Machine Learning-Assisted Device Circuit Co-Optimization: A Case Study on Inverter

Liyuan Xue, *Member, IEEE*, Ankit Dixit, *Member, IEEE*, Naveen Kumar, *Member, IEEE*, Vihar Georgiev, *Senior Member, IEEE*, and Bo Liu, *Senior Member, IEEE*

***Abstract*— This study demonstrates a machine learning (ML)-assisted device circuit co-optimization technique. A basic CMOS inverter cell is employed to demonstrate this proof of concept. The voltage transfer and the switching characteristics are examined to observe the ON- and OFF-state behavior while applying two short square pulses of 140 ns and 100 ps, respectively. By applying the proposed ML-based optimization method using the actor and critic neural networks, the mixed-mode simulation [i.e., technology computer-aided design (TCAD) and SPICE] outputs toward the desired behavior of the circuit, and parameters, including area factor, doping concentration, capacitance value, and width and length of the device, are optimized. Compared with the manual design case, the co-optimized design surpasses several figures of merit (FoMs), such as propagation delay and overshoot, which pave the way for future research on more complex circuit design challenges.**

***Index Terms*— Actor–critic-based optimization, CMOS inverter, machine learning (ML), mixed-mode simulations, technology computer-aided design (TCAD).**

I. Introduction

In THE current era of semiconductor technology, numerous advancements are taking place at the device level, including the development of FinFET, nanosheet [1], and gate-all-around transistors [2]. These devices empower circuit engineers to design intricate circuits, such as memory and processors, with higher efficiency in terms of reduced power consumption and enhanced processing speed. However, the performance of these circuits relies not only on the precise adjustment of external passive components, such as resistors and capacitors [3], but also, predominantly, on the physical characteristics of the device. These characteristics include the width and length of the device and also the process and material of the device, such as the doping concentration in the channel region of CMOS. The manual tuning procedure for determining the accurate values of these parameters is a challenging task involving substantial computing cost on both circuit and technology simulation, which greatly hinders obtaining the optimal performance [4], [5], [6].

Because of the rapid development of machine learning (ML) and artificial intelligence (AI) [7], many recent studies have demonstrated that methods based on ML can greatly help in many fields of semiconductor industry, including but not limited to device modeling [8], [9], [10], [11], [12], [13], [14], device simulation [15], [16], process variation prediction [17], [18], [19], [20], [21], failure troubleshooting [22], and device designing [23], [24], [25], [26], [27], [28]. These methods successfully contribute to reducing simulation time through a pretrained model of electrical characteristics (i.e., I–V relations) for downstream tasks, while effectively obtaining a better design. However, to capture subtle variation of the signal in downstream tasks, the models (either the ML-based model or the conventional compact model) have to be highly accurate, requiring thousands of data, either by simulation or measurement, to be collected for modeling case by case. This makes the modeling phase time-consuming and extends the time to market of the device. Moreover, ML technique in the reported literature only plays an assisting role in helping engineers with performance evaluation, while manually tuning the parameter in circuit level is still vital in the whole design process. Therefore, suboptimal designs are often obtained.

In this article, we present a novel design approach that applies the ML technique to co-optimize the device and circuit parameters simultaneously. Driven by directly modeling performance at the circuit level, the optimization algorithm helps design the device and circuit in a integrated and straightforward way. To the best of our knowledge, this work is the first to utilize the ML technique directly for the entire design workflow of the device and circuit simultaneously.

We used a CMOS inverter as a proof of concept. Since the inverter is the fundamental building block in almost any digital circuit, the appropriate design of inverters greatly facilitates the development of more intricate structures, such as NAND gates,

Received 13 July 2024; revised 17 September 2024; accepted 1 October 2024. This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/V048341/1 and Grant EP/X5257161/1. The review of this article was arranged by Editor N. Xu. *(Corresponding authors: Ankit Dixit; Liyuan Xue.)*

Liyuan Xue and Bo Liu are with the AI-Driven Design Group, James Watt School of Engineering, University of Glasgow, G12 8LT Glasgow, U.K. (e-mail: l.xue.1@research.gla.ac.uk; bo.liu@glasgow.ac.uk).

Ankit Dixit, Naveen Kumar, and Vihar Georgiev are with the DeepNano Group, James Watt School of Engineering, University of Glasgow, G12 8LT Glasgow, U.K. (e-mail: ankit.dixit@glasgow.ac.uk; naveen.kumar@glasgow.ac.uk; vihar.georgiev@glasgow.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TED.2024.3476231.

