Journal of Computational Electronics (2026) 25:58
https://doi.org/10.1007/s10825-026-02510-x

Inverse design and optimization of heterojunction bipolar transistor epitaxy using active learning

A. N. M. Nafiul Islam[1,3] · Kai H. Kwok[2] · Cristian Cismaru[2]

Received: 15 May 2025 / Accepted: 28 January 2026

Abstract
The design and optimization of epitaxial doping profiles for heterojuction bipolar transistors require expert know-how and extensive trial-and-error using technology computer-aided design (TCAD) simulations. The vastness of the design exploration space along with time-intensive device simulation quickly renders conventional approaches infeasible. In this work, we propose a data-driven inverse design framework based on a surrogate Bayesian neural network model that not only predicts device performance with high accuracy, but also provides uncertainty estimates for each prediction. Leveraging these uncertainties, we implement an active learning scheme to iteratively refine the surrogate model by selecting the most informative new simulation points and thus ensuring maximal performance gain with minimal simulation times. Our results demonstrate that active learning leads to > 45% improvement in network performance over random optimization approaches—paving a promising path to automated and efficient optimization. Finally, the efficacy of the approach is showcased by obtaining candidate epitaxial profiles through the framework that surpass expert-tuned state-of-the-art epitaxial designs for linearity and gain figure-of-merits.

Keywords Heterojunction bipolar transistor (HBT) · Inverse design · Active learning · Technology computer-aided design (TCAD)

1 Introduction

The inverse design and optimization of epitaxial doping profiles that satisfy specific figure-of-merit requirements are an integral part of the HBT design process for semiconductor manufacturing. Conventional methods for such optimization relies on expert-driven knowledge, ingenuity and adjustments followed by extensive TCAD (Technology Computer-Aided Design) simulations or prototype fabrication [1]. Such approaches are time-consuming and may overlook large regions of the design space. This has, therefore, prompted recent research looking at data-driven or machine learning (ML)-based techniques [2–13] to automate inverse design–determining the device doping profiles that yield desired electrical performance.

ML systems, however, require an extensive volume of data to effectively capture the complex physical behaviors of devices and avoid overfitting. As experimental data on plethora of doping profiles is rare, these models generally resort to TCAD simulations [6]. Yet, the data volume required for effective inference can become quite large and prohibitively time-consuming for HBTs. Prior works looking at TCAD-based ML models, although limited, have employed various techniques to address these issues. For example, some approaches require device domain-specific knowledge to limit the exploration space [6], while others eliminate the need for any domain-specific knowledge by leveraging manifold learning [5]. Additionally, physics-informed machine learning models have been proposed that take advantage of known physical equations [10, 14].

✉ A. N. M. Nafiul Islam
nafiul@psu.edu

Kai H. Kwok
Kai.Kwok@skyworksinc.com

Cristian Cismaru
Cristian.Cismaru@skyworksinc.com

[1] School of Electrical Engineering and Computer Science, The Pennsylvania State University, University Park, PA 16802, USA

[2] Advanced Technology Group, Skyworks Solutions, Inc., Newbury Park, CA 91320, USA

[3] Present Address: Inamori School of Engineering, Alfred University, Alfred, NY 14802, USA

Published online: 10 February 2026

Furthermore, studies have looked to improve robustness and avoid overfitting by employing noise during training [7] or by utilizing gaussian processes [11, 12]. Bayesian optimization techniques in particular have garnered a lot of attention in the space in recent times [8, 9, 15]. While some improvement in the convergence has been demonstrated by selecting a more effective stopping criterion [9] or with the use of deep partitioning tree (DPT) in formulating the surrogate model [15], these approaches still suffer from the curse of dimensionality as the number of simulations required for convergence increases significantly with the number of parameters. Recently, a genetic algorithm-based study has shown some promise regarding this [13]. Nevertheless, these efforts across many technologies, while showing various degrees of success, are not commonly scalable and generalizable for all use cases, data and technology.

