lee, D. Park, K. Lee, W. Jang, J. Lee, H. Ryu, and D. S. Kim. "Bridging TCAD and AI: Its Application to Semiconductor Design". In: *IEEE Transactions on Electron Devices* 68.11 (2021), pp. 5364–5371. DOI: 10.1109/TED.2021.3093844.

[11] F. Klemme, J. Prinz, V. van Santen, J. Henkel, and H. Amrouch. "Modeling emerging technologies using machine learning: challenges and opportunities". In: Nov. 2020, pp. 1–9. DOI: 10.1145/3400302.3415770.

[12] A. Raj and R. Sharma. "Device Parameter Prediction for GAA Junctionless Nanowire FET Using ANN Approach". In: *Microelectronics Journal* 147 (May 2024), p. 106192. DOI: 10.1016/j.mejo.2024.106192.

[13] A. Ouchen, T. Berghout, F. Djeffal, and M. Ferhati. "Machine Learning-Guided Design of 10-nm Junctionless Gate-All-Around MOSFETs for Nanoscaled Digital Circuits". In: *Physica Status Solidi A* 222.6 (Mar. 2025), p. 2400670. DOI: 10.1002/pssa.202400670.

[14] H. Wong, M. Xiao, J. Wang, J. Zhang, and M. Chan. "TCAD-Machine Learning Framework for Device Variation and Operating Temperature Analysis with Experimental Demonstration". In: *IEEE Journal of the Electron Devices Society* 8 (2020), pp. 992–1000. DOI: 10.1109/JEDS.2020.3024669.

[15] A. Singhal, G. Pahwa, and H. Agarwal. "A Novel Physics-Aware ANN-Based Framework for BSIM-CMG Model Parameter Extraction". In: *IEEE Transactions on Electron Devices* 71.6 (2024), pp. 3307–3314. DOI: 10.1109/TED.2024.3381917.

[16] A. Ashai, A. Jadhav, A. K. Behera, S. Roy, A. Dasgupta, and B. Sarkar. "Deep Learning-Based Fast BSIM-CMG Parameter Extraction for General Input Dataset". In: *IEEE Transactions on Electron Devices* 70.7 (2023), pp. 3437–3441. DOI: 10.1109/TED.2023.3278615.

[17] M.-Y. Kao, F. Chavez, S. Khandelwal, and C. Hu. "Deep Learning-Based BSIM-CMG Parameter Extraction for 10-nm FinFET". In: *IEEE Transactions on Electron Devices* 69.9 (2022), pp. 4765–4768. DOI: 10.1109/TED.2022.3181536.

[18] J.-H. Chen, F. Chavez, C.-T. Tung, S. Khandelwal, and C. Hu. "A Single Neural Network Global I–V and C–V Parameter Extractor for BSIM-CMG Compact Model". In: *Solid-State Electronics* 216 (2024), p. 108898. DOI: 10.1016/j.sse.2024.108898.

[19] F. Chavez and S. Khandelwal. "Deep Learning-Based ASM-HEMT High Frequency Parameter Extraction". In: *Proc. IEEE Wireless Microw. Technol. Conf. (WAMICON)*. 2023, pp. 41–44. DOI: 10.1109/WAMICON57636.2023.10124884.

[20] S. Woo, H. Jeong, J. Choi, H. Cho, J.-T. Kong, and S. Kim. "Machine-Learning-Based Compact Modeling for Sub-3-nm-Node Emerging Transistors". In: *Electronics* 11.17 (2022), p. 2761. DOI: 10.3390/electronics11172761.

[21] K. Ko, J. Lee, M. Kang, J. Jeon, and H. Shin. "Prediction of Process Variation Effect for Ultrascaled GAA Vertical FET Devices Using a Machine Learning Approach". In: *IEEE Transactions on Electron Devices* 66 (Sept. 2019), pp. 4474–4477. DOI: 10.1109/TED.2019.2937786.

[22] K. Kim et al. "Simulator Acceleration and Inverse Design of Fin Field-Effect Transistors Using Machine Learning". In: *Scientific Reports* 12 (2022), p. 1140. DOI: 10.1038/s41598-022-05111-3.

[23] A. Raj, S. Kumar, and R. Sharma. "Machine Learning Based Prediction of I–V and Transconductance Curves for 3-D Multichannel Junctionless FinFET". In: *Indian Journal of Physics* 98 (Dec. 2024), pp. 4515–4523. DOI: 10.1007/s12648-024-03179-3.

[24] H. Carrillo-Núñez, N. Dimitrova, A. Asenov, and V. Georgiev. "Machine Learning Approach for Predicting the Effect of Statistical Variability in Si Junctionless Nanowire Transistors". In: *IEEE Electron Device Letters* 40.9 (Sept. 2019), pp. 1366–1369. DOI: 10.1109/LED.2019.2931839.

[25] E. Gnani, A. Gnudi, S. Reggiani, and G. Baccarani. "Theory of the Junctionless Nanowire FET". In: *IEEE Transactions on Electron Devices* 58.9 (2011), pp. 2903–2910. DOI: 10.1109/TED.2011.2159608.

[26] P. Singh, N. Singh, J. Miao, W.-T. Park, and D.-L. Kwong. "Gate-All-Around Junctionless Nanowire MOSFET With Improved Low-Frequency Noise Behavior". In: *IEEE Electron Device Letters* 32.12 (2011), pp. 1752–1754. DOI: 10.1109/LED.2011.2169645.