Digital Object Identifier 10.1109/TED.2024.3476231

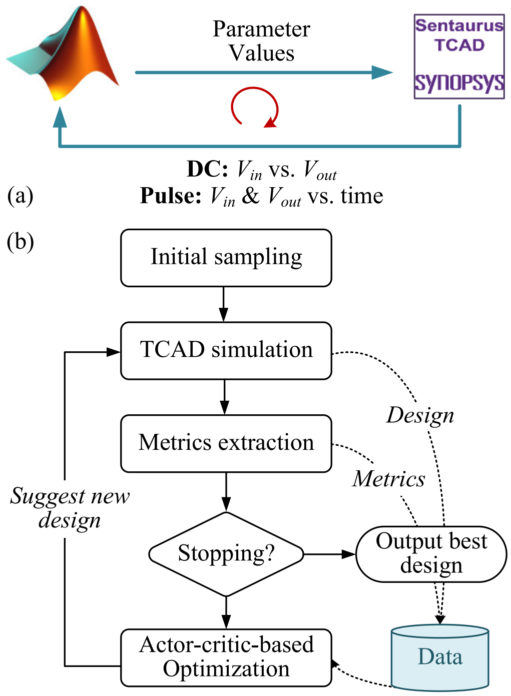

Fig. 1. Dataflow and workflow of the proposed design methodology. (a) Dataflow between programming and simulation. (b) Workflow of the design methodology.

adders, multipliers, and microprocessors, making the process substantially more efficient. By applying a quasi-reinforcement learning algorithm, the inverter was synthesized without pre-defined or nominal configurations. Five types of parameters were considered as the design variables, including gate length, channel width, channel doping concentration for both nMOS and pMOS, area factor, and load capacitance. The goal is to synthesize an inverter that functionally works well in a given circuit topology and achieves optimal switching performance. The resulting performance demonstrates that, by incorporating the ML-based optimization technique, better performance can be obtained more efficiently within acceptable computing cost, which opens up new possibilities for designing complex structures. Note that this work is a proof of concept of the feasibility of ML-based optimization techniques for device circuit co-optimization. Promising applications, including FinFET, gate-all-around (GAA) FET, and nanosheet architecture, can be considered in future works with a refined approach.

II. Co-Optimization Methodology

In traditional design workflow, industrial compact models of devices are constructed based on experimental results or thousands of calibrated technology computer-aided design (TCAD) simulations [9]. Subsequently, engineers use compact models to carry out circuit-level design, where only the width-to-length ratio of devices can be adjusted, known as transistor sizing. Note that circuit performance is related to both circuit and device parameters, but due to an implicit correlation of the performance with device characteristics, manually tuning these parameters in the current workflow is impracticable.

In our methodology, the device and circuit are simulated in a mixed-mode approach (i.e., sequential TCAD and SPICE simulation), while no explicit compact models are constructed. The correlation between device and circuit parameters and performance is learned consecutively by an ML model during optimization. Meanwhile, an algorithm, replacing the manual manipulation of the device parameters (the designer) in the traditional design process, suggests the next candidate design based on the learned pattern directly.

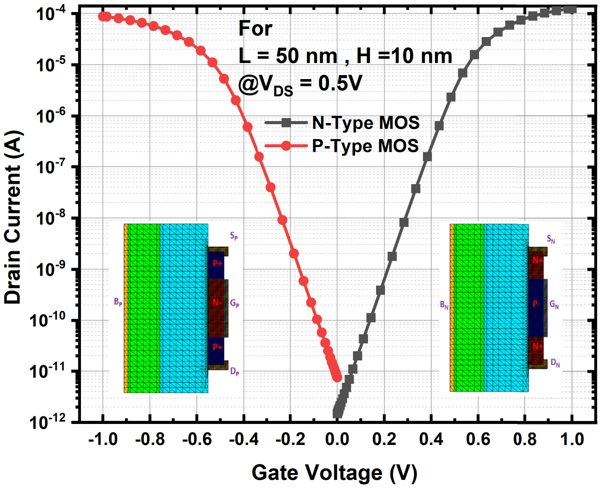

Fig. 2. Two dimension structure of the N/P MOSFET and respective transfer characteristics of the device.

The dataflow and workflow of our methodology are illustrated in Fig. 1(a) and (b). In terms of dataflow, the actor–critic-based optimization (i.e., optimizer) is implemented in MATLAB, while the circuit and device simulation models are constructed in Sentaurus (i.e., simulator). For the latter, mixed-mode simulation of the circuit and device is performed, including TCAD simulation of the device. In each iteration, the simulator provides the performance of a candidate design, and the optimizer provides a new design for simulation. This happens iteratively until the optimal design is obtained.

In terms of the workflow in Fig. 1(b), it commences with several steps. First, a small set of initial sampling for design variables within the predefined search bounds is conducted. For each sample, the circuit performance metrics are extracted from its raw characteristics data. All simulated designs and performance metrics are stored as design–metrics pairs in dataset. Then, the actor–critic-based optimization algorithm is applied to learn the model and suggest the next design with good potential for simulation for the next iteration. Note that only a single new design is suggested for the mixed-mode simulation in each iteration. The iterative process stops when the convergence criteria are satisfied, such as a satisfactory design is obtained, or the computing budget (the maximum iteration) is exhausted. Then, the process is terminated, and the current best design becomes the output. The mixed-mode simulation setup, metrics extraction, and optimization algorithm are described in the following subsections.

