Bayesian Optimization of MOSFET Devices Using Effective Stopping Condition

BOKYEOM KIM AND MINCHEOL SHIN

School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea

Corresponding author: Mincheol Shin (mshin@kaist.ac.kr)

ABSTRACT Current nanometer-scale metal-oxide-semiconductor field-effect transistor (MOSFET) devices exhibit short-channel, quantum, and self-heating effects, making modeling and analysis very complex. A few recent works have employed machine-learning (ML) techniques and neural networks (NN) to model the complex relationships and optimize devices, but a problem with the NN-based device optimization is that it is data-intensive. Bayesian optimization (BO) can realize ML-based data-efficient optimization of the MOSFET device, as it finds the global optimum while requiring few training data. BO stops theoretically when every candidate is explored, so previous works used a fixed number of iterations for the stopping condition. Such an empirical stopping condition is detrimental to the efficiency and reliability of BO, because the global optimum can be found at an earlier stage or even after stopping. Recently, maximum expected improvement (EI_{max}) with a tiny constant has been proposed as a stopping condition for BO. However, there have not been sufficient works for improving efficiency of BO. By advancing the EI_{max} scheme, we have systemically investigated the effective stopping condition (ESC) for BO of MOSFET devices to boost the efficiency and reliability of optimization. We found that EI_{max} less than a 1% of unit value was an efficient and reliable ESC for optimization, which resulted in up-to-87.6% and up-to-47% reductions of required training data compared with the fixed iteration method and the tiny constant method, respectively. Our study provides a novel method to boost efficiency and reliability of BOs for the optimization of MOSFET design in the semiconductor industry.

INDEX TERMS Design optimization, machine learning, metal-oxide-semiconductor field-effect transistor, optimization, nanosheet FET.

I. INTRODUCTION

Modern technology has realized palm-sized computers that have the power and speed of the supercomputers of 1990s, which was enabled by performance boost, low power consumption, and miniaturization of integrated circuits (IC). As a building block of ICs, the size of metal-oxide-semiconductor field-effect transistor (MOSFET) has been continuously miniaturized and its performance has improved following the Moore's law [1]. However, aggressive downscaling has weakened their gate controllability owing to short-channel effects (SCE). Furthermore, the requirements for performance and power consumption have become stricter to meet industrial standards. To minimize SCE multi-gate structures

(e.g., finFETs, nanosheet FETs (NS FET)) have been proposed [2]–[8].

Nanometer-scale MOSFETs have become more complicated, because, in addition to SCE, they exhibit quantum mechanical effects (QME), self-heating effects (SHE), and parasitic capacitance [6], [9]. The correlation between design variables has thus increased, and their relationships with device characteristics can be nonlinear [10]–[13], making device analysis and optimization much more challenging than ever.

Recently, machine learning (ML)-based device optimization frameworks have been proposed to handle the complexity of the optimization of next-generation MOSFET devices [13]–[15]. ML can model the relationships between the design variables and device characteristics based on the data. In [13], a neural network (NN) was trained to model

---

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Abdur Razzaque.

the relationship between seven design parameters, including oxide thickness (t_{ox}), gate length (L_g), and device characteristics, such as threshold voltage (V_{TH}) and ON-state current (I_{ON}) by utilizing 1,055 technology computer-aided design (TCAD) simulations of feedback FETs. They optimized the feedback FETs by analyzing the gradient of the trained NN. In [14] and [15], the NN was trained using 1,510 TCAD simulations of lateral double-diffused MOS. The device was optimized by combining the NN with a Bayesian optimizer in a 4-dimensional (4D) design space. These NN-based optimization frameworks were data-intensive, requiring large training data ($\geq$1,000). As the optimization dimensions increase, the number of training data required to build the NN increases rapidly with the number of independent design variables.

By combining TCAD simulations with Bayesian optimization (BO), device optimization can be conducted in a data-efficient manner. BO is an ML-based global optimization algorithm that models and optimizes expensive black-box functions by sampling the training data in an efficient way [16], [17]. Although there are many optimization algorithms for black-box functions, such as COBYLA [18], ISRES [19], and DIRECT [20], only BO is designed to be data-efficient, because it samples the next observation point in an adaptive manner at each optimization step (iteration) [17]. Each iteration consists of evaluations of the black-box function and the decision of the next sampling point.

Successful applications of BO in electronics have been reported for the design of 3D ICs and analog circuits [21], [22]. The BO showed faster convergence to the global optimum in 3D IC optimization than other optimization algorithms, including as pattern search and nonlinear solver [21], [23]. The BO could optimize the lateral double-diffused MOS within 90 iterations [14]. These works shed light on data-efficient optimization in electronics, because BOs require a small number of training data compared with the total number of design candidates.

However, previous works used finite numbers of iterations (N_{iter}) as the BO stopping conditions. In [21], an N_{iter} of 200 was used, and in [22] an N_{iter} of 100 was used as the stopping criterion. Using N_{iter} as the stopping condition is too empirical, unreliable and inefficient, because an assumption must be made in which the BO finds the global optimum within the N_{iter}. It is possible that the global optimum can be found at an iteration much larger than N_{iter}, which impairs the reliability of BO. It is also possible that the global optimum can be found at an earlier iteration much smaller than N_{iter}, which impairs efficiency. In [14], the squared error from the target value was set as the stopping condition. However, if this stopping condition is not feasible in the design space, the optimization never ends.

To improve the reliability and efficiency of BO, the stopping condition must be studied, but there have been insufficient examinations of BO stopping conditions [24]–[27]. In [24], the maximum expected improvement over the best-observed value (EI_{max}) was proposed. In the work, the sublinear convergence of EI_{max} was proven, and the

optimization process was completed when $EI_{max} < \kappa$, where κ was a small positive constant (e.g., 10^{-9}). However, this method can be harmful for the efficiency of BO because the global optimum can be found when EI_{max} is much larger than a tiny constant. A recent study used an EI_{max} of 0.0015 as the stopping condition, but there was no justification for the value [28]. Although there have been some works about the stopping condition of BO, there have not been sufficient works for boosting the efficiency of BO while not impairing the robustness of optimization.

In this work, we investigate the effective stopping condition (ESC) to maximize the BO efficiency and reliability based on EI_{max} scheme. After constructing an ML-based MOSFET device optimization framework, we explore the ESC by calculating 2,800 single-gate (SG) n-type MOSFET (nMOSFET) devices in a 5D input space. Multi-objective optimization is then conducted by introducing the target function. As an application of our BO framework to a highly-complex, computationally demanding yet practical problem, we optimized a 3-nm node NS FET considering QME, SCE, and SHE using a 3D TCAD simulation. NS FET is one of the most promising devices for the downscaling of silicon-based CMOS technologies and have displayed performance superior to that of FinFETs [2]–[7].

We organized this paper as follows. In Section II, we propose our framework with detailed explanations about Bayeisian optimization, TCAD simulation, and multi-objective optimization. In Section III, we show the efficiency and robustness of ESC based on SG nMOSFET optimization. Then, we provide optimization results of NS FET as an application of ESC. We conclude in Section IV.
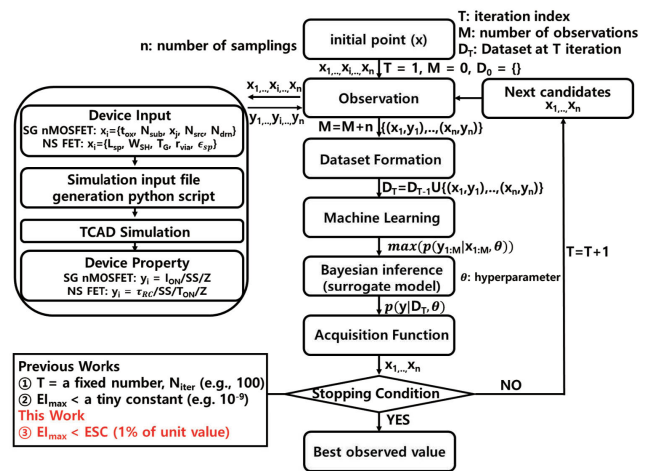

FIGURE 1. BO-based MOSFET optimization framework. BO is directly combined with TCAD simulation. Novelty of this work is highlighted with red letters. Optimization stops when EI_{max} is lower than ESC.

II. METHODS

The overall BO-based MOSFET device optimization framework constructed in this study is shown in Fig. 1. The TCAD simulation is directly combined with the BO. Device input

vector of selected device (x_i in Fig. 1) is converted to an input file of TCAD simulation by our in-house python script. For SG nMOSFET optimization, input vector consists of t_{ox}, substrate doping density (N_{sub}), junction depth (x_j), and source/drain doping density (N_{src}/N_{drn}). For NS FET optimization, input vector is composed of spacer length (L_{sp}), NS width (W_{SH}), gate thickness (T_G), via radius (r_{via}), and spacer dielectric constant (ϵ_{sp}). Each element of input vectors is standardized with a buit-in function of python library [29]. The output of the TCAD simulation (y_i in Fig. 1) is paired with the input vector, i.e., (x_i,y_i). For the optimization of SG nMOSFET, output scalar is I_{ON}, subthreshold slope (SS), or target function **Z**. The target function **Z** is utilized to carry our the multi-objective optimization. In NS FET optimization, output scalar is RC delay (τ_{RC}), SS, ON-state maximum temperature (T_{ON}), or **Z**. The sampling set, $\{(x_1, y_1), \dots, (x_n, y_n)\}$, is constructed with n pairs of (x_i, y_i) when n samples are selected by BO. Then, the training data set at T-1 iteration is augmented with the sampling data set ($D_T = D_{T-1} \cup \{(x_1, y_1), \dots, (x_n, y_n)\}$ in Fig. 1), where T denotes the iteration index.

The surrogate model of BO is learned using the training data set, and next simulation points ($x_1, \dots, x_n$ in Fig. 1) are selected in a global-optimum direction by evaluating the acquisition function. The optimal device can then be found by repeating this automatic feedback process between the BO and TCAD simulations. The BO stops when the stopping condition is met. The novelty of our framework is highlighted with the red letters. Unlike previous works, which stop BO when T becomes a fixed number, N_{iter}, or when EI_{max} is smaller than a tiny positive constant (e.g., 10^{-9}), our framework stops BO when EI_{max} is lower than 1% of unit value. As illustrated in Section I, N_{iter} as a stopping condition can impair the efficiency and robustness of BO since the global optimum can be found at $T \ll N_{iter}$ or $T > N_{iter}$. EI_{max} of a tiny positive constant as a stopping condition can be detrimental to the efficiency of BO because it is possible for BO to reach the global optimum when EI_{max} is much larger than a tiny value such as 10^{-9}. Therefore, huge redundant BO iterations can be generated to meet the stopping criterion, resulting in the inefficiency of BO. The inefficiency of a tiny positive constant method and efficiency of our ESC method are illustrated in Section III-A. Detailed explanation of BO and TCAD simulation follows.

