Accelerating DTCO with a Sample-Efficient Active Learning Framework for TCAD Device Modeling

Chanwoo Park[*], Junghwan Park[*], Premkumar Vincent, Hyunbo Cho
Research & Development Center, Alsemy Inc.
Seoul, South Korea

ABSTRACT

Design-Technology Co-Optimization (DTCO) can be significantly accelerated by employing Neural Compact Models (NCMs). However, the effective deployment of NCMs requires a substantial amount of training data for accurate device modeling. This paper introduces an Active Learning (AL) framework designed to enhance the efficiency of both device modeling and process optimization, particularly addressing the challenges of time-intensive Technology Computer-Aided Design (TCAD) simulations. The framework employs a ranking algorithm that assesses metrics such as the expected variance from the neural tangent kernel (NTK), TCAD simulation time, and the complexity of I-V curves. This strategy considerably reduces the number of required simulations while maintaining high accuracy. Demonstrating the effectiveness of our AL framework, we achieved a 28.5% improvement in MSE within a 30-minute time budget for device modeling, and an 86.7% reduction in the data points required for process optimization of a 51-stage ring oscillator (RO). These results offer a streamlined, adaptable solution for rapid device modeling and process optimization in various DTCO applications.

KEYWORDS

Active Learning, TCAD, DTCO, Neural Compact Model, Device Modeling, Process Optimization

ACM Reference Format:
Chanwoo Park[*], Junghwan Park[*], Premkumar Vincent, Hyunbo Cho. 2024. Accelerating DTCO with a Sample-Efficient Active Learning Framework for TCAD Device Modeling. In *61st ACM/IEEE Design Automation Conference (DAC '24), June 23–27, 2024, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649329.3655925

1 INTRODUCTION

With technology scaling and the increasing complexity of manufacturing processes, Design-Technology Co-Optimization (DTCO) has become vital [1]. A key component of DTCO is Technology Computer-Aided Design (TCAD) simulations, which produce detailed semiconductor device characteristics. The TCAD data is subsequently converted into compact models. These models act as a crucial bridge between TCAD simulations and SPICE (Simulation Program with Integrated Circuit Emphasis) simulations, facilitating the evaluation of Figures of Merit (FoM) for circuits. By providing a link between detailed device characteristics and circuit-level performance, compact models play a pivotal role in the DTCO cycle. The feedback obtained from SPICE simulations, in turn, informs the optimization of device design and manufacturing, creating a continuous loop of improvement.

Traditional compact models, such as the Berkeley Short-channel IGFET Model (BSIM), have been essential in simulating devices like MOSFETs. However, new physical phenomena from device scaling have led to more complex equations, longer development times, and an increased number of fitting parameters in these models. Additionally, Model Parameter Extraction (MPE) in circuit simulations contributes to extended turnaround times for new technology node development.

Neural Compact Models (NCMs) have been introduced as an alternative to mitigate these challenges. Based on Artificial Neural Network (ANN), NCMs can characterize a diverse range of devices without the need for parameter extraction or intricate physics understanding. They provide faster compact model generation, making them a time-efficient solution for the iterative cycles of device and circuit optimization in DTCO. However, a significant challenge remains: these NCMs require large amounts of training data. Given the time-intensive nature of TCAD simulations, producing a sufficient volume of training data to ensure the accuracy of device modeling is not a trivial task.

To address this challenge, we introduce an Active Learning (AL) framework that enhances the efficiency of Neural Compact Model (NCM) generation. This framework strategically selects the necessary TCAD data points for constructing NCMs, ensuring high model accuracy with significantly fewer simulations. The core of our AL framework is a novel ranking system, which starts from a limited dataset and systematically queries the next TCAD data points for simulation. Our AL framework's ranking system effectively balances three key metrics: expected variance based on the neural tangent kernel (NTK), anticipated duration of TCAD simulations, and the complexity of the I-V curve, as determined by a transformer-based pretrained model. By leveraging these metrics, the system prioritizes data points that are most informative, due to their high uncertainty and function complexity, and also efficient in terms of simulation time.

In our experiments, we validated our Active Learning (AL) framework in two key scenarios: device modeling using TCAD calibrated to 45nm technology, aimed at achieving target accuracy with fewer simulations, and process optimization using TCAD calibrated to 32nm technology, focused on optimizing process parameters to minimize Figures of Merit (FoM) such as power and delay in ring

[*] Equal contribution; Email: {chanwoo.park, junghwan.park}@alsemy.com.