To address these concerns of scalability and generalization, we propose an inverse design framework with active learning based on a surrogate Bayesian neural network (BNN) model [16]. In active learning (AL) [17], the learning algorithm proactively selects the most informative data points from a pool of unlabeled data for annotation, i.e., simulation with TCAD to improve model performance with minimal labeled data. We study our framework in the context of HBT epitaxial doping profile design and showcase how different AL strategies impact performance of the surrogate model. The efficacy of the framework is finally highlighted by performing inverse design to obtain candidate HBT epitaxial doping profiles that out-perform linearity and gain figure-of-merits of expert designed state-of-the-art (SOTA) epitaxies.

The paper is structured as follows: defining the HBT epitaxial doping profile and figure-of-merit parameters, data generation process, active learning strategies and the Bayesian neural network model. Next, we showcase the model performance and AL gains and model interpretability. Finally, we discuss the complete inverse design framework and the comparison of candidate epitaxial doping profiles with SOTA epitaxial doping profiles.

2 Methodology

2.1 Device profile parameterization and database generation

To build and showcase the inverse design framework, we focus on designing and optimizing of the HBT collector epitaxy profile (i.e., the doping concentration versus depth in the collector region) due to its strong influence on device gain and linearity [18]. The profile is parameterized as a stack of multiple subregions in the collector, characterized by doping concentration values (denoted n_1 through n_m) and thickness values (w_1 through w_{m-1}) as illustrated in Fig. 1a. This parameterization allows a flexible piecewise-constant or non-uniform profile. The doping values are subject to practical fabrication limits—not every arbitrary concentration is achievable—and we enforce a monotonic increase in doping with depth in the rear collector region to ensure favorable breakdown characteristics. Ultimately, we choose a profile defined by six tunable parameters for this study. Nonetheless, the design space defined by these six parameters is extremely large: even with restricted ranges and discrete increments, the total number of possible combinations exceeds 10^{11}, making exhaustive TCAD evaluation infeasible.

In order to build our surrogate model, we first need to generate a dataset of device simulations using design of experiments techniques. We employed Latin Hypercube Sampling (LHS) to draw 5000 samples from the 6-dimensional input space, ensuring a well-spread coverage of the entire range for each parameter. After applying the doping monotonicity constraint (filtering out samples that do not satisfy the requirement) and removing any duplicate points, we obtained 2464 unique collector profiles. For each profile, two TCAD simulation sweeps were performed: (i) an AC simulation to extract the forward transit frequency (f_T) as a function of collector current density (J_C) at a fixed collector voltage, V_C (Fig. 2a), and (ii) another AC simulation to extract maximum available power gain (MAG) or maximum stable gain (MSG), hereafter referred as "gain", across V_C at a fixed J_C (Fig. 2b).

The TCAD simulations were run on Synopsis Sentaurus Device [19] with the following features: (1) hydrodynamic or energy-balance equations enabled to capture non-local carrier transport in the collector due to potentially large electric field caused by certain dopant profiles, (2) calibrated high-field transferred electron mobility model and saturation velocity model enabled to account for inter-valley scattering effects in the collector, (3) a calibrated lumped thermal resistance attached at the bottom of the sub-collector layer

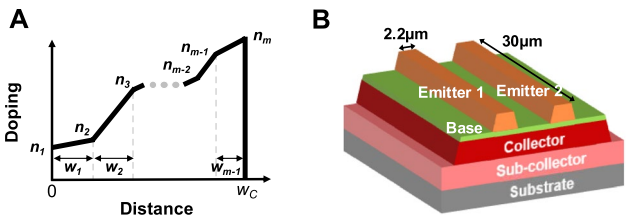

Fig. 1 **a** The HBT collector doping profile is parameterized as a stack of multiple subregions that are denoted by concentration parameters $(n_1, n_2, \ldots, n_m)$ and thickness or depth parameters $(w_1, w_2, \ldots, w_{m-1})$. The collector width, w_C is a fixed value. **b** Simulated HBT device geometry. The device consists of a two-fingered emitter on top of base/collector/sub-collector stack. The fingers have a width of 2.2 µm and length of 30 µm