A. BAYESIAN OPTIMIZATION

BO is a ML-based experimental design algorithm for finding the global optimal point of the black-box objective function. Mathematically it is expressed as:

$$x^* = \arg\max_{x \in \mathcal{X}} f(x), \tag{1}$$

where $f(x)$ is the objective black-box function, and x is the D-dimensional input vector comprising the design variables. D is the number of design variables to be optimized. $\mathcal{X}$ is a design space of interest (i.e., a compact subset of R^D), and

---

Algorithm 1 Bayesian Optimization

Require: Number of initial sampling points N_{init}, number of iterations N_{iter}, and number of samplings at each iteration N_{sp}
1: Randomly sample N_{init} input points in the design space: $x_{1:N_{init}}$
2: Number of observations M = 0
3: Choose Acquisition function $\alpha(x)$
4: **for** iteration T = 1 to N_{iter} **do**
5:     **if** T == 1 **then**
6:         Observe $y_{1:N_{init}}$ corresponding to $x_{1:N_{init}}$
7:         M $+= N_{init}$
8:         Construct the data set $\mathbf{D_T} = \mathbf{D_{1:M}} = \{(x_i, y_i)\}_{i=1}^{N_{init}}$
9:     **else**
10:         Observe $y_{1:N_{sp}}$ corresponding to $x_{1:N_{sp}}$
11:         M $+= N_{sp}$
12:         $\mathbf{D_T} = \mathbf{D_{1:M}} = \mathbf{D_{T-1}} \cup \{(x_i, y_i)\}_{i=1}^{N_{sp}}$
13:     **end if**
14:     Fit probabilistic surrogate model to $\mathbf{D_T}$
15:     Get unexplored input set $\mathcal{X}' = \mathcal{X} \cap \{\{x_i\}_{i=1}^M\}^c$
16:     Set $x_{1:N_{sp}} = \varnothing$
17:     **for** s = 1 to N_{sp} **do**
18:         $x_s = \arg\max_{x \in \mathcal{X}'} \alpha_T(x)$
19:         $x_{1:N_{sp}} = x_{1:N_{sp}} \cup \{x_s\}$
20:         $\mathcal{X}' = \mathcal{X} \cap \{x_s\}^c$
21:     **end for**
22:     Get next sampling point set $x_{1:N_{sp}}$
23: **end for**
24: **return** $y^* = \max y_{1:M}$ and x^* (input of y^*)

---

x^* is the global optimizer. The main assumption is that $f(x)$ does not have a closed form but can be evaluated by observation, y [16]. BO also assumes that y is noisy, so that the black-box function, $f(x)$, is hindered by observation noise ϵ. The stochastic nature of y is expressed as $y = f(x) + \epsilon$, where $\epsilon \sim N(0, \sigma_n^2)$. σ_n^2 is the observation variance. BO is based on Bayes' theorem:

$$p(y|\mathbf{D_{1:M}}, \theta) = \frac{p(\mathbf{D_{1:M}}|y, \theta)p(y|\theta)}{p(\mathbf{D_{1:M}}|\theta)}, \tag{2}$$

where $\mathbf{D_{1:M}} = \{(x_1,y_1), (x_2,y_2), \dots, (x_M,y_M)\}$ is the data set collected after **M** observations. When hyperparameter (θ) is given, $p(y|\theta)$, $p(y|\mathbf{D_{1:M}}, \theta)$, and $p(\mathbf{D_{1:M}}|y, \theta)$ are the distributions of the prior, posterior, and likelihood, respectively [30]. As shown in Algorithm 1, BO starts with $\mathbf{N_{init}}$ observations, where $\mathbf{N_{init}}$ is the number of initial sampling points. Then, the iteration consisting of surrogate model training and the selection of $\mathbf{N_{sp}}$ next sampling points is repeated until T = $\mathbf{N_{iter}}$. $\mathbf{N_{sp}}$ is the number of samplings at each iteration. The surrogate model is trained using hyperparameter tuning, which will be explained later. Most previous works used finite $\mathbf{N_{iter}}$ as the stopping condition of BO. For example, $\mathbf{N_{iter}} = 200$ [21], 100 [22], and 10D [24]. There is no rule for $\mathbf{N_{init}}$. Thus, $\mathbf{N_{init}}$ can be 100 [22] or 3D [24]. To prevent frequent

training of the surrogate model, N_{sp} can be greater than 1, for instance $N_{sp} = 4$ or $N_{sp} = 20$ [22], [31].

BO has two key components: a surrogate model and an acquisition function. The surrogate model is the posterior distribution of the objective function. The acquisition function evaluates the unexplored data points and selects the next query points. In this work, we utilized two types of surrogate model: Gaussian process regression (GPR) [14] and Bayesian linear regression (BLR) [29]. BLR can sometimes be faster than GPR if the optimization requires large training data [32]. Because we do not know in advance the number of training data required for device optimization, we try both and evaluate their suitability for MOSFET device optimization. For an acquisition function, we choose the expected improvement (EI), because it can be used to define the stopping criterion [24]. We use the open-source BO Python library, COMBO [29], to construct our framework. A detailed explanation of the surrogate model and acquisition function is as follows.

1) GAUSSIAN PROCESS REGRESSION

GPR is one of the most typical surrogate models of BO. When M points are observed, $f(x_{1:M})$ is assumed to be jointly Gaussian, and the observation, $y_{1:M}$, is assumed to be normally distributed, given $f(x_{1:M})$ [16]:

$$f(x_{1:M}) \sim N(\mu(x_{1:M}), K(x_{1:M})), \quad (3)$$

$$y_{1:M} \sim N(f(x_{1:M}), \sigma_n^2 I), \quad (4)$$

where $\mu(x_{1:M})$ is the mean function and $K(x_{1:M})$ is the covariance matrix, consisting of kernel functions $k(x_i, x_j)$. I is the identity matrix having a size of M. $\mu(x_{1:M})$ and $K(x_{1:M})$ are expressed as:

$$\mu(x_{1:M}) = [\mu(x_1), \ldots, \mu(x_M)]^T, \quad (5)$$

$$K(x_{1:M}) = \begin{pmatrix} k(x_1, x_1) & \ldots & k(x_1, x_M) \\ \vdots & \ddots & \vdots \\ k(x_M, x_1) & \ldots & k(x_M, x_M) \end{pmatrix}. \quad (6)$$

It is common to use $\mu(x_1) = \mu(x_2) = \cdots = \mu(x_M) = m$, where $\mu(x_{1:M})$ is a constant mean function. m is one of the hyperparameters, and it is common to use a parametric mean function, because it provides a useful bias for the prediction of the model [16], [30], [33]. The kernel function is essential, because it captures the interactions in the model input variables. We adopt the radial basis function kernel [14] as the kernel function, given by

$$k(x_i, x_j) = \alpha^2 exp(-\frac{||x_i - x_j||^2}{2\eta^2}), \quad (7)$$

where α^2 is the signal variance corresponding to the function smoothness, and η is the kernel width that determines the radial action.

The prediction model, $f(x_{M+1})$, for the unexplored point, x_{M+1}, can be constructed using the joint distribution of $f(x_{1:M})$ over M data points. The prediction model follows a normal

distribution with a posterior mean, $\hat{\mu}(x_{M+1})$, and a variance, $\hat{\sigma}^2(x_{M+1})$:

$$f(x_{M+1}) \sim N(\hat{\mu}(x_{M+1}), \hat{\sigma}^2(x_{M+1})), \quad (8)$$

$$\hat{\mu}(x_{M+1}) = m + k(x_{M+1}, x_{1:M})^T (K(x_{1:M}) + \sigma_n^2 I)^{-1} (y_{1:M} - m), \quad (9)$$

$$\hat{\sigma}^2(x_{M+1}) = k(x_{M+1}, x_{M+1}) - k(x_{M+1}, x_{1:M})^T (K(x_{1:M}) + \sigma_n^2 I)^{-1} k(x_{M+1}, x_{1:M}), \quad (10)$$

where $k(x_{M+1}, x_{1:M})^T = [k(x_{M+1}, x_1), \ldots, k(x_{M+1}, x_M)]^T$. The accuracy of the prediction model strongly depends on the training of the surrogate model. The training is performed by finding the optimal hyperparameter vector, θ^*, where $\theta = [m, \sigma_n, \alpha, \eta]$. θ is initialized using the heuristic method developed by Yang *et al.* [34]. Then, it is optimized to maximize the type-II likelihood or, equivalently, to minimize the negative log marginal likelihood $-\log p(y_{1:M}|x_{1:M}, \theta)$ [30].

$$\theta^* = \arg\min_{\theta \in \Theta} -\log p(y_{1:M}|x_{1:M}, \theta), \quad (11)$$

$$-\log p(y_{1:M}|x_{1:M}, \theta) = \frac{1}{2}(y_{1:M} - m_\theta)^T (K(x_{1:M})_\theta + \sigma_{n,\theta}^2 I)^{-1}$$
$$\times (y_{1:M} - m_\theta) + \frac{1}{2}\log |K(x_{1:M})_\theta$$
$$+ \sigma_{n,\theta}^2 I| + \frac{M}{2}\log(2\pi). \quad (12)$$

Θ is a 4D space for the hyperparameter, θ. The dependency of θ is denoted by the θ subscript (e.g., m_θ). The first and second terms in (12) refer to model accuracy and complexity. The third term corresponds to the likelihood tendency over the observation number, M [16]. For minimization, the state-of-the-art gradient-descent optimization algorithm, ADAM, is applied, where the gradient is evaluated via subsampling [35]. The expression, $p(y_{1:M}|x_{1:M}, \theta)$, is derived to marginalize out the uncertainty about θ for the unknown function, $f(x_{M+1})$:

$$f(x_{M+1}) = E_{\theta|D_{1:M}}[f(x_{M+1}; \theta)]$$
$$= \int f(x_{M+1}; \theta) p(\theta|D_{1:M}) d\theta. \quad (13)$$

The posterior belief over θ given $D_{1:M}$ can be expressed via Bayes' rule as:

$$p(\theta|D_{1:M}) = \frac{p(y_{1:M}|x_{1:M}, \theta) p(\theta)}{p(D_{1:M})}. \quad (14)$$

$p(\theta|D_{1:M})$ determines the fitness of θ to the $D_{1:M}$, and it is proportional to $p(y_{1:M}|x_{1:M}, \theta)$. Thus, maximizing $p(y_{1:M}|x_{1:M}, \theta)$ is crucial to improve model accuracy, which can be obtained by (11). This approach is applicable to a binary classification problem. θ can be replaced by a classification criterion (i.e., threshold) and the performance of the classifier can be optimized by maximizing $p(y_{1:M}|x_{1:M}, \theta)$ [36].