DAC '24, June 23–27, 2024, San Francisco, CA, USA

ACM ISBN 979-8-4007-0601-1/24/06…$15.00
https://doi.org/10.1145/3649329.3655925

oscillators (ROs). The AL framework demonstrated substantial efficiency in both scenarios. In device modeling, it achieved a 28.5% improvement in MSE within a 30-minute time budget. For process optimization in a 51-stage ring oscillator, the framework realized an 86.7% reduction in TCAD simulation time. These significant advancements in efficiency enable robust and rapid device modeling across various Design-Technology Co-Optimization (DTCO) applications, offering a viable solution to the challenges of traditional analytical compact models and leveraging the strengths of ANN-based compact models.

The key contributions of this work include:

- Introducing an AL framework for TCAD simulations, significantly reducing the computational resources and time required for device modeling and process optimization.
- Offering pretrained models that enhance device modeling by strategically selecting the most informative data points without the need for further training.
- Demonstrating the efficacy of our framework in real-world scenarios, including device simulation and process optimization using ROs.

2 RELATED WORK

Active Learning. Active learning allows a model to iteratively query unlabeled data to improve its performance. A common method, uncertainty-based selection, often picks highly correlated data, which may not be efficient. Additionally, it typically requires pretrained models, prolonging the process [2, 3]. A novel approach using Gaussian processes (GP) and a neural tangent kernel has been introduced, which selects data to minimize expected variance in test data without needing neural network (NN) training. This method enhances predictive performance and computational efficiency [4].

Neural Tangent Kernel (NTK). The NTK has emerged as a crucial framework for analyzing deep learning models, especially under the infinite-width condition as highlighted by Jacot et al. [5]. It is formulated as $\nabla_\theta f(\mathcal{X};\theta_t)\nabla_\theta f(\mathcal{X};\theta_t)^T$, with $f(\mathcal{X};\theta_t)$ denoting the model's predictive output. This framework simplifies the analysis of neural networks' training dynamics, showing how their behavior becomes more predictable as width increases [6]. The empirical NTK stabilizes through training, ultimately converging to a fixed kernel. Utilizing Gaussian Processes (GP) alongside NTK in our study, we aim to diminish generalization error and bolster initialization robustness, thereby increasing NN reliability across different initial conditions. Further elaboration is in Section 3.3.

Neural Compact Models. NCMs offer efficient solutions for device modeling within Design-Technology Co-Optimization (DTCO), with applications ranging from MOSFET mobility models ensuring charge conservation and precise leakage current modeling [7], to modeling Fin-FETs, tunnel FETs [1], and non-ideal diodes through graph neural networks [8]. These scalable models significantly speed up electrical characteristic prediction, supporting a wide array of circuits in DTCO [9].

Despite their benefits, NCMs face hurdles such as high computational requirements and the need for extensive datasets. Recent efforts have mitigated these challenges by incorporating physics-based insights into ANN designs and data processing [10, 11], and by developing hybrid models that merge traditional and ANN methodologies for devices like GAA MOSFETs and IGBTs [12, 13]. These approaches enhance model accuracy and comprehensiveness, capturing essential device behaviors and variations.

Building on these developments, our active learning framework aims to streamline DTCO by enhancing NCM sample efficiency and TCAD simulation. This strategy reduces device characterization time and costs, boosts simulation precision, and expedites design cycles, distinguishing our work from previous efforts.

3 METHOD

3.1 TCAD setup

A 2D 45nm n-MOSFET device was simulated in Sentaurus TCAD. The n-type Gaussian profile doping, with a peak doping concentration of 8×10^{20} cm^{-3}, forms the source and drain. The p-type substrate has a doping concentration of 3.24×10^{18} cm^{-3}. The gate electrode is a 100 nm thick poly-Si, with the source and drain electrodes designed as electrode contact edges. The additional parameters employed for the TCAD simulations are outlined in Table 1. Quasi-static DC simulation sweeps of drain voltage (V_{DS}), gate voltage (V_{GS}), and substrate/body voltage (V_{BS}) were performed to extract the drain current (I_D). The time required to solve each data point was determined from the log files, facilitating a performance comparison with our AL framework.

Table 1: TCAD Parameters for Device Modeling.