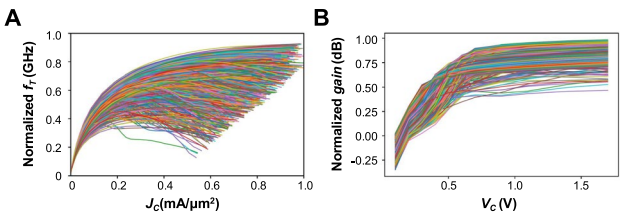

Fig. 2 2464 unique collector doping profiles were simulated with TCAD to generate the initial database for training the surrogate model. **a** AC sweep of the forward transit frequency (f_T) normalized as a function of collector current density (J_C) at a fixed collector voltage, V_C. **b** AC sweep of gain in dB as a function of collector voltage, V_C at a fixed collector current density, J_C

to account for self-heating effects, (4) impact ionization disabled as the highest collector voltage in all simulation runs is much lower than the typical breakdown voltage that can be achieved with the fixed total collector layer thickness chosen for the HBT structure simulated in this study, (5) thermionic emission disabled as no significant conduction band offset at either heterojunction of the transistor was observed when calibrating against measured forward and reverse Gummel plots. The different profiles were constructed with Sentaurus Structure Editor—keeping in mind all practical fabrication limits. The simulated HBT structure is shown in Fig. 1b. Each TCAD run utilized four CPU cores and took approximately 10 min per profile. Thus, the initial dataset required roughly 17 days of computing time in total. It should be noted that simulations for different doping profiles can be performed in parallel to cut down the total time.

From the simulation results, we distilled six output performance metrics for each profile to use as prediction targets. Five of these metrics (L_1 through L_5) quantify the linearity of f_T versus J_C in different current density regions. Specifically, we divide the $f_T(J_C)$ curve into five segments over J_C ranges of 0.1–0.2, 0.2–0.3, 0.3–0.4, 0.4–0.5, and 0.5–0.6 mA/μ m^2 m (in normalized units), and compute L_i as the root-mean-square (RMS) value of the first derivative (slope) of f_T in segment i (using a spline fit to smooth the curve):

$$L_i = \sqrt{\frac{1}{J_{C_{i+1}} - J_{C_i}} \int_{J_{C_i}}^{J_{C_{i+1}}} \left[\frac{\mathrm{d}f_T}{\mathrm{d}J_C}\right]^2 \mathrm{d}J_C} \tag{1}$$

In an ideal linear device, f_T is flat across current, yielding $L_i = 0$ for all segments. The sixth metric gauges the transistor's gain. We measure the area under the *gain* versus V_C curve from 0.7 to 1.7 V and define a parameter A as:

$$A = \int_{0.7}^{1.7} (\text{gain}) \mathrm{d}V_C \tag{2}$$

These six outputs (five f_T-linearity indicators, L_{1-5} and the gain parameter, A) were extracted from every TCAD simulation and form the supervised learning targets in our dataset. The complete initial dataset of 2464 points was then split into training (80%) and validation (20%) subsets for model development. It should be noted that, additional metrics such as breakdown voltages, which are also sensitive to the collector doping profile can be included in this workflow. However, since we have already imposed a monotonicity constraint on the doping profile to favor breakdown performance, the breakdown voltages are omitted in this study for the sake of brevity.

2.2 Active learning strategy for sample efficiency

While a larger training dataset generally improves model performance, generating additional TCAD samples is costly. To maximize the benefit of each new simulation, we incorporated an active learning loop to intelligently select new sample points that would most improve the surrogate model. We prepared a pool of candidate points by again using LHS to sample 5000 new design points across the entire input space (subject to the same constraints). After filtering out any candidates that had already been simulated in the initial set, about 2476 new unique points remained. We then evaluated several query strategies to pick the next 500 points to simulate from this pool:

1. **Random sampling:** 500 points selected at random from the pool (baseline for comparison).
2. **Maximal diversity (distance-based):** 500 points chosen to maximize coverage, e.g., by selecting points farthest from existing training samples in the input space calculated using Euclidean distance (thus filling gaps in the design space).
3. **Expected Error Reduction (EER):** an informativeness-based strategy where points are chosen to minimize the model's expected prediction error over the input distribution. This typically involves selecting points where adding them to the training set is estimated to most reduce overall error [20].
4. **Uncertainty sampling:** points with the highest predictive uncertainty according to the BNN (e.g., highest output variance) are selected, on the assumption that these will be most informative for the model [21].
5. **Expected Improvement (EI):** points that are expected to yield the greatest improvement in a chosen objective or figure-of-merit. Similar to EER, here we select points, adding which yields the most improvement in accuracy [22].