2) BAYESIAN LINEAR REGRESSION

BLR is employed to model the objective function, $f(x)$, as:

$$f(x) = \mathbf{w}^T \phi(x) + \boldsymbol{m}, \qquad (15)$$

$$y = f(x) + \epsilon, \qquad (16)$$

where $\phi: R^D \rightarrow R^\mathbf{d}$ is a feature map, and $\mathbf{w} \in R^\mathbf{d}$ is a weight vector of the feature vector, $\phi(x)$. $\mathbf{d}$ is the number of extracted features. $\mathbf{w}$ determines the shape of the objective function, and the posterior distribution of $\mathbf{w}$ after obtaining $\mathbf{D_{1:M}}$ is given by:

$$\mathbf{w}|\mathbf{D_{1:M}} \sim N(\hat{\mu}_\mathbf{w}, \hat{\Sigma}_\mathbf{w}), \qquad (17)$$

where $\hat{\mu}_w = (\Phi\Phi^T + \sigma_n^2 I)^{-1}\Phi y_{1:M}$, $\hat{\Sigma}_w = \sigma_n^2(\Phi\Phi^T + \sigma_n^2 I)^{-1}$, and Φ is $\mathbf{d} \times M$ matrix in which the ith column corresponds to $\phi(x_i)$ [29]. We apply the random feature map (RFM), which has shown successful results in previous works [31], [32]. The RFM approximates Gaussian kernel function, $k(x_i, x_j)$ in GPR, using a sampling approach.

Using the Bochner's theorem, the kernel of the unit width, $k(\delta) = \exp(-||\delta||^2/2)$, can be expressed as

$$k(x - x') = \int \exp(j\omega^T(x - x'))p(\omega)d\omega, \qquad (18)$$

where j indicates an imaginary number, and $p(\omega) = (2\pi)^{-D/2}\exp(-||\omega||^2/2)$. By introducing $z_{\omega,b}(x) = \sqrt{2}cos(\omega^T x + b)$, where ω follow $p(\omega)$, and b are sampled uniformly from $[0,2\pi]$, $E[z_{\omega,b}(x)z_{\omega,b}(x')] = k(x-x')$. Therefore, $\phi(x)^T\phi(x')$ can approximate the Gaussian kernel with width, η, and variance, α^2 (i.e., $\alpha^2\exp(-||x-x'||^2/2\eta^2)$) by defining $\phi(x) = \alpha(z_{\omega_1,b_1}(x/\eta), \ldots, z_{\omega_\mathbf{d},b_\mathbf{d}}(x/\eta))$, where $\{\omega_i, b_i\}_{i=1}^d$ is the set of $\mathbf{d}$ random samples. When $\mathbf{d} \rightarrow \infty$, the BLR converges to the GPR. The BLR is trained by maximizing the type-II likelihood, as explained in (11)-(12).

3) ACQUISITION FUNCTION

EI is chosen as an acquisition function, because it can be used as the stopping condition, and it is the most popular acquisition function for BO [24]. EI calculates the expectation value of the improvement over the maximum in the observation dataset, $y^* = \max y_{1:M}$. At iteration T, it can be expressed as

$$\mathbf{EI}(x)_T = \sigma(x)\xi(z') + (\mu(x) - y_T^*)\Xi(z'), \qquad (19)$$

where $\sigma(x)$ and $\mu(x)$ are the estimation variance and mean of the surrogate model, respectively. They can be obtained using (9) and (10) in GPR.

For the BLR, they can be obtained by inserting $\hat{\mu}_w$ and $\hat{\Sigma}_w$ into (15) $z'(x) = (\mu(x)-y_T^*)/\sigma(x)$, where y_T^* is the maximum observed value at iteration T. ξ and Ξ correspond to the standard normal probability density function and the standard normal cumulative distribution function, respectively.

We plot the Bayesian optimization of the toy function in Fig. 2 using the Python library, *skopt*. The surrogate model is GPR and acquisition function is $\mathbf{EI}(x)$ following Algorithm 1. $\mathbf{N_{init}}$, $\mathbf{N_{sp}}$, and $\mathbf{N_{iter}}$ are 5, 1, and 100 respectively. At each iteration, T, the next sampling point is chosen
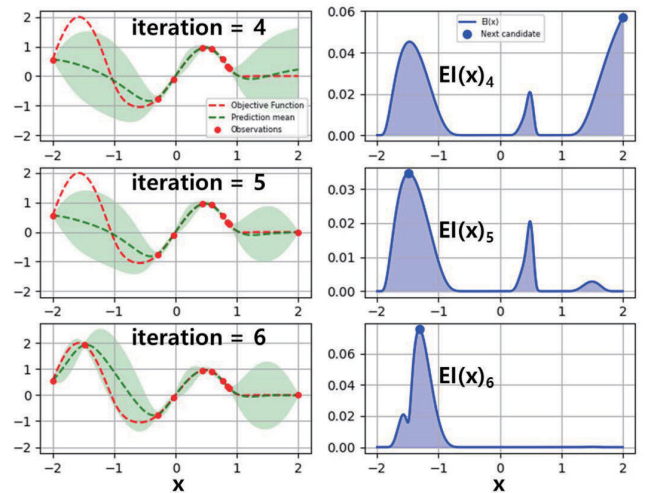

FIGURE 2. Bayesian optimization of the toy function. x indicates the input space. Red dashed line indicates the objective function with zero noise. Red circle denotes the observation value. Green dashed line indicates the mean of the prediction. Green shaded region denotes the possible values that the prediction model can have.

by maximizing $\mathbf{EI}(x)_T$. Then, that point is observed at the $T + 1$ iteration. By repeating this automatic feedback process, the BO reaches the global maximum of the non-concave objective function within a few observations. Algorithm 1 continues until T becomes the $\mathbf{N_{iter}}$. If BO finds the global optimum at $T \ll \mathbf{N_{iter}}$, many redundant observations are generated after reaching the global optimum in this algorithm. To maximize the efficiency of BO, we modify Algorithm 1 to use ESC, as shown in Algorithm 2. At each iteration T, $EI_{max,T}$ is calculated. If $EI_{max,T}$ is less than ESC, a 1% of unit value, the optimization stops and reports the best-observed value. The Algorithm in the previous work stops BO when EI_{max} is less than a tiny constant κ (e.g., 10^{-9}) [24]. We replace κ with ESC. Line 20-39 in Algorithm 2 is same with line 16-21 in Algorithm 1 when $N_{sp} = 1$. It is a method to choose sampling points for the next iteration when $N_{sp} > 1$, which is implemented in the COMBO library. When $N_{sp} > 1$ and $\alpha(x)_T = \mathbf{EI}(x)_T$, Algorithm 1 chooses samples with N_{sp} highest $\mathbf{EI}(x)_T$ values. However, Algorithm 2 selects one sample with the highest $\mathbf{EI}(x)_T$ value when the sampling index s is one. For the rest $N_{sp}-1$ points, at $s > 1$, posteriors of previously selected points $(f(x_1) \ldots f(x_{s-1}))$ are sampled and trained N_{ps} times, generating N_{ps} different prediction models $(N(\mu(x)_p, \sigma(x)_p))$, with acquisition functions $\mathbf{EI}(x)_{T,p}$. N_{ps} is the number of posterior samplings and p is the posterior sampling index. The sampling point when $s > 1$, x_s, is chosen when it exhibits highest average value of $\mathbf{EI}(x)_{T,p}$ (i.e., $x_s = \text{argmax}_{x \in \mathcal{X}'} \frac{1}{N_{ps}} \Sigma_{p=1}^{N_{ps}} \mathbf{EI}(x)_{T,p}$).

B. TCAD SIMULATIONS

In this work, we apply our BO scheme to two different devices; SG nMOSFET, and NS FET. The first one is chosen, because its computational loads are sufficiently small

Algorithm 2 Bayesian Optimization Using ESC
___
Require: Number of initial sampling points N_{init}, number of iterations N_{iter}, number of samplings at each iteration N_{sp}, and small positive constant κ (e.g. 10^{-9}), number of posterior samplings N_{ps}

1: Randomly sample N_{init} input points in the design space: $x_{1:N_{init}}$
2: Set Number of observations $M = 0$
3: Set κ = ESC (contribution of this work)
4: **for** iteration T = 1 to ∞ **do**
5:    **if** T == 1 **then**
6:       Observe $y_{1:N_{init}}$ corresponding to $x_{1:N_{init}}$
7:       M + = N_{init}
8:       Construct the data set $\mathbf{D_T} = \mathbf{D_{1:M}} = \{(x_i, y_i)\}_{i=1}^{N_{init}}$
9:    **else**
10:       Observe $y_{1:N_{sp}}$ corresponding to $x_{1:N_{sp}}$
11:       M + = N_{sp}
12:       $\mathbf{D_T} = \mathbf{D_{1:M}} = \mathbf{D_{T-1}} \cup \{(x_i, y_i)\}_{i=1}^{N_{sp}}$
13:    **end if**
14:    Fit probabilistic surrogate model $f(x)$ to $\mathbf{D_T}$
15:    Get unexplored input set $\mathcal{X}' = \mathcal{X} \cap \{\{x_i\}_{i=1}^M\}^c$
16:    $EI_{max,T} = \max_{x \in \mathcal{X}'} \mathbf{EI}(x)_T$
17:    **if** $EI_{max,T} < \kappa$ **then**
18:       Stop the optimization and go to line 42
19:    **end if**
20:    Set $x_{1:N_{sp}} = \varnothing$
21:    **for** s = 1 to N_{sp} **do**
22:       **if** s == 1 **then**
23:          $x_s = \text{argmax}_{x \in \mathcal{X}'} \mathbf{EI}(x)_T$
24:       **else**
25:          **for** p = 1 to N_{ps} **do**
26:             Set temporary data set $\mathbf{D_{T,temp}} = \mathbf{D_T}$
27:             **for** q = 1 to s-1 **do**
28:                $x_q = x_{1:N_{sp}}(q)$
29:                sample $f(x_q)$ from $N(\hat{\mu}(x_q), \hat{\sigma}^2(x_q))$
30:                $\mathbf{D_{T,temp}} = \mathbf{D_{T,temp}} \cup \{x_q, f(x_q)\}$
31:             **end for**
32:             $\sigma(x)_p, \mu(x)_p \leftarrow$ Fit $f(x)$ to $\mathbf{D_{T,temp}}$
33:             Get $\mathbf{EI}(x)_{T,p}$ using $\sigma(x)_p, \mu(x)_p$
34:          **end for**
35:          $x_s = \text{argmax}_{x \in \mathcal{X}'} \frac{1}{N_{ps}} \Sigma_{p=1}^{N_{ps}} \mathbf{EI}(x)_{T,p}$
36:       **end if**
37:       $x_{1:N_{sp}} = x_{1:N_{sp}} \cup \{x_s\}$
38:       $\mathcal{X}' = \mathcal{X} \cap \{x_s\}^c$
39:    **end for**
40:    Get next sampling point set $x_{1:N_{sp}}$
41: **end for**
42: **return** $y^* = \max y_{1:M}$ and x^* (input of y^*)
___

to be used as test system for developing the BO scheme. The developed BO scheme is then applied to the NS FET, which is much larger in terms of computational size. For SG nMOSFETs, the simulations were conducted in 2D mesh space, so that a complete dataset of 2,800 SG nMOS FETs can
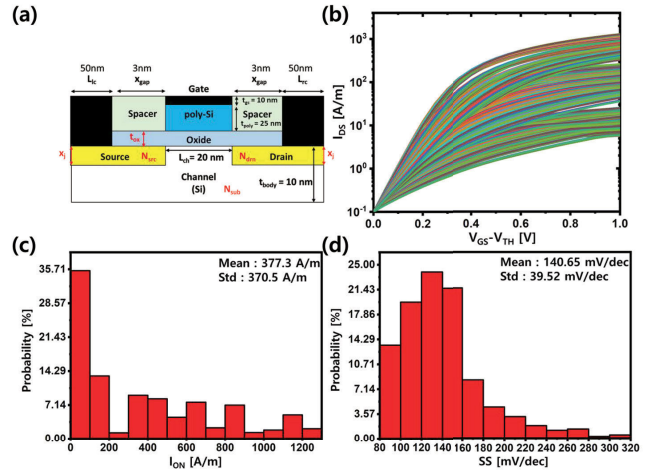

FIGURE 3. (a) Device schematics, (b) current-voltage characteristics, (c) I_{ON} distribution, and (d) SS distribution of 2,800 SG nMOSFETs. Red letters in (a) indicate design variables for optimization.