Parameters	Value
Length	45 nm
Gate oxide thickness	1 nm
Gate electrode material	Poly-Si
Gate electrode thickness	100 nm
Boron doping	3.24×10^{18}
Arsenic doping	8×10^{20} cm^{-3}
Doping profile	Gaussian
Junction depth (X_j)	14 nm
Extension junction depth	$X_j * 0.4$
Source/Drain electrode length	50 nm
Substrate thickness	300 nm

3.2 Active Learning Procedure

Our AL framework is illustrated in Fig. 1 and detailed in Alg. 1. We start with a training set of 105 data points, selected through grid sampling from the minimum to the maximum range of the input values. This initial dataset is split into training and validation subsets. The criterion for determining whether further TCAD simulations are needed for additional data acquisition is the MSE measured on the validation set. At each iteration, we enhance the training set by incorporating 10 more samples, selected according to a designated ranking system, continuing this process until the desired validation error is reached. Subsequently, the trained NCMs are employed in SPICE simulations to analyze the FoMs of the circuits, thereby setting up a feedback loop for ongoing TCAD device optimization.

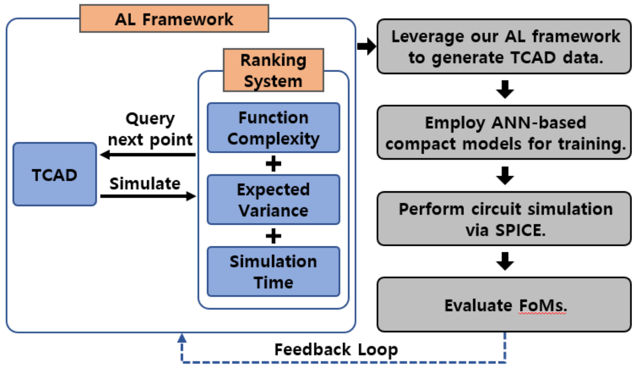


Figure 1: Workflow of the Proposed AL Framework. The framework iteratively queries TCAD (as an oracle) using three metrics, streamlining device modeling and accelerating DTCO.

Algorithm 1 Active Learning Procedure

```
Input: Initial labeled data $\mathcal{L}$, validation data $\mathcal{V}$, unlabeled data $\mathcal{U}$, query size $q$, function complexity estimator $f_c$, simulation time estimator $f_t$, GP with NTK ($GP_{\text{NTK}}$)
while error > target do
    for each batch of $q$ iterations do
        for $i$ from 1 to $|\mathcal{U}|$ do
            $m_c = f_c(x_i|\mathcal{L})$;
            $m_t = f_t(x_i)$;
            $m_{ev} = -\frac{1}{|\mathcal{U}|}\sum_{x\in\mathcal{U}} \sigma^2_{GP_{\text{NTK}}}(x|\mathcal{L}\cup\{x_i\})$;
            Compute the aggregated score:
            $\text{score}_i = w_c \cdot m_c + w_t \cdot m_t + w_{ev} \cdot m_{ev}$;
        end for
        $x^* = \arg\max_{x\in\mathcal{U}}\{\text{score}_i\}_{i=1}^{|\mathcal{U}|}$;
        $\mathcal{L} = \mathcal{L}\cup\{(x^*, y^*)\}$, $\mathcal{U} = \mathcal{U}\setminus\{x^*\}$;
    end for
    error = $\frac{1}{|\mathcal{V}|}\sum_{(x,y)\in\mathcal{V}}(GP_{\text{NTK}}(x|\mathcal{L}) - y)^2$
end while
```

Our Active Learning (AL) framework, depicted in Fig. 1 and outlined in Alg. 1, streamlines device modeling and process optimization in DTCO. It starts with an initial set of 105 TCAD data points, derived through grid sampling, which are then split into training and validation subsets. The framework first interacts with the TCAD simulator, applying a ranking system that assesses uncertainty, estimated simulation time, and function complexity to determine the most informative points for subsequent simulations. This ranking system is integral to selecting new data points that are added to the training set in each iteration, enhancing the dataset for Neural Compact Model (NCM) training.

After augmenting the dataset, the trained NCMs are utilized in SPICE simulations to evaluate circuit Figures of Merit (FoM), such as power and delay. The results from SPICE provide feedback for adjusting the TCAD setup, particularly the process parameters, leading to a continuous improvement loop. The framework repeats this process, consistently refining the training set and applying the updated insights to the TCAD simulator, until the desired validation error on the MSE is achieved. This method offers a dynamic and iterative approach, effectively balancing the demands of simulation and accuracy, thus advancing the efficiency of DTCO.