Each strategy produced a set of 500 new design points. We then ran TCAD simulations for those points (5 sets

× 500 = 2500 additional simulations) to obtain their performance metrics. For each augmented dataset (original 2464 plus 500 new points from a given strategy), we retrained the neural network model using the same hyperparameters as the initial training. The improvement in model performance was evaluated by the reduction in validation loss relative to the initial model (and relative to the random-sampling case). This procedure allows us to directly compare the efficacy of different active learning criteria in enhancing the surrogate model.

2.3 Bayesian neural network model

For training a neural network-based predictive model, we chose a Bayesian neural network (BNN). In comparison to other architectures, BNNs give a way for quantifying model uncertainty, which is pivotal for model interpretability. We map the collector profile inputs to the six performance metrics with a BNN as our surrogate model. The network has an input layer of size 6 (accepting $[n_1, n_2, n_3, n_4, w_1, w_2]$) and an output layer of size 6 (predicting $[L_1 - L_5, A]$). Between these, we employed two hidden layers with 400 neurons each, using ReLU activation functions. The BNN is implemented with Bayesian layers such that the weights are treated as random variables with a prior distribution; in practice, we utilized the TorchBNN library [23] on top of PyTorch [24] to realize this architecture (Fig. 3a). All input features and output targets were normalized to zero mean and unit variance prior to training. Given the relatively small size of the dataset, preventing overfitting was a top priority. We incorporated several regularization techniques: dropout layers [25] and batch normalization [26] in the network, and early stopping based on validation loss [27]. The BNN was trained for up to 5000 epochs using the Adam optimizer [28] with an initial learning rate of 0.01. The loss function $\mathcal{L}$ was defined as a weighted sum of a mean-squared-error (MSE) term and a Kullback–Leibler (KL) divergence term [29]:

$$\mathcal{L} = \mathrm{MSE}(Y_{\mathrm{pred}}, Y_{\mathrm{true}}) + \lambda D_{KL}(q(\mathbf{w})||p(\mathbf{w})) \tag{3}$$

where $q(\mathbf{w})$ is the learned posterior weight distribution and $p(\mathbf{w})$ the prior; D_{KL} measures how much the posterior deviates from the prior. We set the weighting factor λ such that it encourages the network to find a balance between fitting the data and staying close to the prior (thus controlling complexity). The KL term effectively acts as a regularizer reflecting model uncertainty. This loss was minimized via backpropagation. Training proceeded until convergence, with Fig. 3b showing an example of the training and validation loss curves over epochs; the early stopping criterion halted training once the validation loss stopped improving to avoid overfitting. Table 1 shows the network training hyperparameters.

As previously mentioned, a key advantage of the BNN is that its predictions come with uncertainty estimates. At inference time, instead of outputting a single point estimate, the BNN yields a probability distribution for each output. We obtained predictive means and variances by performing multiple stochastic forward passes (100 Monte Carlo samples of the network's weights) for each input and computing the mean and standard deviation of the outputs. This approach (sometimes called Monte Carlo sampling for Bayesian inference) gives us a measure of confidence in the prediction for each of the six metrics. For instance, if the model is unsure about a particular region of the input space, the predicted output for that region will have a larger standard deviation. We leveraged this capability extensively in the active learning process by directing new samples to regions of high uncertainty.

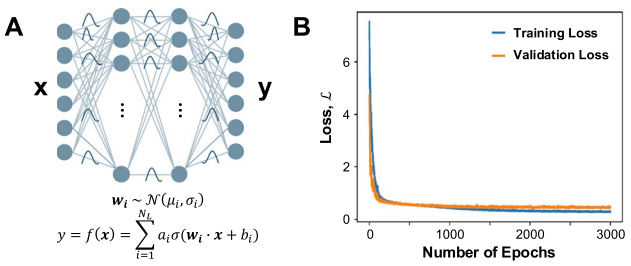

Fig. 3 **a** The surrogate Bayesian neural network model used for prediction. The input and output layer consists of 6 neurons representing the input parameters and output metrics with two hidden layers with 400 neurons each. The network weights are characterized by random variables with a normal distribution which are learnt through training. **b** The training and validation loss over the number of training epochs shows convergence. Regularization techniques such as dropout, batch normalization and early stopping were used to prevent overfitting