TABLE 1. Input space of SG nMOSFET optimization.

Variables	Values
t_{ox} [nm]	1, 1.5, 2, 2.5, 3
N_{sub} [cm^{-3}]	$10^{11},10^{12},10^{13},10^{14},10^{15},10^{16},10^{17}$
x_j [nm]	1, 1.5, 2, 2.5, 3
N_{src} [cm^{-3}]	$10^{18},10^{19},10^{20},10^{21}$
N_{drn} [cm^{-3}]	$10^{18},10^{19},10^{20},10^{21}$
Total	5×7×5×4×4 = 2,800 devices

be obtained within a reasonable time. For the 2D mesh simulations, SILVACO ATLAS [37] is used with default material parameters. For the 3-nm node NS FET optimization, where device calculation in 3D mesh requires several hours, Synopsys Sentaurus [38] is used. The numerical transport calculation is conducted by self-consistently solving several equations, including continuity, Poisson, density gradient, and contact equation. SCE is calculated in every simulation of a nanometer-scale device by solving the numerical equations. We construct a multi-dimensional input space for optimization by varying design parameters independently. Design variables for optimization are chosen based on previous studies [5]–[14]. Each variable can improve or degrade device characteristics.

1) SG nMOSFET SIMULATION
The structure of the SG nMOSFET is shown in Fig. 3(a). The channel length (L_{ch}), and body thickness (t_{body}) are assumed to be 20 and 10 nm, respectively. The supply voltage (V_{DD}) is 1.0 V. I_{ON} is defined as I_{DS} at $V_{GS} - V_{TH} = V_{DD}$. V_{TH} is the V_{GS} at $I_{DS} = 0.1$ A/m for every simulation of the SG nMOSFET and NS FET. SS, $\partial V_{GS}/\partial \log_{10}(I_{DS})$, is measured at V_{TH}. 5D input space consisting of 2,800 SG nMOSFETs is constructed by changing five design variables independently: t_{ox}, N_{sub}, x_j, N_{src}, and N_{drn}. Details of the input space are presented in Table 1. The current-voltage characteristics of 2,800 SG nMOSFETs are given in Fig. 3(b),

where I_{ON} is distributed with a mean ($\mu_{I_{ON}}$) of 377.3 and a standard deviation ($\sigma_{I_{ON}}$) of 370.5 A/m. The SS is distributed with a μ_{SS} of 140.65 and a σ_{SS} of 39.52 mV/dec. To include the effects of the doping density and electrical fields, we apply the CVT model [39]. To calculate the oxide leakage, the Fowler-Nordheim tunneling model [40] is applied.

2) 3-nm NODE NS FET SIMULATION

For an application of our framework to a highly sophisticated optimization, we choose a 3-nm node NS FET as the optimization target. The structure of the three-stacked NS FET is shown in Fig. 4. We benchmark the structure of the NS FETs from previous works [5]–[8].
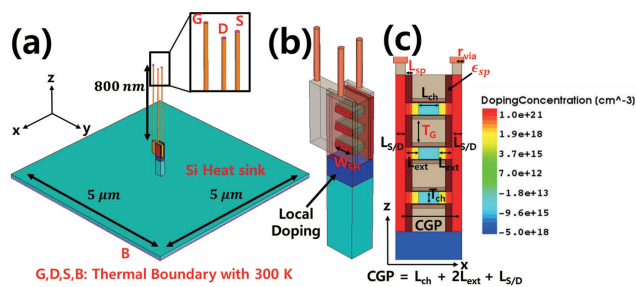

FIGURE 4. Device schematics of 3-nm node NS FET: (a) 3D structures with thermal boundaries, (b) enlarged device region, and (c) x-z plane view at y = 0. Red letters in (b) and (c) indicate design variables for optimization.

TABLE 2. Device parameters of 3-nm node NS FET optimization.

Parameter	Value
Channel Length, L_{ch}	8.5 nm
Extension Length, L_{ext}	5.75 nm
Channel Thickness, T_{ch}	5 nm
Contacted Gate Pitch, CGP	24 nm
S/D Length, $L_{S/D}$	4 nm
SiO$_2$ thickness	0.45 nm
HfO$_2$ thickness	1.5 nm
EOT	0.7 nm
Channel Doping	1×10^{15} cm^{-3}
S/D Doping	1×10^{21} cm^{-3}
Local Doping	5×10^{18} cm^{-3}
Doping Gradient	1 nm/dec
Supply Voltage, V_{DD}	0.55 V
Channel Thermal Conductivity	7.5 W/K·m [8]
Gate Oxide Thermal Conductivity	2.5 W/K·m [8]
S/D Thermal Conductivity	3.8 W/K·m [8]
Local Doping Thermal Conductivity	33 W/K·m [8]
Bottom Substrate Thermal Conductivity	150 W/K·m [8]
Contact Pad Thermal Conductivity	238 W/K·m [5]

For a realistic simulation, we include the QME and SHE. For the SHE, a thermodynamic transport model is used. The thermal conductivities for the SHE simulations and other device parameters are listed in Table 2. For QME, the eQuantumPotential is used. A BandGapNarrowing having an input-band gap of 1.241 eV [41] and Fermi statistics are used. For mobility estimation, Enormal (Lombardi) and Phumob are used to consider Coulomb and interface scattering, respectively. The saturation velocity is assumed

TABLE 3. Input space of 3-nm node NS FET optimization.

Variables	Values (min,max,interval)
L_{sp} [nm]	(3.0, 7.0, 0.5)
W_{SH} [nm]	(8.0, 40.0, 1.0)
T_G [nm]	(2.0, 20.0, 1.0)
r_{via} [nm]	(3.0, 7.0, 0.5)
ϵ_{sp} [ϵ_0]	1.0 (Air), 3.9 (SiO$_2$), 7.5 (Nitride), 22.0 (HfO$_2$)
Total	$9 \times 33 \times 19 \times 9 \times 4 = 203{,}148$ devices

to be 2.2×10^7 cm/s [7]. The thermal contact and electrical contact resistance are assumed to be 2×10^{-4} cm^{2}K/W [8] and 3×10^{-9} Ω-cm^2 [7], respectively. A $5 \times 5 \times 2.5$ μm^3 Si substrate and an 800-nm back end of lines are included in the simulation for thermal boundaries. The device is assumed to be surrounded by SiO$_2$ for isolation. We use the contacted gate pitch (CGP) as an optimization constraint to maintain the device scale in the lateral direction [6]–[8]. For the optimization, we vary five design parameters which alter the device characteristics of the NS FET: L_{sp}, W_{SH}, T_G, r_{via}, and ϵ_{sp}. We construct 5D input space by varying each parameter independently as shown in Table 3, resulting in 203,148 candidates.

C. MULTI-OBJECTIVE OPTIMIZATION

MOSFET optimization requires multiple figures-of-merits (FOMs) to be jointly optimized, because the practical device needs to exhibit high performance with low-power consumption to meet the industrial standard. We perform multi-objective optimization by transforming it into a mono-objective optimization. The scalarization by the weighted sum of different objective functions has been used for multi-objective BO, in which the mono-objective function has been called as the "target function" [21]. This approach can provide Pareto optimal points where the improvement of one objective is infeasible without degradation of the other objectives [17], [42]. Although previous works used the weighted sum of objectives, we use the weighted sum of normalized objectives, because each objective has a different scale. In our work, the target function, **Z**, is given as:

$$\mathbf{Z} = \Sigma_{i=1}^{n} w^i \times norm(y^i), \qquad (20)$$

where y^i and w^i indicate the observation value and weight of the ith objective, respectively. $norm()$ is the normalization function that transforms y^i into values between 0 and 1 (i.e., [0,1]): $norm(y^i) = y^i/y^i_{max}$ for the maximizing object, and $norm(y^i) = y^i_{min}/y^i$ for the minimizing object. y^i_{max} and y^i_{min} are obtained from the single-objective optimization of y^i. w^i is given by