A. Simulation Setup

Fig. 2 depicts the 2-D architecture of planar nMOS and pMOS devices, as well as their transfer characteristics (I_d versus V_g), designed using the Sentaurus TCAD structure

TABLE I
KEY PARAMETERS IN DEFINING THE CMOS INVERTER

Parameter	Bound / Value
L_{G} (Gate Length)	14 $\sim$ 90 nm
T_{Si} (Channel Thickness)	5 $\sim$ 30 nm
W_{Si} (Aspect Ratio of P-MOS)	1 $\sim$ 4
* L_{ext} (S/D Length)	30 nm
* EOT (Effective Oxide Thickness)	2 nm
* N_{SDC} (Doping in the S/D Region)	$1{\times}10^{18}$ cm^{-3}
N_{Ch} (Doping in Channel)	$5{\times}10^{16} \sim 5{\times}10^{17}$ cm^{-3}
C_{L} (Load Capacitor)	0.01 $\sim$ 10 fF

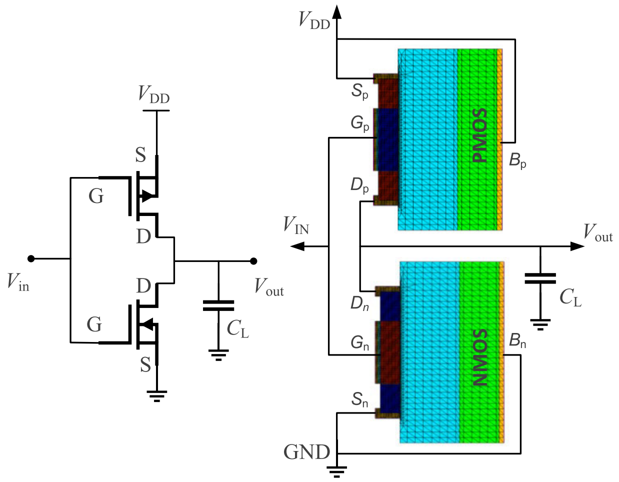

Fig. 3. Schematics and topology of inverter circuit using nMOS and pMOS device showing external node connections and electrical components.

editor and Sdevice tools [29]. The transfer curves show the expected device behavior for a transistor with a gate length of 50 nm and a height of 10 nm. Both n- and p-type devices have the same width area factor. The figure depicts different gate bias conditions required to operate the transistor from liner to saturation regions.

Table I displays the complete list of the parameters required to define this device. Parameters that are not used as design variables are marked with (*). In the TCAD simulation, we included the doping dependence and oldSlotboom mobility along with Shockley–Read–Hall (SRH) recombination models. A mixed-mode simulation setup is then developed, utilizing an individual MOS transistor. Fig. 3 shows all the passive components, external node connections, and power supplies needed for the nMOS and pMOS transistors to operate as an inverter circuit.

Two types of responses are considered to characterize the performance of the inverter: the voltage transfer response (dc characteristics) and the pulse response (pulse characteristics). The dc characteristics are depicted by the input versus output voltage of the circuit, denoted by V_{in} and V_{out}, while pulse characteristics are depicted by V_{in} and V_{out} over time.

B. Metrics Extraction

For an ideal inverter, the output voltage must trigger, vary, and switch simultaneously as the input pulse rises and falls. This can be manually identified and quantified by defining the figures of merit (FoMs), such as rise time, fall time, edge rate, and propagation delay when the circuit functions effectively [30]. However, when nominal device configurations are not provided, setting these figures as optimization metrics is not straightforward. A typical initial response included in Section III demonstrates the challenge of extracting FoMs, where the performance is far from that of a practical inverter.

Therefore, in this study, we transform the raw inverter characteristics V into extracted metrics N compared with the ideal response V_{ideal}

$$f^i = \mathrm{mse}\big(V^i - V^i_{\mathrm{ideal}}\big), \quad i = 1, 2, \ldots, N \tag{1}$$

where mse(·) is the mean square function. For dc characteristics, we divided the curve into three parts to extract features: 1) the high-level region; 2) the low-level region; and 3) the central switching point. Equation (1) is then applied to compute the metric value for each part. Similarly, for pulse characteristics, we derive four parts using the same idea: 1) the high-level region; 2) the low-level region; 3) the rising edge; and 4) the falling edge, comparing with the steepest ideal switching (green dashed line) as well. For an ideal—or rather, theoretical—inverter, all features that we extracted would be zero initially. Even when the circuit's operation state deviates significantly from a practical inverter, the above metrics can be used to discriminate candidate designs with different qualities.

C. Actor–Critic-Based Optimization

The actor–critic-based optimization is a quasi-reinforcement learning algorithm [31] developed from the deep deterministic policy gradient (DDPG) algorithm [32], which is commonly used for continuous space control problems. Traditional reinforcement learning algorithms, such as DDPG, are designed to train agents to operate in specific environments toward defined targets. However, these algorithms typically require thousands or even tens of thousands of operating trajectories for training, which is impractical for semiconductor device and circuit design due to high computational costs. In addition, optimization in most contexts is a non-Markovian process, which violates the fundamental assumptions underlying reinforcement learning. Therefore, applying these reinforcement learning algorithms directly to such problems is inappropriate without adaptation. The actor–critic-based optimization is proposed for expensive optimization tasks, adapting the concepts of "actor" and "critic" from their traditional use in DDPG but with a different purpose. The algorithm trains a critic network based on the current dataset, uses actor network to exploit the promising design region over the model, and outputs the best-predicted design for the next evaluation. We first clarify the main terms of the algorithm and then describe the procedure in detail.