Table 1 Training hyperparameters

Parameter	Value
Network architecture	6, 400, 400, 6
Optimizer	Adam
Learning rate	0.01
Number of epochs	5000
Early stopping patience	100
Weight decay	1e-5
Dropout rate	0.5
Activation function	ReLU
Train/validation split	80%/20%
Weighting factor, λ	0.2

3 Results and discussion

3.1 Baseline model performance

The trained BNN surrogate achieved reasonable accuracy in predicting the HBT metrics on the validation dataset. Figure 4 compares the BNN-predicted values of the six outputs to the true values from TCAD for the validation set, shown against the ideal $y = x$ line (dotted black line). The results indicate that the model has reasonably learned the complex relationships between the collector doping profile and performance metrics. The handful of points showing large deviations can be primarily attributed to the model not being able to capture outliers in the limited dataset.

3.2 Active learning gains

Incorporating active learning proved highly beneficial in improving the surrogate model (Fig. 5a). We evaluated the performance of the model after augmenting the training set with 500 new points selected by each strategy. Using random sampling as a baseline, the validation loss decreased by about 8% relative to the initial model (simply owing to more training data). Other strategies achieved greater improvements, demonstrating the value of informed sample selection as can be seen in Fig. 5b. In particular, the Bayesian uncertainty strategy (selecting points where the BNN was most unsure) yielded the largest performance gain: the validation loss dropped by > 50% compared to the initial model, which is a 46% improvement over the random-sampling case. In other words, the model retrained with uncertainty-guided samples was significantly more accurate, approaching the TCAD ground truth behavior much closer than the model retrained on an equal number of randomly chosen samples. The EER and distance-based selection strategies also outperformed random sampling, though not to the extent of the uncertainty-based method; they provided moderate reduction in error, indicating that both exploring sparsely sampled regions and directly aiming to reduce expected error are useful but perhaps less targeted than using the model's full Bayesian insight. The expected improvement strategy, which in our context attempted to sample profiles that might improve certain device performance metrics, did not improve the surrogate model's accuracy as effectively—likely because it focuses on extremes of the performance space rather than the regions where the model lacks knowledge. These findings highlight that when the goal is to globally learn the input–output mapping (rather than optimize a specific device metric), querying by model uncertainty is an excellent criterion.

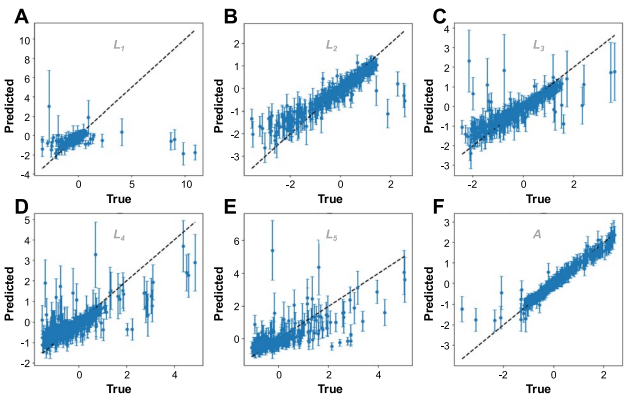

Fig. 4 BNN-predicted values versus the true values obtained from TCAD for the validation set for **a**–**e** the linear metrics, L_1–L_5 and **f** gain metric, A. *Note*: the values are normalized to zero mean and unit variance like the training set prior to inference. The error bars represent the mean and one standard deviation. The black dotted line corresponds to the ideal case where prediction is equal to true values. It can be seen that the predicted values are generally close to their true values and points that deviate also have a greater uncertainty, i.e., larger standard deviations

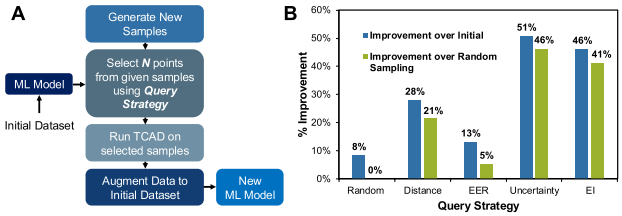

Fig. 5 **a** Setup for evaluating the improvement of performance due to active learning strategies with respect to the initial model. **b** Percentage improvement over the initial model and the model augmented with the active learning strategy of random sampling. We find that Uncertainty sampling and Expected Improvement yield the largest relative improvement, 46% and 41%, respectively, in comparison to the random-sampling query strategy

4 Inverse design framework and evaluation

By combining all the components discussed thus far, we can now design our complete inverse design framework with active learning in place. As seen in Fig. 6, the framework is divided into two blocks: one for iteratively training the model and the other for extracting the inverse-designed device parameters for testing.