$$w^i = (1 - \frac{y^i_{max} - \delta y^i}{y^i_{max}})^{-1}, \qquad (21)$$

$$w^i = (1 - \frac{y^i_{min}}{y^i_{min} + \delta y^i})^{-1}. \qquad (22)$$

(21) and (22) are used to maximize and minimize objects, respectively. δy^i is the controlling factor of weight, where the
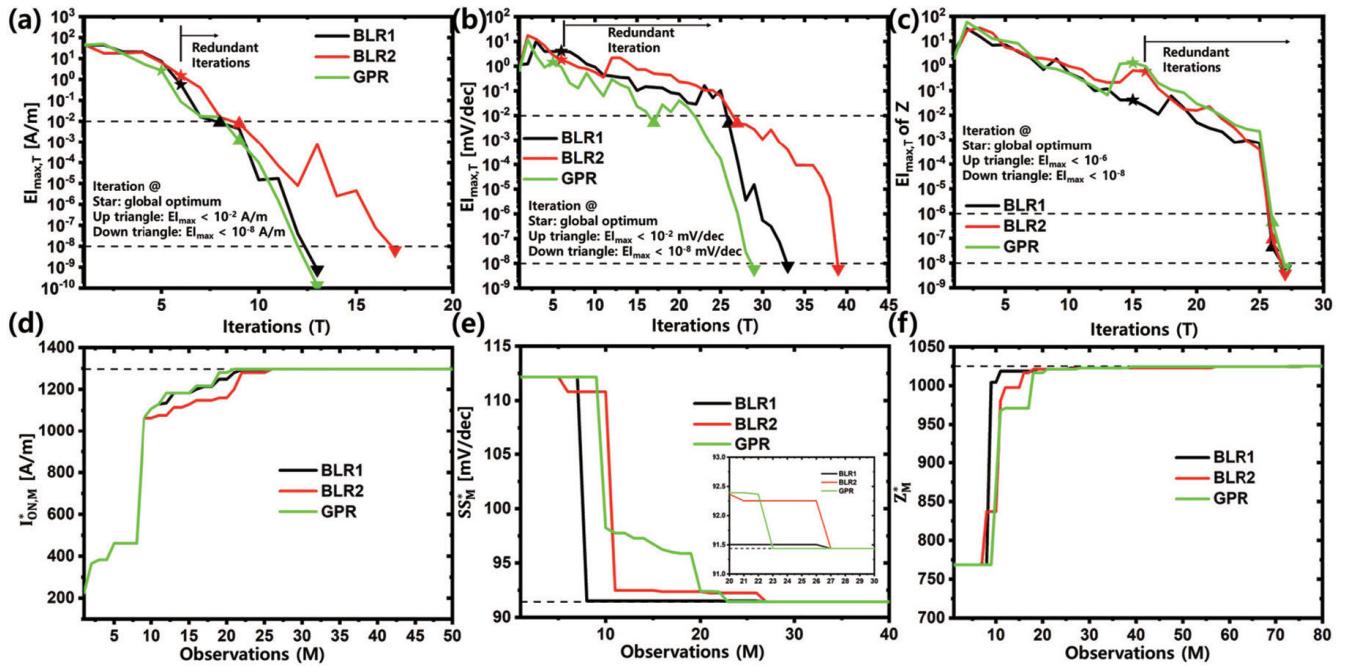
FIGURE 5. Results of SG nMOSFET optimization. EI_{max} at T iteration during the optimization of (a) I_{ON}, (b) SS, and (c) Z optimization. Stars in (a)-(c) indicate the iteration when BO found global optimum. Up triangles in (a)-(c) denote iteration reached effective stopping conditions of 10^{-2} A/m for I_{ON}, of 10^{-2} mV/dec for SS, and of 10^{-6} for Z. Down triangle in (a)-(c) denote iteration reached stopping conditions of $EI_{max} < 10^{-8}$. Best observed values at M observations when it comes to (d) I_{ON}, (e) SS, and (f) Z optimization. Dashed lines in (d)-(f) indicate global optimum for each optimization.

δy^i improvement near optimum gives one point to the target function, **Z**.

III. RESULTS AND DISCUSSION

A. EFFECTIVE STOPPING CONDITION FOR THE BO OF MOSFET DEVICES

We first investigated the efficiency of ESC and carried out simulations of SG nMOSFETs for this purpose. To compare our method with the tiny positive constant method in the previous work [24], we first set κ of 10^{-8} as the stopping condition for the optimization. This choice of κ could require large redundant iterations, resulting in a decrease in the efficiency of BO. We scanned the whole design space of SG nMOSFETs so that we know the global optimum of the design space more precisely. We measured the efficiency of the ESC when it was applied to I_{ON}, SS, and multi-objective optimization. We compared the optimization trajectories of different surrogate functions (i.e, BLR with **d** = 1,000 (BLR1) and **d** = 5,000 (BLR2), and GPR) to determine the suitability of each surrogate model. **d** is the number of extracted features for BLR. The number of initial sampling points, N_{init}, and the number of samplings at each iteration, N_{sp}, were each set to be 5.

We conducted I_{ON} optimization of the SG nMOSFET with a 5D design space. The optimization stopped when the $EI_{max} < 10^{-8}$ A/m. The globally optimal I_{ON}^* was found to be 1,297 A/m, with the transfer characteristics shown in Fig. 6(a). All of the attempted optimizations with different surrogate models succeeded in finding the global

optimum of the I_{ON} before $EI_{max} < 10^{-8}$ A/m and before T = 7 (Fig. 5(a)). However, to meet the stopping condition of $EI_{max} < 10^{-8}$ A/m, 7 to 11 redundant iterations corresponding to 35 to 55 redundant device calculations were required, even after the optimizer found the global maximum, because the stopping condition, κ, was too strict. Therefore, to minimize redundant evaluations of the device characteristics, κ must be relaxed.

By utilizing the ESC, we can boost the efficiency of the BO. We set κ to 10^{-2} A/m as the ESC for I_{ON} which is 1% of the unit value. The unit value means the smallest number that carries a meaningful contribution to the observation. We set the unit value of I_{ON} to 1 A/m, such that an I_{ON} improvement of less than 1 A/m is meaningless. Compared with the tiny constant method, a substantial reduction in the required N_{iter} for optimization can be achieved by utilizing the ESC: 38% in BLR1 (13 to 8), 47% in BLR2 (17 to 9), and 31% in GPR (13 to 9). The required N_{iter} was much smaller than N_{iter} from the a previous work that used an N_{iter} of 100 [22] for a preset stopping condition.

Next, the SS optimization of the SG nMOSFET design space was carried out, as SS affects the standby power dissipation of the MOSFET device. The optimization stopped when $EI_{max} < 10^{-8}$ mV/dec. The globally optimal SS^* was found at T≤6 (Fig. 5(b)) and it turned out to be 91.44 mV/dec with the transfer characteristics shown in Fig. 6(a). We have set κ to 10^{-2} mV/dec as ESC for SS (i.e., 1% of 1 mV/dec). Enhancing the efficiency of the BO by utilizing the ESC was also achieved during the SS optimization. A considerable
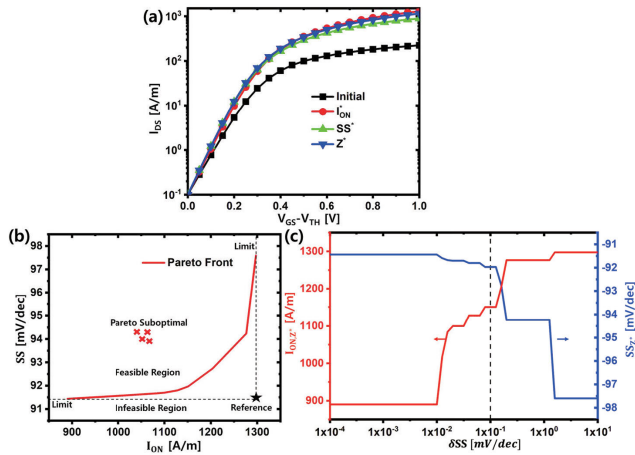
FIGURE 6. (a) Current-voltage characteristics of initial and the global optimal devices in SG nMOSFET optimization, (b) Pareto front of SG nMOSFET design space, and (c) Pareto optimal points corresponding to δSS.

TABLE 4. Device specification in SG nMOSFET optimization.

Device	t_{ox} [nm]	N_{sub} [cm^{-3}]	x_j [nm]	N_{src} [cm^{-3}]	N_{drn} [cm^{-3}]	I_{ON} [A/m]	SS [mV/dec]
Initial	1.5	10^{11}	3	10^{19}	10^{20}	225	112.17
I_{ON}^*	1	10^{16}	3	10^{21}	10^{21}	1,297	97.60
SS*	1	10^{17}	1	10^{20}	10^{18}	890	91.44
Z*	1	10^{16}	3	10^{20}	10^{19}	1,151	91.97

With an analytical reasoning alone, it could be difficult to attain the results that showed that N_{sub} should be 10^{16} cm^{-3} for I_{ON}^* and **Z***, but it should be 10^{17} cm^{-3} for SS* with different design parameters such as N_{src} and N_{drn}. By utilizing the automatic ML-based optimization framework, we can obtain a globally optimized device without intensive human effort.

We found that the GPR was sufficient for SG nMOSFET optimization. More than 900 observations were required for BLR to be more efficient than GPR (Fig. 7). In the 5D SG nMOSFET optimization, small observations ($\ll$900) were required, so using GPR as a surrogate model, we found that ESC was robust and efficient.
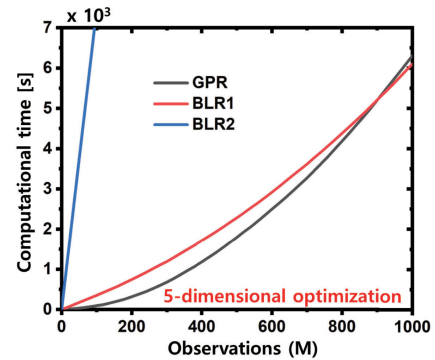

FIGURE 7. Comparison of computational time required for optimization of SG nMOSFET among three different surrogate models: BLR1, BLR2, and GPR. The time was measured by Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20 GHz.

reduction of N_{iter} was obtained compared with the tiny constant method: 21% in BLR1 (33 to 26), 31% in BLR2 (39 to 27), and 41% in GPR (29 to 17).

For a multi-objective optimization, the SS and I_{ON} were simultaneously optimized by introducing the target function, **Z**: $\mathbf{Z} = w^1 \times SS + w^2 \times I_{ON}$ (Fig, 5(f)). The set of Pareto optimal points (Pareto front) in Figs. 6(b) and (c) was determined by changing the δSS and by maximizing **Z** [42]. The δI_{ON} was fixed to 10 A/m. The Pareto front in Fig. 6(b) denotes the design boundary of the multi-objective optimization, that is, the design outside the Pareto front is infeasible. Although the target value method [14] can be used for multi-objective optimization, the target values can be Pareto suboptimal points or outside the design boundary. Therefore, we used the weighted sum of objectives, such that the maximum of **Z** was on the Pareto front. Because the design boundary of SS (SS$_{\mathbf{Z}*}$) was abruptly degraded when δSS was greater than 0.1 mV/dec (Fig. 6(c)), we used a δSS of 0.1 mV/dec and a δI_{ON} of 10 A/m. The optimization of **Z** stopped when $EI_{max} < 10^{-8}$. All of the attempted optimizations with three surrogate models found the globally optimal device before $EI_{max} < 10^{-8}$, as shown in Fig. 5(c), with an I_{ON} of 1,151 A/m and SS of 91.97 mV/dec. Both I_{ON} and SS were close to the globally optimal values from the single-objective optimization: their deviations were 146 A/m and 0.53 mV/dec from I_{ON}^* and SS*, respectively (Fig. 6(a)). We have set κ to a value of 10^{-6} as an ESC of **Z** (i.e., 1% of 10^{-4}). Because 1 point improvement of SS can be offset by 1 point degradation of I_{ON}, the unit value must be a small positive constant. We found that a 10^{-4} of **Z** was sufficiently small as the unit value. The reduction of N_{iter} was 1 (27 to 26, 4%) for all surrogate models. A rapid reduction in EI_{max} was observed when $EI_{max} < 10^{-3}$ (Fig. 5(c)).

The device specifications of the initial device and globally optimal device are listed in Table 4. Each specification was the solution of the complex function in a 5D design space.

We optimized the SG nMOSFET with different initial devices which were generated with 34 different random seeds. All the optimization runs stopped at ESC: κ for I_{ON}, SS, and **Z** were 10^{-2} A/m, 10^{-2} mV/dec, and 10^{-6}, respectively (Figs. 8(a)-(c)). I_{ON}, SS, and **Z** converged to the global optimum before the EI_{max} met the ESC (Figs. 8(d)-(f)). Few SS optimization runs finished at sub-optimal points, but the mean deviation of SS optimization from the global optimum was 0.05 mV/dec, which can be regarded as negligible in a device analysis.