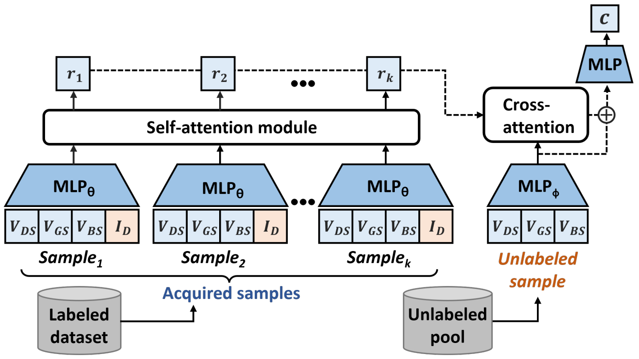


Figure 2: Function Complexity Network Architecture. Using attention techniques, the context set of TCAD data is encoded. This information guides MLPs to estimate the I-V curve's curvature at unlabeled biases.

The AL framework employs a neural network model with four hidden layers and 64 nodes. The model is trained for 500 epochs using the Adam optimizer, with a learning rate of 0.01 and a batch size of 128. We implement a StepLR learning rate scheduler with a 75 step size and a 0.8 gamma factor. To ensure robustness and reproducibility, each experiment was conducted five times using different random seeds, allowing us to compute the mean and standard deviation of the results.

3.3 Framework Strategy and Evaluation Metrics

Our AL framework utilizes a ranking algorithm that integrates three metrics for querying the next point to simulate: targeting regions that can potentially decrease GP's expected variance for test data; focusing on areas with lower data acquisition cost, synonymous with short TCAD simulation times; and highlighting the high estimated complexity of the target function. The function's complexity is inferred using a pre-trained estimator, which is trained on the current-voltage (I-V) characteristics of various MOSFET devices.

To estimate the expected variance, we employ a GP coupled with the NTK. Leveraging the NTK allows us to compute the GP's expected variance based on the similarities among input points. This process essentially assists in narrowing down regions that can significantly reduce the GP's expected variance for the test data, thereby enhancing the reliability and performance of the active learning framework.

In our study, we introduce a transformer-based encoder-decoder module (shown in Fig. 2) to estimate the complexity of I-V characteristic curves at new, unseen data points. This module is specifically designed to give higher importance to points in regions with high curvature, utilizing data from TCAD simulations. As outlined in Alg. 2, our approach starts with a self-attention mechanism applied to the already acquired TCAD data, enabling the model to focus on different aspects of the input depending on the task. Following this, a cross-attention mechanism processes new, unlabeled data points in relation to the existing dataset. The resulting representation of unlabeled data is processed by a MLP to determine the estimated complexity of the function at these new points. Our approach thus ensures a comprehensive understanding of the data, enabling precise predictions of function complexity in various regions of the curve.

In Fig. 3, we visually represent function complexity with a color scheme: red denotes high complexity data points, and blue indicates lower complexity points. Both 3a and 3b corroborate this, showing that regions of higher estimated curvature of the I_D curve are marked red, signifying complexity, while linear sections appear blue.

The loss was evaluated using the Mean Squared Error (MSE) between the output of the ANN and the average curvature of the reference devices' I_D curve. This curvature, κ, was determined based on V_{DS}, V_{GS}, and V_{BS} using the equation: $\kappa = \frac{\frac{d^2y}{dx^2}}{(1+(\frac{dy}{dx})^2)^{\frac{3}{2}}}$.

For the pretraining, reference devices were chosen from SPICE data, incorporating a diverse range of temperature (T), width (W), and length (L) combinations: (25, 0.5, 2), (125, 0.12, 2), (75, 1, 2), and (50, 1, 1). We specifically selected SPICE MOSFET data with longer channel lengths for training the ANN, showcasing that the ANN's performance is not tied to specific software or MOSFET technology.