1) *Design Space:* This is the space of the design variables x for the inverter, which is continuous and bounded within given intervals.
2) *Action Space:* This refers to the space of the perturbations, or actions, a applied to the design variables,

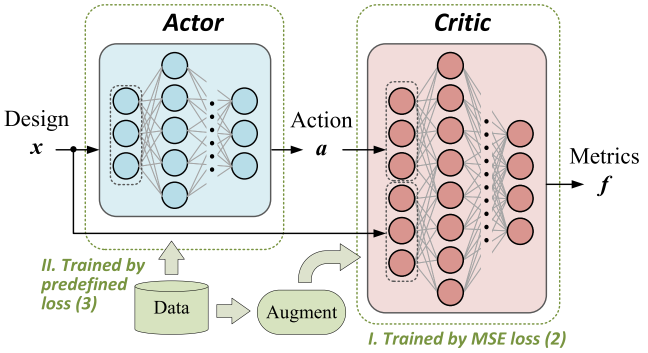
Actor
Critic
Design
x
Action
a
Metrics
f
II. Trained by predefined loss (3)
Data
Augment
I. Trained by MSE loss (2)

Fig. 4. Structure of the actor and critic network as well as their training process.

Algorithm 1 Actor–Critic-Based Optimization

```
Input: Actor network $\mu_\theta$, Critic network $Q_\phi$, dataset $\mathcal{D}$, and
    inherited parameter $\phi^-$(optional).
Initialize $\theta$ and $\phi$
if $\phi^-$ is defined then
    $\phi \leftarrow \phi^-$
end if
Augment dataset $\mathcal{D}$ into $\mathcal{B}$
Train the critic network $Q_\phi$ on $\mathcal{B}$ by Eqn. 2
Train the actor network $\mu_\theta$ on $\mathcal{D}$ using Eqn. 3
$\mathcal{F} \leftarrow \{\}$
for each $x \in \mathcal{D}$ do
    $a \leftarrow \mu_\theta(x) + \epsilon(x_{r_1} - x_{r_2})$    ▷ Add noise on action.
    $\hat{f} \leftarrow Q_\phi(x, a)$
    if $\hat{f} < \min(\mathcal{F})$ then
        $x^* \leftarrow x + a$
    end if
    $\mathcal{F} \leftarrow \hat{f} \cup \mathcal{F}$
end for
$\phi^- \leftarrow \phi$
Output: Suggest the new design $x^*$
```

where $x + a$ forms a new design within the design space. The action space is continuous and typically bounded by twice the design intervals.

3) *Data Augmentation:* As the only available data for the training model are the design–metrics pairs extracted from the simulation during optimization, the data volume is quite limited. To mitigate training challenges, we augment the design–metrics pairs into design–action–metrics triples. Given a set of design–metrics pairs $\mathcal{D} = \{(x, f)\}$, the augmented dataset is $\mathcal{B} = \{(x, a, f) | (x + a, f) \in \mathcal{D}\}$. Obviously, if $\mathcal{D}$ contains N elements, then $\mathcal{B}$ contains N^2 elements.

4) *Critic Network:* A neural network $Q_\phi(x, a)$, parameterized by ϕ, accepts a design x and an action a and predicts the performance of the design $x + a$. Q_ϕ is trained using the mean square error on a set of design–action–metrics triples by

$$L_\phi(x, a) = \big(Q_\phi(x, a) - f(x + a)\big)^2. \tag{2}$$

The critic network inherits its parameter ϕ from the last training, denoted by ϕ^-. The inheritance makes training faster and easier in a sequential manner.

5) *Actor Network:* A neural network $\mu_\theta(x)$, parameterized by θ, produces the most promising action for a given design x. μ_θ is trained after the training of the critic network. The loss function aims to minimize the weighted sum of metrics estimated by the critic network, effectively exploring the design space while heading to promising regions of lower metrics. In addition, to ensure that the produced design $x + \mu_\theta(x)$ stays within the given bounds, a penalty term is added to the loss function

$$L_\theta(x) = w(Q_\phi(x, \mu_\theta(x))) + \psi(x + \mu_\theta(x)) \tag{3}$$

where $w(\cdot)$ is the weighted sum function and $\psi(\cdot)$ enforces the bound constraints defined by

$$\psi(x) = \left\| \frac{\max(0, x_{\text{low}} - x)}{x_{\text{high}} - x_{\text{low}}} \right\|_2^2 + \left\| \frac{\max(0, x - x_{\text{high}})}{x_{\text{high}} - x_{\text{low}}} \right\|_2^2$$

where x_{low} and x_{high} are the lowest and highest values of the design in the dataset $\mathcal{D}$.

Fig. 4 outlines the structure of the actor and the critic network. Specifically, the actor network is employed to generate the best action over a given design, and the critic network is employed as an estimator to evaluate how good the new design is. Both networks are sequentially stacked by a fully connected layer, batch-normalization layer, and rectified activation layer. The training algorithm is stochastic gradient descent.

The pseudocode of the actor–critic-based optimization is shown in Algorithm 1. Following network training, each design in the current dataset is passed to the actor network to generate an action. The action is then added with the noise consisting of the difference between two randomly selected designs from the dataset, scaled by a hyperparameter ϵ. The purpose is to balance the exploration and exploitation and enhance the algorithm's robustness. The new design added with the action is input into the critic network for querying metrics, and only the one with the best predictive metrics is output for simulation in the next iteration.