ESC only required 12.4, 23.1 and 27.4 iterations on average for the optimization of I_{ON}, SS and **Z**, respectively. The use of ESC was efficient, because less than 5% observations of the total of 2,800 devices were needed for the optimization of the SG nMOSFET on average. Compared with the previous empirical method which used the stopping condition of $N_{iter} = 100$ [22], 87.6%, 76.9%, and 72.6% reductions of required N_{iter} were achieved in optimizations of I_{ON}, SS and **Z**, respectively.
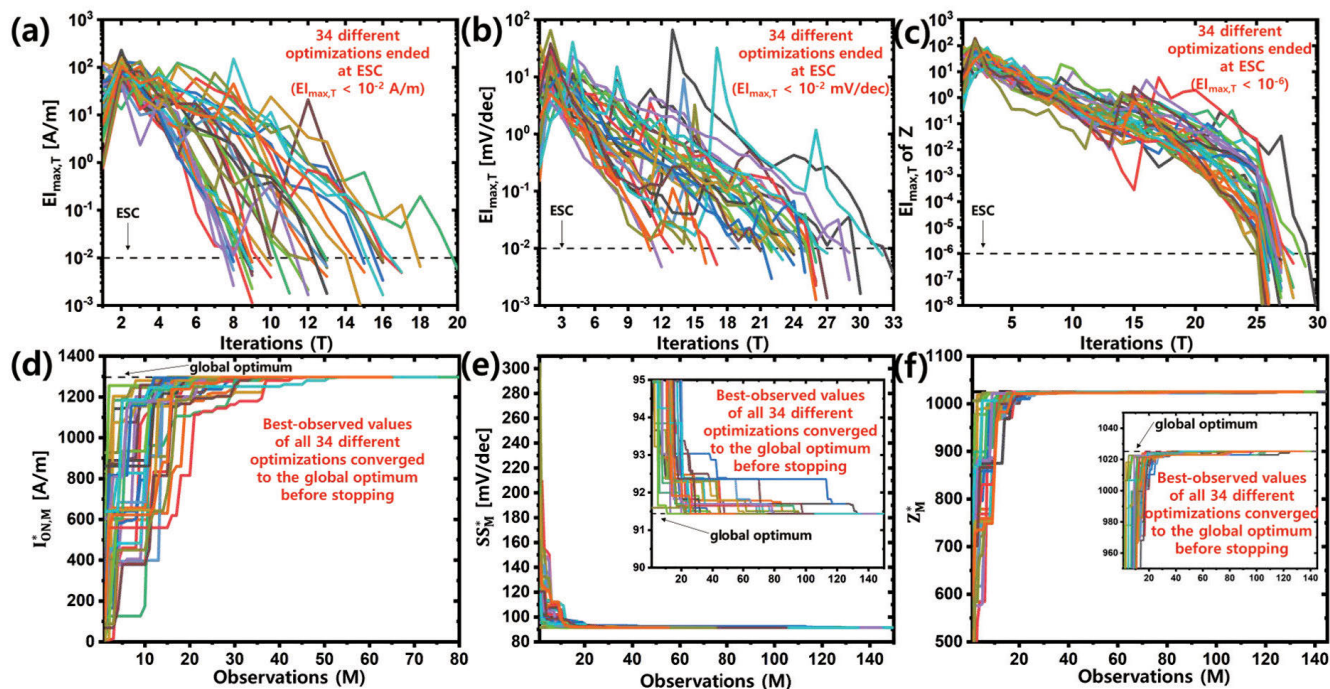
FIGURE 8. Robustness test of ESC. EI_{max} at T iteration of 34 different initial seeds when it comes to (a) I_{ON}, (b) SS, and (c) Z optimization. Each $EI_{max,T}$ of 34 different optimization is plotted with different colors. Every optimization ended at ESC. Dashed lines in (a)-(c) indicate ESC value for each optimization: 10^{-2} A/m for I_{ON}, of 10^{-2} mV/dec for SS, and of 10^{-6} for Z. Best observed value at M observations of 34 different optimizations when it comes to (d) I_{ON}, (e) SS, and (f) Z optimization. Inset figures depict enlarged optimization results around the global optimum. Best-observed values of 34 different optimization are plotted with different colors. Every optimized values converged to the global optimum. Dashed lines in (d)-(f) indicate global optimum for each optimization. The globally optimum points are known in advance because we calculated every device in the input space consisting of 2,800 SG nMOSFETs to confirm that the optimal point from BO matches the true global optimum of the input space.
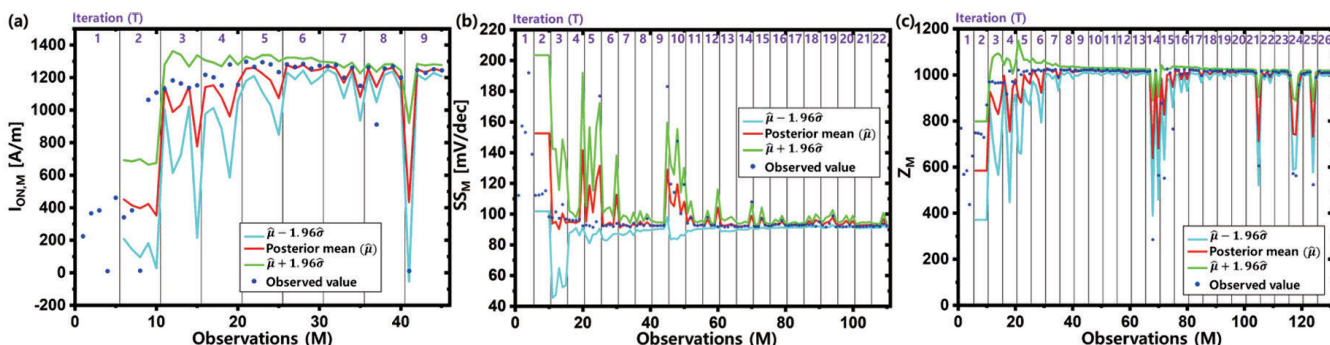

FIGURE 9. Posterior distribution of SG nMOSFET optimization when it comes to I_{ON}, (b) SS, and (c) Z optimization. GPR and ESC were applied for the optimization. posterior mean ($\hat{\mu}(x)$), with upper/lower bound ($\hat{\mu}(x) \pm 1.96\hat{\sigma}(x)$) are plotted to illustrate 95% confidence interval. Posterior values at T iteration were predicted at T-1 iteration. Observed values are plotted with posterior values. Most of observed values were in the prediction bound. As T increased, prediction mean values got closer to observed values due to improvement of model accuracy.

As explained in (8), the posterior distribution is assumed to follow normal distribution, $N(\hat{\mu}(x), \hat{\sigma}^2(x))$. We have depicted posterior mean ($\hat{\mu}(x)$), with upper/lower bound ($\hat{\mu}(x) \pm 1.96\hat{\sigma}(x)$), indicating 95% confidence interval (Fig. 9). I_{ON}, SS, and **Z** were optimized using GPR and stopped at ESC. Posterior values at T iteration were predicted at T-1 iteration. Most of observed values were in the prediction bound (95% confidence interval). As T got bigger, prediction mean got closer to observation data. It is because increased training data improved model accuracy. When T = 2, observation data

were far from the prediction mean values in Fig. 9, but got more closer to the prediction mean when T = 7.

A few data were out of the prediction bound at T > 7. It is because the BO is designed to sample training data near optimum as T increases. The prediction accuracy for values far from the optimal value got relatively low, resulting in lower bound overestimation in the maximization problem, and upper bound underestimation in the minimization problem (T = 8 in I_{ON}, T = 9 in SS, and T = 14 in **Z** optimization). These out-of-prediction bound data helped model
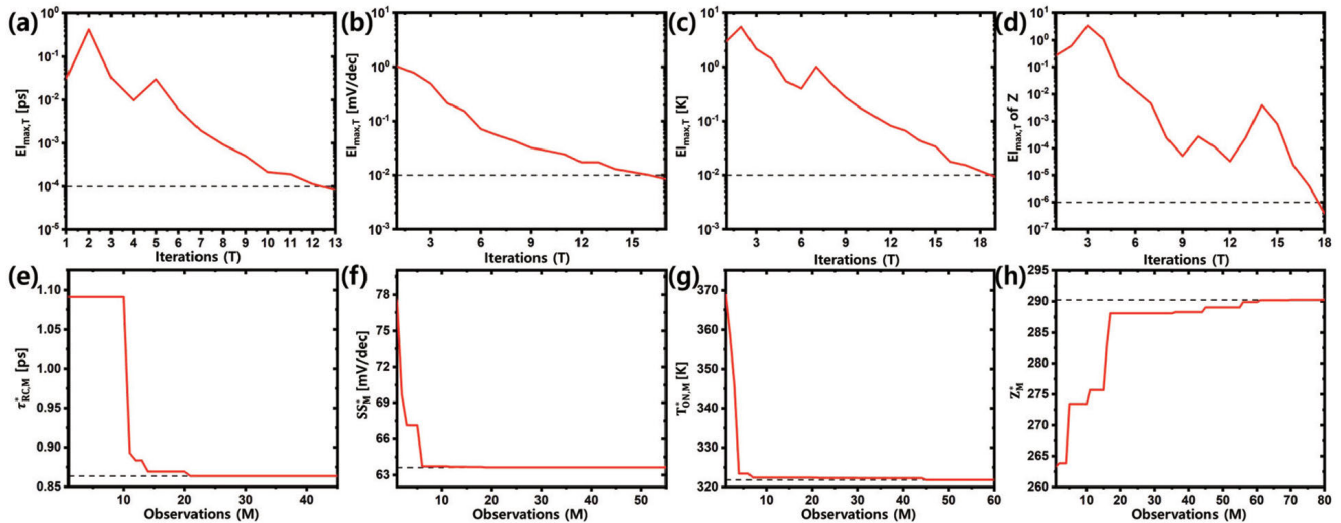
FIGURE 10. Results of NS FET optimization. EI_{max} at T iteration during the optimization of NS FET: (a) τ_{RC}, (b) SS, (c) T_{ON}, and (d) Z optimization. Dashed lines in (a)-(d) indicate ESC value for each optimization: 10^{-4} ps for τ_{RC}, of 10^{-2} mV/dec for SS, of 10^{-2} K for T_{ON}, and of 10^{-6} for Z. Best observed value at M observations during the optimization of NS FET: (e) τ_{RC}, (f) SS, (g) T_{ON}, and (h) Z optimization. Dashed lines in (e)-(h) indicate optimum for each optimization.

to improve its prediction accuracy at the next iteration by changing hyperparameters of the prediction model. There was improvement of model accuracy after the prediction failure (see T = 8 and T = 9 in I_{ON}, T = 9 and T = 10 in SS, and T = 15 and T = 16 in **Z** optimization). Even though prediction models sometimes failed to predict the observation data during optimization, we found it did not affect for BO to find the global optimum because every optimization found the global optimum before stopping (Fig. 5 and Fig. 8). Therefore, we found our model assumption in (8) was a reasonable one for the BO of SG nMOSFETs.