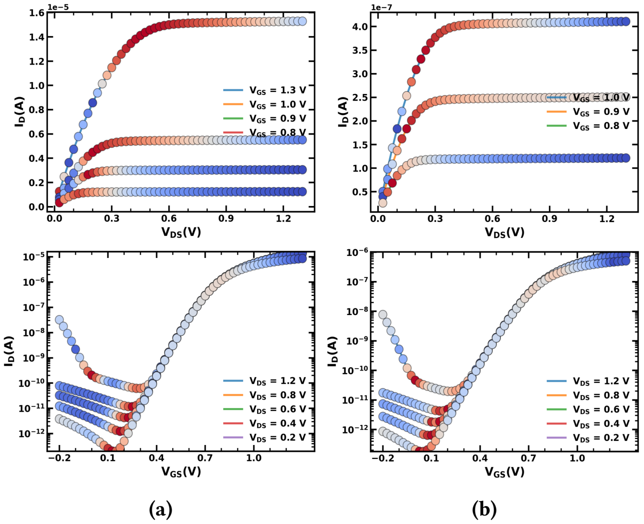
We emphasize that our function complexity estimator, trained using SPICE simulation data, can be directly applied to various TCAD setups without re-training. This demonstrates its adaptability and efficiency in different simulation environments, accurately representing secondary non-linear characteristics in the sampled dataset. This approach is in line with our AL framework's goal to minimize data requirements and simulation time, utilizing readily available SPICE datasets for initial training, thus standing robust for diverse TCAD setups without additional training.

Algorithm 2 Function Complexity Estimator

```
Input: Pretrained neural networks $h_\theta$, $h_\phi$, $g$;
Acquired TCAD data points: $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ where each $x_i$ is an input vector $[V_{DS}, V_{GS}, V_{BS}]$ and $y_i$ is the scalar output $I_D$;
Embeddings: $r_i = h_\theta(\text{concat}(x_i, y_i))$, for $i = 1, \ldots, k$;
Form the matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{k \times d}$: $\mathbf{Q} = \mathbf{K} = \mathbf{V} = [r_1; r_2; \ldots; r_k]$;
Compute attention scores: $\mathbf{S} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})$;
Compute self-attention output: $\mathbf{A} = \mathbf{S}\mathbf{V}$;
Unlabeled sample: $x^*$;
Embedding: $r^* = h_\phi(x^*)$;
Form the query vector for cross-attention: $\mathbf{q}^* = r^* \in \mathbb{R}^{1 \times d}$;
Compute cross-attention scores: $\mathbf{s}^* = \text{softmax}(\frac{\mathbf{q}^*\mathbf{A}^T}{\sqrt{d}})$;
Compute cross-attention output: $\mathbf{a}^* = \mathbf{s}^*\mathbf{A}$;
Predict function complexity at $x^*$ using $g$: $g(\mathbf{a}^*)$;
```

4 EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we compare various baseline methods with our proposed *Aggregated Ranking* method, demonstrating its potential to enhance sampling efficiency.



(a)

(b)

Figure 3: Function Complexity Visualization. Red indicates high and blue low complexity. (a) and (b) show conditions at T=25 °C, W=0.5 μm, L=2 μm and T=125 °C, W=0.12 μm, L=2 μm. Our AL framework prioritizes points with higher I-V curve curvature.

4.1 Baseline Sampling Methods

We begin our evaluation with *Random Sampling* and proceed to analyze the *GP with Matern kernel* and *GP with NTK* [4], highlighting the difference in kernels used in the same GP framework. Following this, we delve into *Monte Carlo (MC) Dropout* [3] and the *Ensemble* method [2], before exploring the properties of *BALD* [14] and *Core-Set* [15]. The GP methods involved optimization of hyperparameters using the gpytorch library with the Adam optimizer for the Matern kernel, while the NTK utilized NCM as the kernel function in the GP. MC Dropout adopted a 0.1 dropout rate and was evaluated through ten forward passes.

We then introduce our *Aggregated Ranking* method, combining the *Expected Variance*, *Simulation Time*, and *Function Complexity* metrics. Our method demonstrated superior performance in our evaluations, yielding the best results with a weight ratio of 5:-1:2 for the respective metrics, with the negative weight indicating a focus on reducing simulation times.

4.2 AL Framework for Device Modeling

Fig. 4 illustrates the total time required for each active learning (AL) iteration, including both TCAD simulation and training time. Fig. 5 depicts the reduction in mean squared error (MSE) as the number of labeled data points increases. While learning-based methods such as the Ensemble [2] can outperform other strategies with a fixed number of data points, they require the time-intensive training of multiple neural networks. Our method distinguishes itself by utilizing a randomly initialized neural network as the kernel. Moreover, it incorporates a one-time pretrained function complexity estimator that is applicable to any TCAD configurations without the need for additional training, thereby offering a significantly more efficient AL framework for time-limited objectives.