The number of hidden layers and neurons in each layer is set to ensure that the networks have sufficient modeling capability; therefore, these are generally determined by the number of parameters and metrics. In our experiment, the critic network has two hidden layers, each containing 16 neurons, while the actor network has three hidden layers, each containing 16, 23, and 16 neurons, respectively. In addition, ϵ is set to 0.1 for good balancing ability, while the number of initial samples is set to 25.

III. Results and Discussion

The evaluation of the CMOS inverter's performance was conducted through an analysis of its dc response and pulse transient response. To validate our methodology, two design cases in nanosecond and picosecond pulses are considered, respectively. The input pulse for the nanosecond case is 100 ns, while that for the picosecond case is 140 ps. These two-pulse cases are chosen based on [33]. Due to the limitation of the device's maximum oscillation frequency (MUF), meeting the

TABLE II
OPTIMAL PARAMETER VALUES FOR VARIOUS CASES

	Case 1 (ns pulse)		**Case 2 (ps pulse)**	
	NMOS	PMOS	NMOS	PMOS
L_G (nm)	58		24	
T_{Si} (nm)	24	27	12	13
N_{Ch} ($\times 10^{17}$cm^{-3})	3.8	5	0.5	0.7
W_{Si}	-	1.7	-	4
C_L (fF)	0.5		0.01	

	Case 2 (diff. L_G) (ps pulse)		
	NMOS	PMOS	Man. N/P
L_G (nm)	17	20	50
T_{Si} (nm)	12	15	20
N_{Ch} ($\times 10^{17}$cm^{-3})	2.6	2.3	5 / 6
W_{Si}	-	3.1	- / 2
C_L (fF)	0.01		0.01

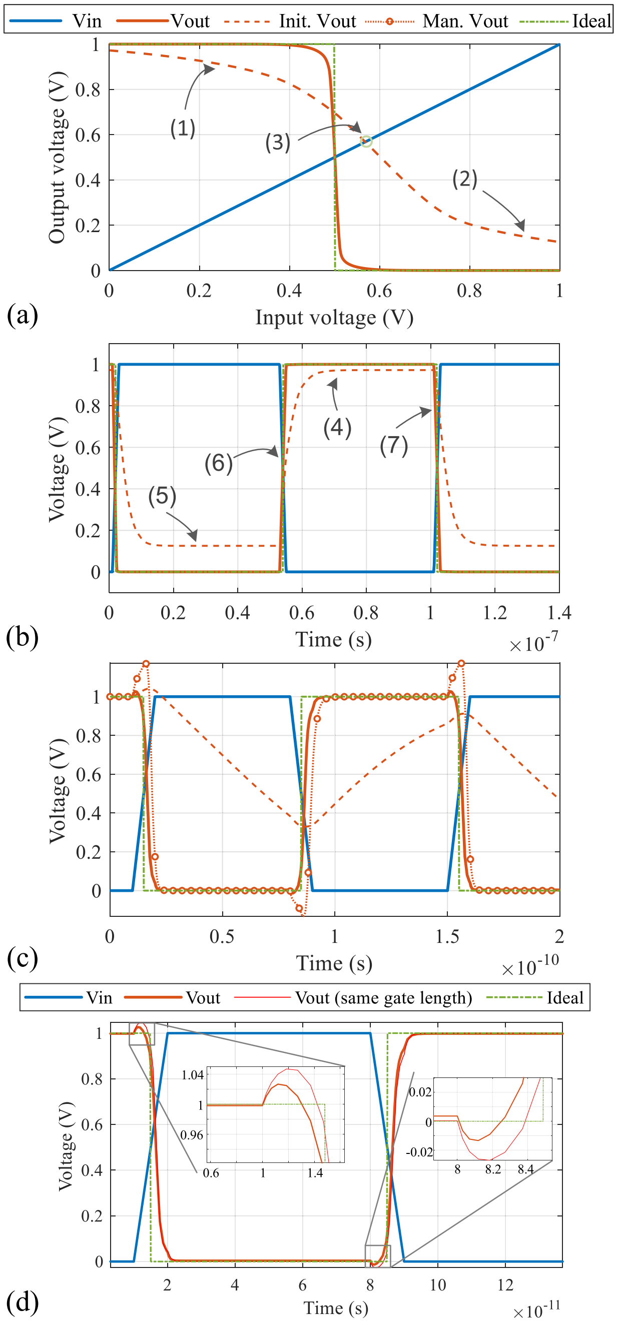

Fig. 5. DC and pulse performance of two design cases, as well as feature illustration. (a) DC characteristics. (b) Pulse characteristics of nanosecond design case. (c) Pulse characteristics of picosecond design case (different gate lengths). (d) Comparison of pulse characteristics of the same and different gate lengths.

standard with a nanosecond pulse for a CMOS inverter is relatively straightforward, while maintaining the same performance with a picosecond pulse is more challenging. In this case, the output signal may deviate significantly from the requirement, leading to additional delay, overshoot, and other performance degradation. Therefore, it was chosen as an excellent case to demonstrate how our methodology can help to find the optimal solution.

Considering manufacturability, we keep the same gate length of nMOS and pMOS for both cases and also provide the most flexible case in which gate lengths can be different to fully explore the design space. The resulting designs are compared with respect to transfer characteristics, providing much inspiration for the future technology node.