B. APPLICATION TO AN ADVANCED LOGIC DEVICE: OPTIMIZATION OF 3-nm NODE NS FETs

Next, we applied our BO framework to a highly sophisticated optimization of 3-nm node NS FET. The NS FET is a front-edge nanoscale device that requires demanding 3D simulations and involves heat transport as well as a charge transport. Our goal is the multi-objective optimization of the 3-nm node NS FET to meet industrial standards.

We first carried out a single-objective optimization for each of the three FOMs: τ_{RC}, SS and T_{ON}. τ_{RC} is defined as $C_{ON} \times V_{DD}/I_{ON}$, where C_{ON} is the C_{GG} at $V_{GS} = V_{DD} - V_{TH}$ and T_{ON} is the maximum lattice temperature in the device when $V_{GS} = V_{DD} - V_{TH}$. Because the industry requires MOSFETs having high-speed and low-power consumption, these FOMs must be minimized. A reduction in τ_{RC} improves the speed of the device, while decreases in the SS and T_{ON} lower the standby and ON-state power consumption of the device, respectively. T_{ON} affects the device performance and reliability through the SHE which degrades the carrier mobility and the device lifetime [5].

We set the unit values of τ_{RC}, SS, and T_{ON} as 10^{-2} ps, 1 mV/dec, and 1 K, respectively. The corresponding κ values

for the ESC were 10^{-4} ps, 10^{-2} mV/dec, and 10^{-2} K, respectively. The single-objective optimization for the three FOMs was completed at the ESC (Figs. 10(a)-(c)). The optimization of τ_{RC}, SS and T_{ON} required 13, 17, and 19 iterations, respectively. All single-objective device optimization runs ended within 19 iterations and 95 observations, which were 0.05% of the total input candidates. The optimized FOMs were τ_{RC}^* of 0.86 ps, SS^* of 63.6 mV/dec, and T_{ON}^* of 321.8 K (Figs. 10(e)-(g)). We inserted these extreme values into the target function **Z** and set $\delta\tau_{RC}$, δSS, and δT_{ON} as 0.01 ps, 1 mV/dec, and 2 K, respectively. More weight was given to τ_{RC} than T_{ON}, because a reduction of 0.01 ps, 1 mV/dec and 2 K gives similar increase to **Z**. The unit value of **Z** was 10^{-4} and the corresponding κ value for ESC was 10^{-6}.

Multi-objective optimization was completed after 18 iterations when it met the ESC (Figs. 10(d) and 10(h)) corresponding to 0.04 % of the total candidates. The optimum of **Z** fulfilled 97% of τ_{RC}^*, 85% of SS^*, and 93% of T_{ON}^* (Fig. 11). A significant improvement of FOMs compared with the initial device was obtained. The multi-objectively optimized device exhibits 22% faster speed with 23.9 K less T_{ON} than initial device.

As shown in Fig.11 and Table 5, τ_{RC} was minimized by increasing I_{ON} through the W_{SH} expansion and the L_{sp} reduction and by decreasing C_{ON} through T_G and r_{via} reduction, compared with the initial device. SS and T_{ON} were minimized by reducing power consumption via W_{SH} reduction. The C_{ON} of SS^* was larger than that of initial device, and C_{ON} of T_{ON}^* was smaller than the initial one. For multi-objective optimal point, $\mathbf{Z}^*$, all the device parameters were finely tuned. Low T_{ON} is attributed to Si_3N_4 spacer with higher thermal conductivity (18.5 W/K·m) than air spacer (10^{-8} W/K·m). The device specifications in Table 5 were the optima of the complex black box function in the 5D design space.
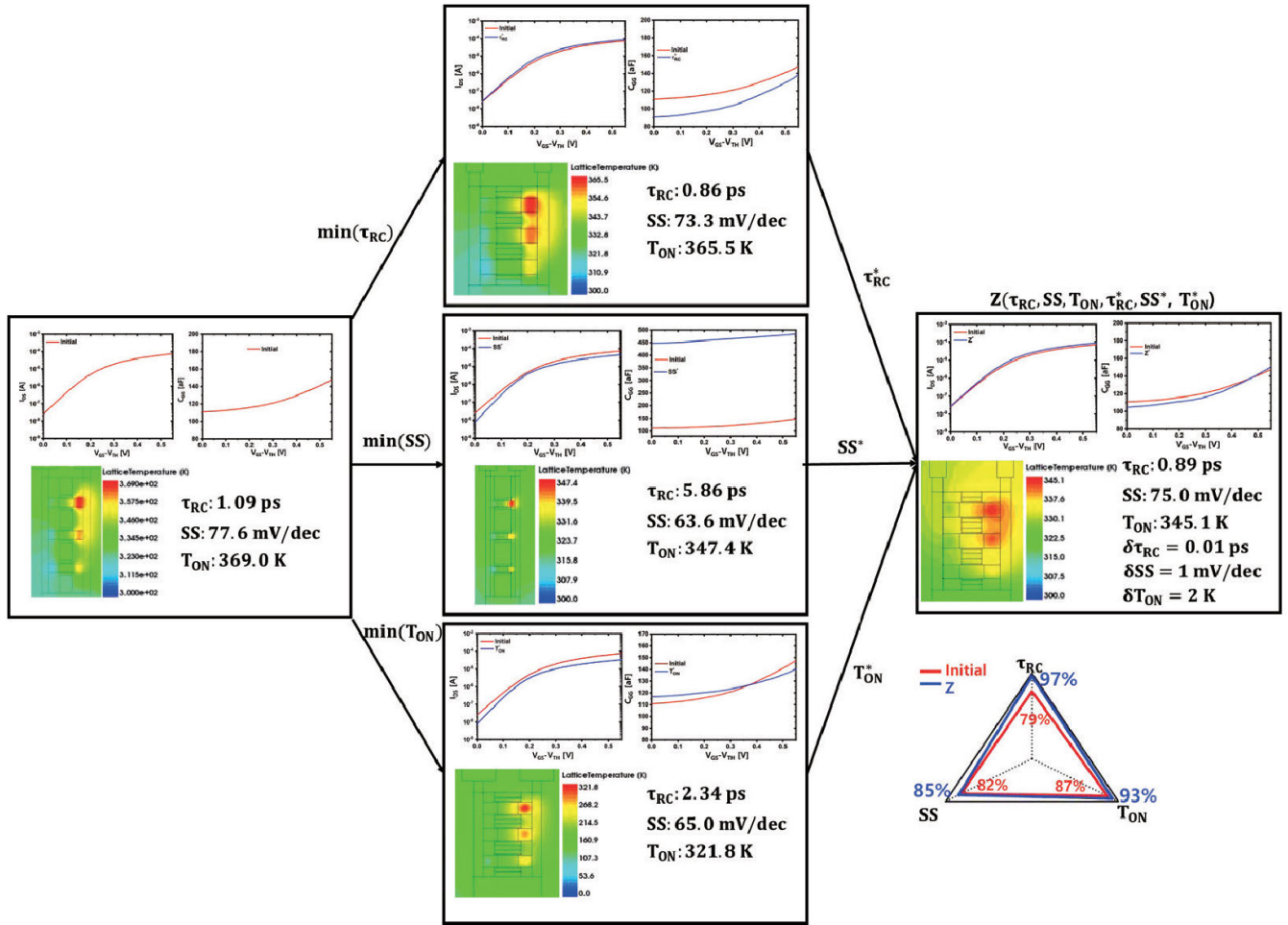
FIGURE 11. The schematic view of multi-objective optimization of 3-nm NS FET. First, 3 FOMs are independently optimized from the initial device. Using the optimized values from the single-objective optimization, 3 FOMs are jointly optimized. I_{DS}-V_{GS}, C_{GG}-V_{GS}, and the lattice temperature of the initial and optimized devices are drawn in the boxes. Comparison between the initial and the optimal device are drawn with triangles.

TABLE 5. Device specification in 3-nm node NS FET optimization.

Device	L_{sp} [nm]	W_{SH} [nm]	T_G [nm]	r_{via} [nm]	ϵ_{sp} [ϵ_0]	τ_{RC} [ps]	SS [mV/dec]	T_{ON} [K]
Initial	6.5	37	10	7	1	1.09	77.6	369.0
τ_{RC}^*	5.5	40	2	3	1	0.86	73.3	365.5
SS*	3	8	18	7	22	5.86	63.6	347.4
T_{ON}^*	4.5	8	2	7	7.5	2.34	65.0	321.8
$\mathbf{Z}^*$	6.5	40	2	3	7.5	0.89	75.0	345.1

As shown in the previous work [21], we have drawn the 5D optimization trajectories of NS FET in 3D space (Figs. 12(a)-(d)). Different input points in 5D space are overlapped when they have same input parameters in 3D space. We have also depicted the distribution of EI values at the last iteration in 2D plane (Figs. 12(e)-(h)). When 5D input parameters related to EI values have same input parameters in 2D space, only the maximum EI value of 2D point was plotted in 2D plane. As shown in Figs. 12(a)-(d), device input parameters were distributed around the optimized points tabulated in Table 5. This can be explained by distribution of EI values because input parameters which were expected to improve FOMs were weighted with high EI values as

the number of training data increased and input parameters with high EI values were sampled during the optimization. In τ_{RC} minimization, input parameters were concentrated on W_{SH} of 40 nm, T_G of 2 nm, ϵ_{sp} of $1\epsilon_0$, L_{sp} of 5.5 nm, and $r_{via} < 4$ nm (Fig. 12(a)). This distribution was similar to the distribution of high EI values. EI values at the last iteration were high at wide W_{SH}, thin T_G, low ϵ_{sp}, L_{sp} of 6 nm, and r_{via} of 3.5 nm (Fig. 12(e)).