Table 2: Evaluation of Sampling Strategies. A comparison of MSE ($\times 10^{-2}$), TCAD simulation and training times, and the number of data points sampled by various strategies under 30 and 60-min wall-clock budgets.

Method	30 min. Budget				60 min. Budget			
	TCAD (sec)	Training (sec)	# Data	MSE	TCAD (sec)	Training (sec)	# Data	MSE
Random	1,772	N/A	205	1.3	3,583	N/A	425	0.35
GP (Matern Kernel)	1,727	1.30	205	1.4	3,543	4.00	425	0.37
BALD [14]	1,566	173	185	1.4	2,862	666	345	0.75
MC Dropout [3]	1,561	174	185	1.3	2,864	669	345	0.72
Ensemble [2]	1,409	361	165	1.2	2,218	1,084	265	0.65
Core-Set [15]	1,750	N/A	205	1.1	3,567	N/A	425	0.40
GP (NTK) [4]	1,739	10.1	205	1.1	3,562	20.9	425	0.35
Ours (Aggregated)	1,752	10.3	205	**0.93**	3,570	21.4	425	**0.31**

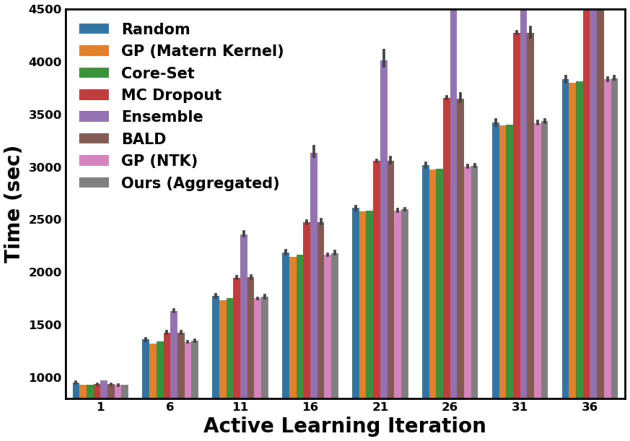


Figure 4: Sampling Methods Evaluation. The total computational time spent in each active learning iteration, encompassing both simulation and training time.

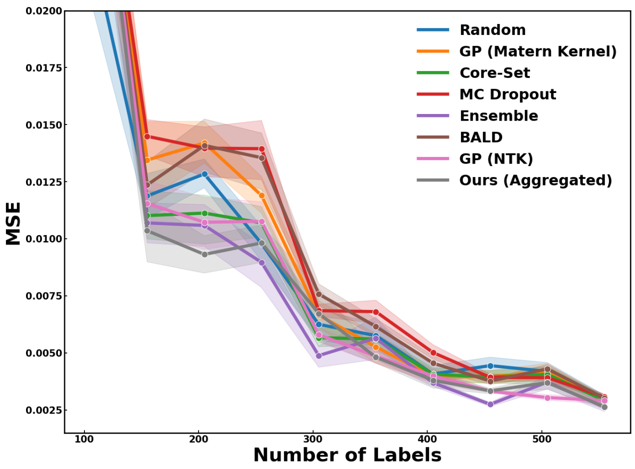


Figure 5: Loss Convergence Comparison. The performance of different methods as the number of labeled data points increases, depicting mean and standard deviation across five random initializations.

Table 2 presents the performance of various methods within 30 and 60-minute time budgets. Methods requiring intensive training struggled to allocate sufficient time for TCAD simulations within these restricted timeframes, leading to less optimal results. In contrast, our method, which does not require further training, excelled in selecting the most informative data points using the proposed metrics. Specifically, our method achieved a target MSE of 0.93×10^{-2} under the 30-minute budget and 0.31×10^{-2} under the 60-minute budget. These figures represent a 28.5% and 11.4% reduction in MSE compared to random sampling under respective time constraints, highlighting the effectiveness of our framework, particularly in scenarios with tighter time budgets.