(1) **Iterative Model Training Block:** This module aims to refine the surrogate model in a data-efficient way. Initially, a seed dataset is collected by running TCAD simulations. The BNN surrogate is then trained on this dataset to predict the device performance metrics (e.g., gain and linearity parameters). To improve accuracy without having to run a prohibitively large number of simulations, we adopt AL

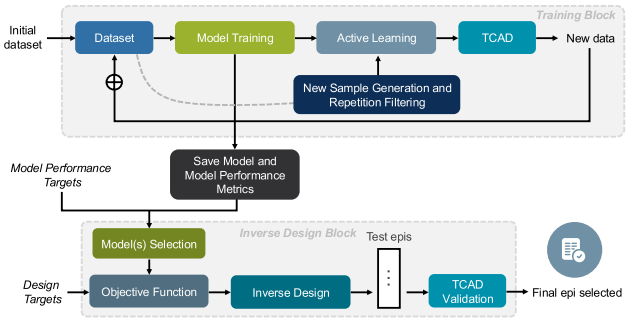

Fig. 6 Complete proposed machine learning-driven TCAD framework for inverse design and optimization with active learning. The framework consists of two blocks: a training block which iteratively improves the surrogate model in a data-efficient way through active learning, and an inverse design block that spits out test epitaxial doping profiles (referred as "test epis" in figure) that satisfy design targets to accelerate the design process

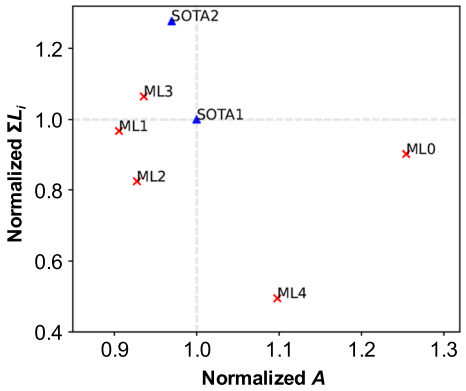

Fig. 7 Summed linearity metrics versus gain metric obtained from five candidate doping profiles, ML1 through ML5, along with two state-of-the-art doping profiles used industrially by Skyworks Solutions Inc., referred as SOTA1 and SOTA2. The values were normalized with respect to SOTA1 values

that selects new, highly informative points from the design space (we chose uncertainty sampling here) for subsequent TCAD simulations. The newly generated simulation results are added to the training set, and the BNN is retrained. This iterative cycle helps converge on a surrogate model that covers the design space more effectively than conventional random sampling.

(2) **Inverse Design Block:** Once the BNN has been trained to a desired accuracy, it serves as a computationally lightweight proxy for the full TCAD simulator. We then solve the inverse design problem: given a target set of performance metrics (e.g., minimal nonlinearity in the transit frequency f_T versus current density J_C, and a desired area under curve of gain versus V_C, what collector epitaxial parameters (n_i's, w_i's) will yield those metrics? We formulate a cost function comparing the BNN-predicted metrics $\hat{y}$ to the desired targets y_{target}. By minimizing this cost function with respect to the input parameters, we identify candidate device profiles that should achieve the desired performance. For our case, we employ the Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS) [30]-based minimization implemented with SciPy [31].

Because the BNN is orders of magnitude faster than TCAD, we can quickly explore many potential designs. The final step is to select the most promising profiles from the optimization output and verify them with an actual TCAD simulation. This closes the loop, ensuring that the final recommended profiles truly match or exceed the target specifications.