[27] IEEE International Roadmap for Devices and Systems. *IRDS 2022 Edition: More Moore*. 2022. URL: https://irds.ieee.org/editions/2022.

[28] H. B. Michaelson. "The Work Function of the Elements and Its Periodicity". In: *Journal of Applied Physics* 48.11 (1977), pp. 4729–4733. DOI: 10.1063/1.323539.

[29] Y.-C. Yeo, T.-J. King, and C. Hu. "Metal-Dielectric Band Alignment and Its Implications for Metal Gate Complementary Metal-Oxide-Semiconductor Technology". In: *Journal of Applied Physics* 92.12 (2002), pp. 7266–7271. DOI: 10.1063/1.1521517.

[30] M. D. McKay, R. J. Beckman, and W. J. Conover. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code". In: *Technometrics* 21.2 (1979), pp. 239–245. DOI: 10.1080/00401706.1979.10489755.

[31] F. Yu and V. Koltun. "Multi-Scale Context Aggregation by Dilated Convolutions". In: *Proc. Int. Conf. Learn. Represent. (ICLR)*. 2016. DOI: 10.48550/arXiv.1511.07122.

[32] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[33] D. Hendrycks and K. Gimpel. "Gaussian Error Linear Units (GELUs)". In: *arXiv:1606.08415* (2016). DOI: 10.48550/arXiv.1606.08415.

[34] J. L. Ba, J. R. Kiros, and G. E. Hinton. "Layer Normalization". In: *arXiv:1607.06450* (2016). DOI: 10.48550/arXiv.1607.06450.

[35] R. K. A. Bennett, H. F. Gault, A. I. Khan, L. Hoang, T. Peña, K. Neilson, Y. S. Song, Z. Zhang, A. J. Mannix, and E. Pop. "Deep Learning to Automate Fitting and Parameter Extraction of 2D Transistors". In: *Proc. Silicon Nanoelectronics Workshop (SNW)*. 2025, pp. 122–123. DOI: 10.23919/SNW65111.2025.11097275.

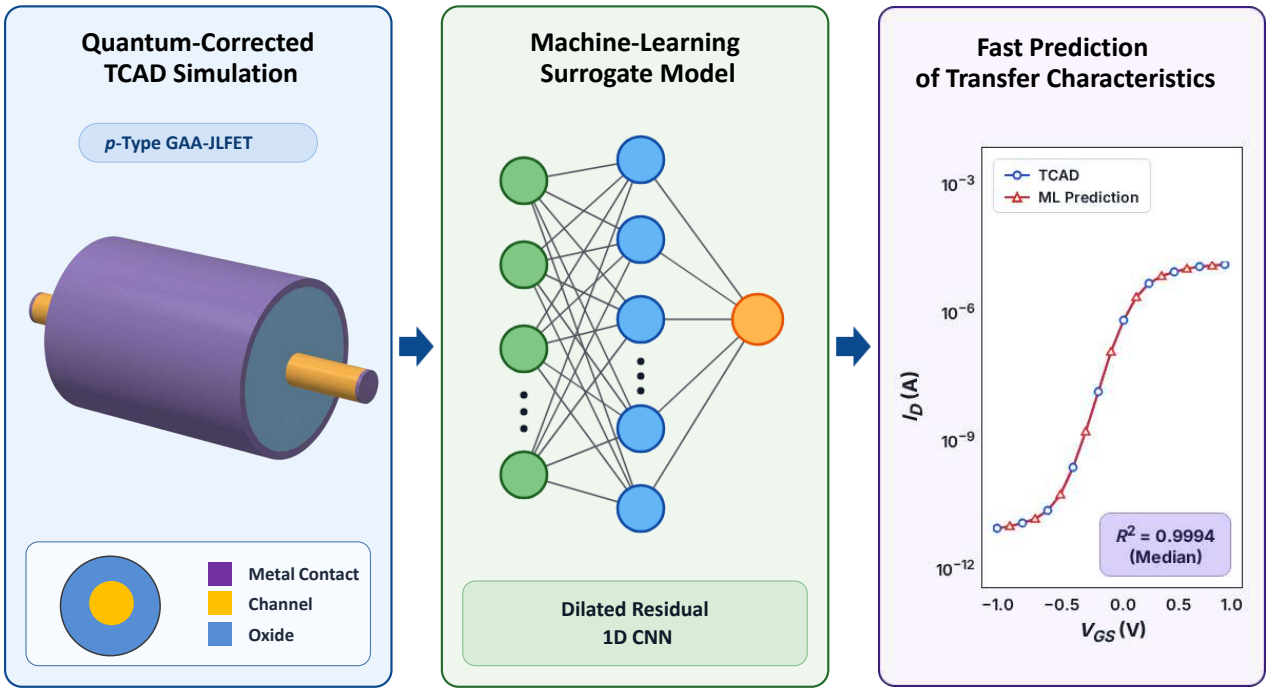
[36] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *Proc. Int. Conf. Learn. Represent. (ICLR)*. 2015. DOI: 10.48550/arXiv.1412.6980.

[37] X. Glorot and Y. Bengio. "Understanding the Difficulty of Training Deep Feedforward Neural Networks". In: *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*. 2010, pp. 249–256.

[38] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. URL: https://www.tensorflow.org/.

caption

Table of Contents



A machine-learning model predicts how a nanoscale p-type transistor behaves directly from its design, replacing slow physics-based simulation. Trained on gate-all-around junctionless transistor data that accounts for quantum effects, it reproduces current curves accurately across untested designs.