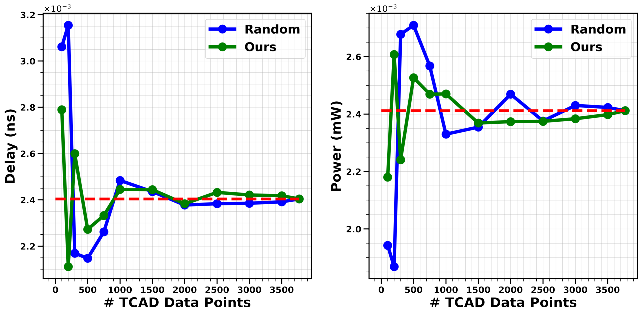
Table 3: TCAD Parameter Ranges for Process Optimization.

Process Variable	Symbol	Unit	Range
Gate length	Lg	nm	25.5-34.5
Spacer length	Lsp	nm	19-21
Oxide thickness	Tox	nm	0.595-0.805
High-K thickness	Thk	nm	1.7-2.3
S/D Doping Conc.	Nsd	cm^{-3}	0.75-1.25
Halo Doping Conc.	Nhalo	cm^{-3}	3.75-6.25
Channel Doping Conc.	Nch	cm^{-3}	0.75-1.25

4.3 AL Framework for Process Optimization

In this section, we apply our AL framework for process optimization on a 51-stage Ring Oscillator (RO) using 32nm single gate devices. The devices are characterized by seven predefined process parameters described in Table 3 [16]. We compare the results obtained using random sampling and our strategy in terms of delay and power. The reference values for delay and power, established using a comprehensive set of 3,770 TCAD data points, serve as the benchmark. The main objective is to explore the accuracy with which our method can approximate the delay and power of the RO with a reduced set of TCAD data points, ranging from 100 to 3,500.

Utilizing the proposed framework, we sampled within the given budget of TCAD data points and subsequently trained a NCM



(a) Delay **(b) Power**

Figure 6: Impact of Data Size on Process Optimization: Using varying TCAD data points, the optimized delay and power of a 51-stage RO are shown. The red dotted line represents reference values from a 3,770 data point set.

Table 4: Process Optimization Results: Performance on a 51-stage RO, highlighting efficacy in scarce data regimes.

# Data	**Delay** ($\times 10^{-10}$ s)		**Power** (mW)	
	Random	**Ours**	**Random**	**Ours**
100	3.06(27.4%)	**2.79(16.0%)**	1.94(19.5%)	**2.18(9.6%)**
200	3.15(31.2%)	**2.11(12.1%)**	1.86(22.5%)	**2.61(8.1%)**
300	2.17(9.7%)	**2.60(8.1%)**	2.67(11.0%)	**2.24(7.1%)**
500	2.15(10.7%)	**2.27(5.5%)**	2.71(12.3%)	**2.52(4.8%)**
750	2.26(5.9%)	**2.33(3.0%)**	2.56(6.5%)	**2.47(2.4%)**
1,000	2.48(3.3%)	**2.45(1.7%)**	2.33(3.4%)	**2.47(2.4%)**
1,500	**2.44(1.3%)**	2.44(1.6%)	2.35(2.4%)	**2.36(1.8%)**
2,000	2.38(1.1%)	**2.38(0.86%)**	2.47(2.4%)	**2.37(1.6%)**
2,500	**2.38(0.86%)**	2.43(1.1%)	**2.37(1.5%)**	2.37(1.5%)
3,000	2.39(0.77%)	**2.42(0.71%)**	**2.43(0.75%)**	2.38(1.1%)
3,500	**2.39(0.52%)**	2.42(0.58%)	**2.42(0.47%)**	2.40(0.57%)

Errors compared to delay, power reference values.

comprising 4 layers and 64 hidden nodes with ELU activation. We employed a learning rate of 3×10^{-3}, which decayed every 50 epochs with a gamma of 0.8. Following the convergence of the training loss, we exported the NCM and executed SPICE simulations to determine the RO's delay and power characteristics.

Fig. 6 shows that both random sampling and our approach progressively approximate the benchmark delay and power values as the number of TCAD data points included in the analysis increases. Nonetheless, as reflected in Table 4, our framework excels over random sampling, especially when operating with fewer than 1,000 data points. To illustrate, it achieves error rates of 5.5% and 4.8% for delay and power respectively, utilizing only 500 out of the 3,770 available data points — a reduction of 86.7%. This finding indicates significant sample efficiency achieved through the incorporation of expected variance and pretrained function complexity.