Table II lists the best design parameter values, which we found within 100 iterations for the first case and 200 iterations for the second case. All the parameter values are obtained by the algorithm to get the desired output characteristics close to the ideal one (the target data). We allocated more iterations to the second case to ensure convergence. Compared with other modeling-based methodologies, our approach incurs significantly lower computing costs [14], [25], since thousands of simulations for training device models are saved.

The corresponding dc and pulse characteristics are illustrated in Fig. 5 by solid lines. As shown in Fig. 5(a), the voltage transfer characteristic of the inverter reveals the expected inverting behavior. The input voltage V_{in} is plotted against the output voltage V_{out}, along with the typical initial and manual V_{out} points to depict the switching threshold. Three distinct regions can be observed as Region (1): this represents the cutoff region where V_{out} is high and almost equal to V_{DD}, indicating that the nMOS is in the OFF state and the pMOS is in the ON state. Region (2): the saturation region where V_{out} approaches ground level, indicating that nMOS is on and pMOS is off. And Region (3): the transition region is characterized by a sharp fall in V_{out} as V_{in} increases. This region is crucial as it defines the inverter's switching threshold and gain.

Voltage transfer characteristics indicate a superior switching behavior of the optimized inverter over the manually designed inverter with steeper slopes for both rising and falling edges. Regions (1) and (2) indicate that the typical initial design's pull-up and pull-down functions are not as effective as those of the co-optimized design. The intersection point of the input and output curves, which ideally should be half V_{DD} for the inverter, is significantly different in the typical scenario as

TABLE III
COMPARISON OF FoMs OF THE DESIGNS OPERATING ON PICOSECOND PULSE

	Co-opt.	Manual	[36]	[34]
Technology	Planar FET	Planar FET	SOI C-FinFET	LDD-FinFET
Supply voltage (V)	1.0	1.0	1.0	1.5
Signal Period (ps)	140	140	200	140
Rise time (ps)	4.20	7.95	10.00	4.56
Fall time (ps)	3.6	8.22	10.00	4.10
Edge rise (ps)	3.9	8.08	10.00	4.33
High to low delay (ps)	1.45	3.66	2.11	3.27
Low to high delay (ps)	1.52	4.5	1.49	3.55
Propagation delay (ps)	1.58	4.08	1.30	3.41
Contamination delay (ps)	1.45	3.66	0.9	3.27
MUF (THz)	0.12	0.06	0.05	0.11
Overshoot (V)	0.026	0.170	-	-

compared with the co-optimized design. This suggests a better noise margin and a more robust operation in the presence of voltage variations.

The pulse characteristics, as shown in Fig. 5(b), demonstrate the dynamic responses of the inverter, including the following: Region (4)—the high-level output, Region (5)—the low-level output, Region (6)—the rising edge, and Region (7)—the falling edge. We compare the performance with the existing literature [33], [34] and the manually designed circuit. Table III provides a comprehensive comparison of FoMs for CMOS inverters based on different technology nodes and design optimizations. The metrics compared include rise time, fall time, edge rise, delay times, propagation delay, contamination delay, MUF, and overshoot voltage.

The optimized inverter design using planar FET technology shows superior performance compared with the manually optimized (manual) planar FET design across several parameters, as shown in Fig. 5(b) and (c). Specifically, the ML-assisted design demonstrates faster rise and fall times (4.20 and 3.6 ps, respectively) compared with the manual design (7.95 and 8.22 ps, respectively). This indicates a more rapid transition between logic states, which is critical for high-speed applications.

Voltage overshoot during the rise and fall periods of the pulse is a critical parameter for signal integrity and reliability. This depends on the charging and discharging times of the output capacitor further determined by its time constant. When comparing MOS devices with the same and different gate lengths, as seen in Fig. 5(d), we found that the voltage deviation at rise and fall periods is smaller when the gate length is used as an optimized parameter separately for nMOS and pMOS devices than when the variable is constrained to the same value throughout optimization. This is because the gate length influences the fringing capacitance associated with the channel components as affects the intrinsic capacitance value of the device [33]. The ML-assisted design demonstrates a negligible undershoot and a minimal overshoot of 0.026 V, which is significantly lower than the manual design's overshoot of 0.170 V. This suggests that the ML-assisted design is not only faster but also more precise, with less risk of damaging other components or causing logic errors due to excessive voltage. Hence, the proposed methodology is helpful for finding the most optimal parameter set. Moreover, this work can be easily extended to complicated circuits and speed up time to market of new devices.

To demonstrate the robustness of the optimization algorithm, we conducted experiments with different configurations of algorithm parameters. The results indicate that the bound constraint $\psi(\cdot)$ in (3) is indispensable. Without this constraint in loss function of actor network, the algorithm suggests new designs that are impractical and far outside the given range. The settings for the number of hidden layers and neurons in each layer prove to be robust, provided the modeling capacity is sufficient. Therefore, configuring two or more hidden layers, with each layer having more than $2d$ neurons, where d is the number of neurons of the input layer, is adequate and has a limited impact on the final outcome. The noise figure balances efficiency and exploration; a larger ϵ results in a slower convergence rate. For this optimization task, a value of 0.1 was deemed appropriate by our experiments.