4.1 Evaluation with state-of-the-art epitaxies

To demonstrate the effectiveness of the framework, we conducted a proof-of-concept study on inverse designing collector profiles for HBTs. As described in earlier sections, we first trained a BNN that was subsequently refined with AL. Once the BNN's predictive accuracy stabilized (i.e., validation losses improved marginally with new data), we used the model for inverse design. We specified target values for the linearity and gain metrics—aiming to achieve lower-than-nominal nonlinearity across the entire collector current range while maintaining or even boosting the transistor gain. Using the optimizer, we generated candidate solutions in the 6D design parameter space.

We generated 10 candidate parameter sets and ran full TCAD simulations on them to extract the performance metrics, $L_1 - L_5$ and A. The candidates are then compared with two state-of-the-art doping profiles utilized industrially by Skyworks Solutions Inc. that were expert designed and have great linearity and gain performance. We combine the five linearity metrics into a singular metric, $\sum_{i=0}^{5} L_i$—signifying the summation of the different slope components across the f_T-versus-J_C curve. Thus, we would want our new candidate profiles to have a lower $\sum_{i=0}^{5} L_i$. Now that we have shrunk the number of metrics to two, we can effectively visualize the results using a simple 2D plot. Figure 7 shows the metrics for the top five candidates, referred as ML1 through ML5, normalized to the corresponding values obtained from one of the SOTA doping profiles (referred as SOTA1). Here, the region of interest is the fourth quadrant, where candidates will not only have higher gain relative to the reference but will also have a lower $\sum_{i=0}^{5} L_i$ indicating improved linearity over the f_T -versus-J_C curve. Encouragingly, we do find two candidate doping profiles within the fourth quadrant that show better linearity and gain performance than the two reference doping profiles. These results demonstrate that an actively

trained BNN combined with an inverse design optimization step can substantially expedite the design process and achieve similar or better performance to expert-tuned SOTA epitaxial designs.

5 Conclusion

We presented an inverse design and optimization framework using active learning strategies in the context of HBT epitaxy design. The iterative training approach ensures maximal accuracy gain with minimal samples, while the flexible inverse design procedure allows us to obtain many candidate profiles by tailoring the objective. For the future work, the active learning framework can be expanded to encompass the inverse design objective function to ensure more focused improvements, rather than global improvements, that can lead to even faster explorations. Thus, this work has the potential to transform HBT development by automating design tasks, reducing simulation times, and driving innovation in advanced HBT technologies. In fact, we believe the scalability, adaptability and generalization capability of the framework can make it ideal for any advanced technology development framework beyond HBTs.