In this experiment, we analyzed over 140 random process variables, pinpointing the min/max values for power (P) and delay (D) within the sampled data sets. We defined the Figure of Merit (FoM) as a normalized measure of system performance, formulated as:

$$\text{FoM} = \frac{D - D_{\text{min}}}{D_{\text{max}} - D_{\text{min}}} + \frac{P - P_{\text{min}}}{P_{\text{max}} - P_{\text{min}}}$$

Where $D_{\text{min}} = 2.38\times10^{-10}$ s, $D_{\text{max}} = 3.26\times10^{-10}$ s, $P_{\text{min}} = 1.97$ mW, and $P_{\text{max}} = 2.79$ mW.

The optimized FoM was found to be 0.556, corresponding to an optimized delay of 2.40×10^{-10} s and an optimized power of 2.41 mW. These results were obtained with the following process variables: Lg = 25.5 nm, Lsp = 20.0 nm, Nch = 8.75×10^{17} cm^{-3}, Nhalo = 5×10^{18} cm^{-3}.

5 CONCLUSION

This paper introduced an Active Learning (AL) framework that significantly reduces TCAD simulation time in device modeling and process optimization, central to Design Technology Co-Optimization (DTCO). Our approach integrates expected variance using NTK and a function complexity estimator to efficiently identify the most informative simulation points without additional training. The framework notably enhanced device modeling accuracy, achieving a 28.5% improvement in MSE within a 30-minute budget. For process optimization in a 51-stage ring oscillator, it demonstrated an 86.7% reduction in TCAD simulation time, effectively using only 500 out of 3,770 data points. These achievements underscore the framework's efficiency and precision in scenarios where rapid turnaround is crucial, demonstrating its significant potential to enhance DTCO processes.

REFERENCES

[1] Z. Zhang *et al.*, "New-generation design-technology co-optimization (dtco): Machine-learning assisted modeling framework," in *Silicon Nanoelectronics Workshop (SNW)*. IEEE, 2019, pp. 1–2.

[2] W. H. Beluch *et al.*, "The power of ensembles for active learning in image classification," in *Proc. CVPR*, 2018, pp. 9368–9377.

[3] Y. Gal *et al.*, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. ICML*. PMLR, 2016, pp. 1050–1059.

[4] A. Hemachandra *et al.*, "Training-free neural active learning with initialization-robustness guarantees," *ICML*, 2023.

[5] A. Jacot *et al.*, "Neural tangent kernel: Convergence and generalization in neural networks," *NeurIPS*, vol. 31, 2018.

[6] J. Lee *et al.*, "Wide neural networks of any depth evolve as linear models under gradient descent," *NeurIPS*, vol. 32, 2019.

[7] Y. Kim *et al.*, "Physics-augmented neural compact model for emerging device technologies," in *Proc. SISPAD*. IEEE, 2020, pp. 257–260.

[8] X. Gao *et al.*, "Physics-informed graph neural network for circuit compact model development," in *Proc. SISPAD*. IEEE, 2020, pp. 359–362.

[9] C.-T. Tung *et al.*, "Neural network-based and modeling with high accuracy and potential model speed," *IEEE TED*, vol. 69, no. 11, pp. 6476–6479, 2022.

[10] K. Sheelvardhan *et al.*, "Machine learning augmented compact modeling for simultaneous improvement in computational speed and accuracy," *IEEE TED*, 2023.

[11] C. Park *et al.*, "Hierarchical mixture-of-experts approach for neural compact modeling of mosfets," *Solid-State Electronics*, vol. 199, p. 108500, 2023.

[12] Y.-S. Yang, Y. Li, and S. R. Kola, "A physical-based artificial neural networks compact modeling framework for emerging fets," *IEEE TED*, 2023.

[13] Q. Yao *et al.*, "A novel prediction technology of output characteristics for igbt based on compact model and artificial neural networks," *IEEE TED*, 2023.

[14] Y. Gal *et al.*, "Deep bayesian active learning with image data," in *Proc. ICML*. PMLR, 2017, pp. 1183–1192.

[15] O. Sener *et al.*, "Active learning for convolutional neural networks: A core-set approach," *arXiv preprint*, 2017, arXiv:1708.00489.

[16] S. Natarajan *et al.*, "A 32nm logic technology featuring 2nd-generation high-k + metal-gate transistors, enhanced channel strain and 0.171μm2 sram cell size in a 291mb array," *2008 IEEE International Electron Devices Meeting*, pp. 1–3, 2008.