IV. CONCLUSION

In conclusion, this study showcased how ML techniques, in particular, actor–critic-based ML techniques, can be effectively applied to device circuit co-optimization. Through the demonstration of an inverter cell, we obtained high-quality voltage transfer characteristics and switching characteristics manifesting the good on and off behaviors, by optimizing critical parameters, including area factor, doping concentration, and load capacitance. As per the observations, significant advantages over manual optimizations were shown, although the design process is relatively opaque and lacks explainability compared with manual design processes. The challenges of the proposed method lie in its capability to be applied to more intricate 3-D structures, such as CFET and GAA FET, which will be addressed in our future work. Nonetheless, this study paves the way for future research in applying ML to more complex circuit design challenges, highlighting the impact of ML in the field of semiconductor engineering.

REFERENCES

[1] V. B. Sreenivasulu and V. Narendar, "Design insights of nanosheet FET and CMOS circuit applications at 5-nm technology node," *IEEE Trans. Electron Devices*, vol. 69, no. 8, pp. 4115–4122, Aug. 2022, doi: 10.1109/TED.2022.3181575.

[2] N. Loubet et al., "Stacked nanosheet gate-all-around transistor to enable scaling beyond FinFET," in *Proc. Symp. VLSI Technol.*, Jun. 2017, pp. T230–T231, doi: 10.23919/VLSIT.2017.7998183.

[3] B. Razavi, *Design of Analog CMOS Integrated Circuits*. New York, NY, USA: McGraw-Hill, 2005.

[4] B. H. Calhoun et al., "Digital circuit design challenges and opportunities in the era of nanoscale CMOS," *Proc. IEEE*, vol. 96, no. 2, pp. 343–365, Feb. 2008.

[5] G. Biswas, S. R. Goldman, D. Fisher, B. Bhuva, and G. Glewwe, "Assessing design activity in complex CMOS circuit design," in *Cognitively Diagnostic Assessment*. Evanston, IL, USA: Routledge, 2012, pp. 167–188.

[6] P. K. Kori, A. Dixit, C. Rajan, and D. P. Samajdar, “22 nm LDD FinFET based novel mixed signal application: Design and investigation,” *Silicon*, vol. 14, no. 15, pp. 9453–9465, Oct. 2022.

[7] C. Jeong et al., “Bridging TCAD and AI: Its application to semiconductor design,” *IEEE Trans. Electron Devices*, vol. 68, no. 11, pp. 5364–5371, Nov. 2021, doi: 10.1109/TED.2021.3093844.

[8] Y. Lyu, W. Chen, M. Zheng, B. Yin, J. Li, and L. Cai, “Machine learning-assisted device modeling with process variations for advanced technology,” *IEEE J. Electron Devices Soc.*, vol. 11, pp. 303–310, 2023, doi: 10.1109/JEDS.2023.3277548.

[9] J. Wang, Y.-H. Kim, J. Ryu, C. Jeong, W. Choi, and D. Kim, “Artificial neural network-based compact modeling methodology for advanced transistors,” *IEEE Trans. Electron Devices*, vol. 68, no. 3, pp. 1318–1325, Mar. 2021, doi: 10.1109/TED.2020.3048918.

[10] B. Kim and M. Shin, “A novel neural-network device modeling based on physics-informed machine learning,” *IEEE Trans. Electron Devices*, vol. 70, no. 11, pp. 6021–6025, Nov. 2023, doi: 10.1109/TED.2023.3316635.

[11] Z. Wang et al., “Improving semiconductor device modeling for electronic design automation by machine learning techniques,” *IEEE Trans. Electron Devices*, vol. 71, no. 1, pp. 263–271, Jan. 2024, doi: 10.1109/TED.2023.3307051.

[12] G. Choe, J. Kwak, and S. Yu, “Machine learning-assisted compact modeling of W-doped indium oxide channel transistor for back-end-of-line applications,” *IEEE Trans. Electron Devices*, vol. 71, no. 1, pp. 231–238, Jan. 2024, doi: 10.1109/TED.2023.3296715.

[13] K. Sheelvardhan, S. Guglani, M. Ehteshamuddin, S. Roy, and A. Dasgupta, “Machine learning augmented compact modeling for simultaneous improvement in computational speed and accuracy,” *IEEE Trans. Electron Devices*, vol. 71, no. 1, pp. 239–245, Jan. 2024, doi: 10.1109/TED.2023.3251296.

[14] Y.-S. Yang, Y. Li, and S. R. R. Kola, “A physical-based artificial neural networks compact modeling framework for emerging FETs,” *IEEE Trans. Electron Devices*, vol. 71, no. 1, pp. 223–230, Jan. 2024, doi: 10.1109/TED.2023.3269410.

[15] H. Dhillon, K. Mehta, M. Xiao, B. Wang, Y. Zhang, and H. Y. Wong, “TCAD-augmented machine learning with and without domain expertise,” *IEEE Trans. Electron Devices*, vol. 68, no. 11, pp. 5498–5503, Nov. 2021, doi: 10.1109/TED.2021.3073378.

[16] P. Aleksandrov, A. Rezaei, T. Dutta, N. Xeni, A. Asenov, and V. Georgiev, “Convolutional machine learning method for accelerating nonequilibrium green’s function simulations in nanosheet transistor,” *IEEE Trans. Electron Devices*, vol. 70, no. 10, pp. 5448–5453, Oct. 2023, doi: 10.1109/TED.2023.3306319.