Author contributions ANMNI, KHK and CC contributed to the study conception. ANMNI conducted the TCAD data collection, model training and inverse design analysis. KHK and CC supervised the project. All authors contributed to the writeup of the manuscript. All authors read and approved the final manuscript.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability All data needed to evaluate the conclusions in the paper are present in the paper.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. Rosenbaum, T., Saxod, O., Vu, V., Celi, D., Chevalier, P., Schröter, M., Maneux, C.: Calibration of 1D doping profiles of SiGe HBTs. In: 2015 IEEE Bipolar/BiCMOS Circuits and Technology Meeting-BCTM, pp. 64–67. IEEE (2015)
2. Ferguson, R., Roulston, D.J.: Artificial neural networks for reverse engineering bipolar transistors. In: 1997 21st International Conference on Microelectronics. Proceedings, vol. 2, pp. 459–462. IEEE (1997)
3. Peurifoy, J., Shen, Y., Jing, L., Yang, Y., Cano-Renteria, F., DeLacy, B.G., Joannopoulos, J.D., Tegmark, M., Soljačić, M.: Nanophotonic particle simulation and inverse design using artificial neural networks. Sci. Adv. **4**(6), 4206 (2018)
4. Bankapalli, Y., Wong, H.: TCAD augmented machine learning for semiconductor device failure troubleshooting and reverse engineering. In: 2019 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), pp. 1–4. IEEE (2019)
5. Mehta, K., Wong, H.-Y.: Prediction of FinFET current-voltage and capacitance-voltage curves using machine learning with autoencoder. IEEE Electron Device Lett. **42**(2), 136–139 (2020)
6. Caron, G., Juditsky, A., Guitard, N., Céli, D.: Recovery of intrinsic heterojunction bipolar transistors profiles by neural networks. In: 2022 IEEE BiCMOS and Compound Semiconductor Integrated Circuits and Technology Symposium (BCICTS), pp. 228–231. IEEE (2022)
7. Raju, S.S., Wang, B., Mehta, K., Xiao, M., Zhang, Y., Wong, H.-Y.: Application of noise to avoid overfitting in TCAD augmented machine learning. In: 2020 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), pp. 351–354. IEEE (2020)
8. Heimerl, J.P., Teng, J.W., Lee, H.P., Nergui, D., Moody, J.P., Cressler, J.D.: TCAD-based design of SiGe HBT germanium profiles via bayesian optimization. In: 2024 IEEE BiCMOS and Compound Semiconductor Integrated Circuits and Technology Symposium (BCICTS), pp. 278–281. IEEE (2024)
9. Kim, B., Shin, M.: Bayesian optimization of MOSFET devices using effective stopping condition. IEEE Access **9**, 108480–108494 (2021)
10. Jie, X., Wang, J., Ouyang, X., Zhuang, Y., Wang, Z., You, S., Wang, D., Yu, Z.: Characteristics prediction and optimization of InP HBT using machine learning. J. Comput. Electron. **23**(2), 305–313 (2024)
11. Jaiswal, R., Martínez-Ramón, M., Busani, T.: Probabilistic analysis of solar cell performance using gaussian processes. IEEE J. Photovoltaics **12**(2), 652–658 (2022)
12. Jaiswal, R., Martínez-Ramón, M., Busani, T.: A semi-empirical approach to calibrate simulation models for semiconductor devices. Sci. Rep. **13**(1), 10436 (2023)
13. Kim, S., Min, S.-J., Jung, S.-G., Yu, H.-Y.: Multi-objective optimization and inverse design of complementary field-effect transistor using combined approach of machine learning and non-dominated sorting genetic algorithms for next-generation semiconductor devices. Eng. Appl. Artif. Intell. **137**, 109064 (2024)
14. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L.: Physics-informed machine learning. Nat. Rev. Phys. **3**(6), 422–440 (2021)
15. Torun, H.M., Swaminathan, M.: High-dimensional global optimization method for high-frequency electronic design. IEEE Trans. Microw. Theory Tech. **67**(6), 2128–2142 (2019)
16. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: International Conference on Machine Learning, pp. 1613–1622. PMLR (2015)

17. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. Advances in neural information processing systems 7 (1994)
18. Iwamoto, M., Asbeck, P.M., Low, T.S., Hutchinson, C.P., Scott, J.B., Cognata, A., Qin, X., Camnitz, L.H., D'Avanzo, D.C.: Linearity characteristics of GaAs HBTs and the influence of collector design. IEEE Trans. Microw. Theory Tech. **48**(12), 2377–2388 (2000)
19. https://www.synopsys.com/manufacturing/tcad/device-simulation/sentaurus-device.html
20. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: ICML, vol. 1, p. 5. Citeseer (2001)
21. Lewis, D.D.: A sequential algorithm for training text classifiers: corrigendum and additional data. In: ACM Sigir Forum, vol. 29, pp. 13–19. ACM, New York (1995)
22. Eggensperger, K., Feurer, M., Hutter, F., Bergstra, J., Snoek, J., Hoos, H., Leyton-Brown, K., et al.: Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In: NIPS Workshop on Bayesian Optimization in Theory and Practice, vol. 10, pp. 1–5 (2013)
23. Lee, S., Kim, H., Lee, J.: Graddiv: Adversarial robustness of randomized neural networks via gradient diversity regularization. IEEE Trans. Pattern Anal. Mach. Intell. (2022)
24. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32. Curran Associates Inc, New York, pp. 8024–8035 (2019). http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
25. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
26. Ioffe, S.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
27. Yao, Y., Rosasco, L., Caponnetto, A.: On early stopping in gradient descent learning. Constr. Approx. **26**(2), 289–315 (2007)
28. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization (2017). arXiv:abs/1412.6980
29. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22**(1), 79–86 (1951)
30. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**(1), 503–528 (1989)
31. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ, Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P.: SciPy 1.0 contributors SciPy 1.0 fundamental algorithms for scientific computing in python. Nat. Methods **17**, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2