[17] H. Carrillo-Nu nez, N. Dimitrova, A. Asenov, and V. Georgiev, “Machine learning approach for predicting the effect of statistical variability in Si junctionless nanowire transistors,” *IEEE Electron Device Lett.*, vol. 40, no. 9, pp. 1366–1369, Sep. 2019.

[18] K. Ko, J. K. Lee, M. Kang, J. Jeon, and H. Shin, “Prediction of process variation effect for ultrascaled GAA vertical FET devices using a machine learning approach,” *IEEE Trans. Electron Devices*, vol. 66, no. 10, pp. 4474–4477, Oct. 2019.

[19] G. Choe et al., “Machine learning-assisted statistical variation analysis of ferroelectric transistor: From experimental metrology to adaptive modeling,” *IEEE Trans. Electron Devices*, vol. 70, no. 4, pp. 2015–2020, Apr. 2023, doi: 10.1109/TED.2023.3244764.

[20] J. K. Lee, K. Ko, and H. Shin, “Prediction of random grain boundary variation effect of 3-D NAND flash memory using a machine learning approach,” *IEEE Trans. Electron Devices*, vol. 69, no. 1, pp. 447–449, Jan. 2022, doi: 10.1109/TED.2021.3130858.

[21] K. Ko, J. K. Lee, and H. Shin, “Variability-aware machine learning strategy for 3-D NAND flash memories,” *IEEE Trans. Electron Devices*, vol. 67, no. 4, pp. 1575–1580, Apr. 2020, doi: 10.1109/TED.2020.2971784.

[22] Y. S. Bankapalli and H. Y. Wong, “TCAD augmented machine learning for semiconductor device failure troubleshooting and reverse engineering,” in *Proc. IEEE Int. Conf. Simul. Semicond. Processes Devices (SISPAD)*, Sep. 2019, pp. 1–4, doi: 10.1109/SISPAD.2019.8870467.

[23] S. K. Singh, D. Sharma, P. Srinivasan, and A. Dixit, “Quantum-dot-based thermometry using 12-nm FinFET and machine learning models,” *IEEE Trans. Electron Devices*, vol. 71, no. 3, pp. 2043–2050, Mar. 2024.

[24] J. Wang, L.-Y. Xue, B. Liu, and C. Li, “Design of terahertz InP pHEMT using machine learning assisted global optimization techniques,” in *Proc. 16th Eur. Microw. Integr. Circuits Conf. (EuMIC)*, pp. 67–70, Apr. 2022, doi: 10.23919/EuMIC50153.2022.9784068.

[25] M. Ehteshamuddin, K. Sheelvardhan, A. Kumar, S. Guglani, S. Roy, and A. Dasgupta, “Machine learning-assisted multiobjective optimization of advanced node gate-all-around transistor for logic and RF applications,” *IEEE Trans. Electron Devices*, vol. 71, no. 2, pp. 976–982, Feb. 2024, doi: 10.1109/TED.2023.3345288.

[26] H. Xu et al., “A machine learning approach for optimization of channel geometry and source/drain doping profile of stacked nanosheet transistors,” *IEEE Trans. Electron Devices*, vol. 69, no. 7, pp. 3568–3574, Jul. 2022, doi: 10.1109/TED.2022.3175708.

[27] C.-C. Chiang et al., “Design and process co-optimization of 2-D monolayer transistors via machine learning,” *IEEE Trans. Electron Devices*, vol. 70, no. 11, pp. 5991–5996, Nov. 2023, doi: 10.1109/TED.2023.3310942.

[28] T. Wu and J. Guo, “Multiobjective design of 2-D-material-based field-effect transistors with machine learning methods,” *IEEE Trans. Electron Devices*, vol. 68, no. 11, pp. 5476–5482, Nov. 2021, doi: 10.1109/TED.2021.3085701.

[29] *Sentaurus Device User Guide, V-2023.09*, Synopsys, Sunnyvale, CA, USA, 2023.

[30] S. C. Rustagi et al., “CMOS inverter based on gate-all-around silicon-nanowire MOSFETs fabricated using top-down approach,” *IEEE Electron Device Lett.*, vol. 28, no. 11, pp. 1021–1024, Nov. 2007, doi: 10.1109/LED.2007.906622.

[31] A. F. Budak, P. Bhansali, B. Liu, N. Sun, D. Z. Pan, and C. V. Kashyap, “DNN-Opt: An RL inspired optimization for analog circuit sizing using deep neural networks,” in *Proc. 58th ACM/IEEE Des. Automat. Conf. (DAC)*, Dec. 2021, pp. 1219–1224.

[32] T. P. Lillicrap et al., “Continuous control with deep reinforcement learning,” 2015, *arXiv:1509.02971*.

[33] A. Dixit and D. P. Samajdar, “Extraction of performance parameters of nanoscale SOI LDD-FinFET using a semi-analytical model of capacitance and channel potential for low-power applications,” *Appl. Phys. A*, vol. 126, no. 10, p. 782, Oct. 2020.

[34] S. S. Zaman, P. Kumar, M. P. Sarma, A. Ray, and G. Trivedi, “Design and simulation of SF-FinFET and SD-FinFET and their performance in analog, RF and digital applications,” in *Proc. IEEE Int. Symp. Nanoelectron. Inf. Syst. (iNIS)*, Dec. 2017, pp. 200–205.