In SS minimization, input parameters were distributed around W_{SH} of 8 nm, $T_G > 10$ nm, $\epsilon_{sp} \geq 7.5\epsilon_0$, and L_{sp} of 3 nm with large variation of r_{via} (Fig. 12(b)). High EI values at the last iteration were calculated around W_{SH} of 8 nm, T_G of 14-18 nm, $\epsilon_{sp} \geq 7.5\epsilon_0$, L_{sp} of 3 nm, and almost
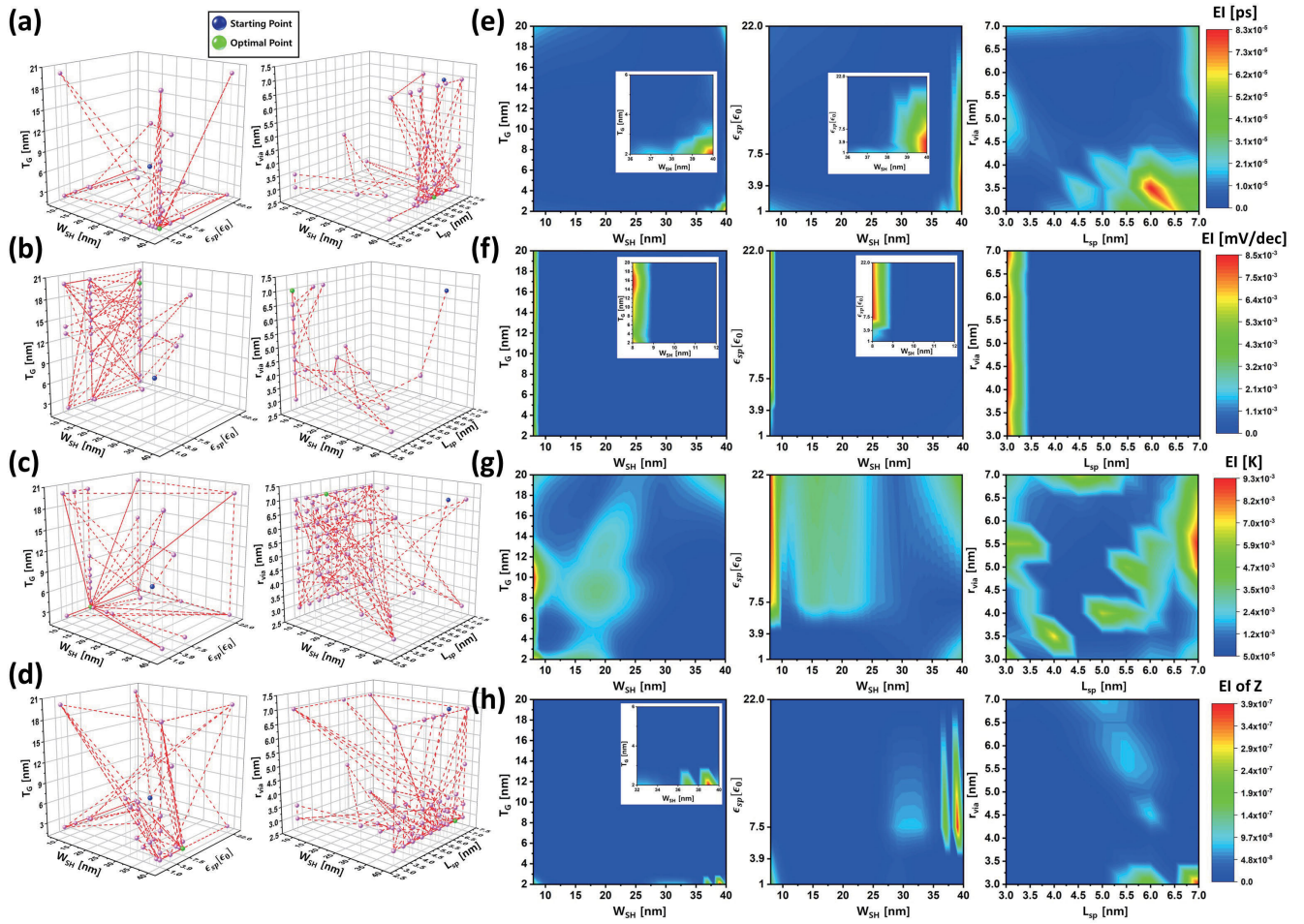
FIGURE 12. 5D optimization trajectories of NS FET in 3D space when it comes to (a) τ_{RC}, (b) SS, (c) T_{ON} and (d) Z optimization. Starting point is marked with blue sphere and the optimal point is marked with green sphere. Different input points in 5D space are overlapped when they have same input parameters in 3D space. Input parameters were distributed around the optimal point during optimization. EI values at the last iteration in 2D plane when it comes to (e) τ_{RC}, (f) SS, (g) T_{ON} and (h) Z optimization. When 5D input parameters related to EI values have same input parameters at each coordinate of 2D plane, only the maximum EI value of 2D point was plotted in 2D plane. Inset figures depict enlarged EI distribution around high values of EI.

all range of r_{via} (Fig. 12(f)). Because all range of r_{via} were evaluated with similar EI values, there were large variation of r_{via} (Fig. 12(b)).

In T_{ON} minimization, input parameters were populated around W_{SH} of 8 nm, T_G of 2 nm, and ϵ_{sp} of $7.5\epsilon_0$ with large variation of L_{sp} and r_{via} (Fig. 12(c)). High EI values were evaluated around W_{SH} of 8 nm, T_G of 2 or 10 nm, and $\epsilon_{sp} \geq 7.5\epsilon_0$ (Fig. 12(g)). The scattered high EI values (yellow color) in $L_{sp} - r_{via}$ plane explained the large variations of two parameters (Fig. 12(c)).

In **Z** maximization, input parameters were crowded around W_{SH} of 40 nm, T_G of 2 nm, ϵ_{sp} of $7.5\epsilon_0$ r_{via} of 3 nm with large variation of L_{sp} (Fig. 12(d)). The distribution can be attributed to high EI values near wide W_{SH}, T_G of 2 nm, ϵ_{sp} of $7.5\epsilon_0$, and r_{via} of 3 nm (Fig. 12(h)). High EI values at L_{sp} of 7 nm at the last iteration implied that EI values were highly evaluated near the optimum (L_{sp} of 6.5 nm).

All the MOSFET optimization tasks utilizing our framework required a small number of training data

(<200 observations). It could be attributed to the sample-efficiency of BO. As shown in Figs. 12(a)-(d), training data were sampled around the optimum during optimization. Therefore, BO required a tiny portion of input space because only the data near the optimum were needed for training.

As shown in Fig. 13, we have depicted 95% confidence interval of the prediction model. τ_{RC}, SS, T_{ON} and **Z** were optimized using GPR and stopped at ESC as shown in Fig. 10. Most of observed values were in the prediction bound (95% confidence interval). Especially, 100% of observed values of SS during optimization were in the prediction bound and almost match prediction mean when $T \geq 6$. As T increases, prediction mean got closer to observed data due to increased training data. When $T = 2$, observed values were far from the prediction mean values in Fig. 13, but they got more closer to the prediction mean values when $T > 7$. Observed data out of the prediction bound helped model to improve its prediction accuracy at the next iteration by changing the hyperparameters of the prediction model.
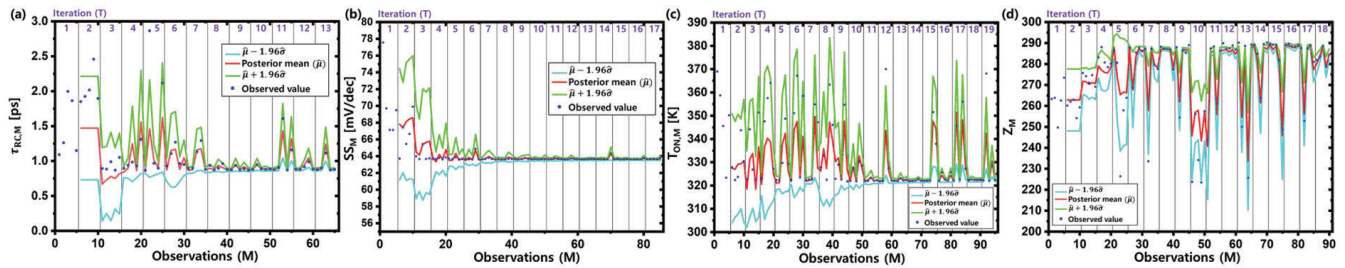
FIGURE 13. Posterior distribution of NS FET optimization when it comes to (a) τ_{RC}, (b) SS, (c) T_{ON} and (d) Z optimization. GPR and ESC were applied for the optimization. posterior mean ($\hat{\mu}(x)$), with upper/lower bound ($\hat{\mu}(x) \pm 1.96\hat{\sigma}(x)$) are plotted to illustrate 95% confidence interval. Posterior values at T iteration were predicted at T-1 iteration. Observed values are plotted with posterior values. Most of observed values were in the prediction bound. As T increased, prediction mean values got closer to observed values due to improvement of model accuracy.

Therefore, we found our model assumption in (8) was a reasonable one for the BO of NS FETs.

We found that our framework, the Bayesian optimization of the MOSFET device with ESC which maximized the efficiency while not impairing the reliability, can significantly accelerate the development of new transistor devices in the semiconductor industry. Our framework was extremely efficient, because it required a tiny number of training data (0.05% of input space) and small N_{iter} ($\ll 100$).

IV. CONCLUSION

We have systematically investigated the ESC of the Bayesian optimization when applied to the MOSFET design. By combining TCAD simulations with BO, we used ESC as the stopping condition for the BO. The ESC values were tested and explored by utilizing extensively large numbers of data of SG nMOSFETs. We found that 1% of unit value was the most efficient and reliable ESC for MOSFET optimization. Compared with the fixed iteration method and the tiny constant method, our ESC can reduce the number of BO iterations by up to 87.6% and 47%, respectively. By qualitatively analyzing optimization trajectories, we found that in MOSFET optimization BO required a small number of training data ($\ll 1,000$) because training data were efficiently sampled around the optimum during optimization. As an example application to a highly sophisticated, advanced CMOS device, 3-nm node NS FET was optimized by our framework, where the τ_{RC}, SS, and T_{ON} were minimized via multi-objective optimization. Our framework can boost the BO in the MOSFET design and accelerate the rapid development of devices to fulfill industrial requirements. In this work, we limited out study of ESC to MOSFET optimization only. However, our 1%-of-unit-value rule has potential to improve efficiency and reliability of BO in other engineering problems in general.

REFERENCES

[1] *Internationals Roadmap for Devices and System.* 2020. [Online]. Available: https://irds.ieee.org/

[2] Y. M. Lee, M. H. Na, A. Chu, A. Young, T. Hook, L. Liebmann, E. J. Nowak, S. H. Baek, R. Sengupta, H. Trombley, and X. Miao, "Accurate performance evaluation for the horizontal nanosheet standard-cell design space beyond 7 nm technology," in *IEDM Tech. Dig.*, Dec. 2017, pp. 29.3.1–29.3.4, doi: 10.1109/IEDM.2017.8268474.

[3] 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.

[4] D. Jang, D. Yakimets, G. Eneman, P. Schuddinck, M. G. Bardon, P. Raghavan, A. Spessot, D. Verkest, and A. Mocuta, "Device exploration of nanosheet transistors for sub-7-nm technology node," *IEEE Trans. Electron Devices*, vol. 64, no. 6, pp. 2707–2713, Jun, 2017, doi: 10.1109/TED.2017.2695455.

[5] L. Cai, W. Chen, G. Du, X. Zhang, and X. Liu, "Layout design correlated with self-heating effect in stacked nanosheet transistors," *IEEE Trans. Electron Devices*, vol. 65, no. 6, pp. 2647–2653, Jun. 2018, doi: 10.1109/TED.2018.2825498.

[6] H. Kim, D. Son, I. Myeong, J. Park, M. Kang, J. Jeon, and H. Shin, "Optimization of stacked nanoplate FET for 3-nm node," *IEEE Trans. Electron Devices*, vol. 67, no. 4, pp. 1537–1541, Apr. 2020, doi: 10.1109/ted.2020.2976041.

[7] D. Ryu, I. Myeong, J. K. Lee, M. Kang, J. Jeon, and H. Shin, "Investigation of gate sidewall spacer optimization from off-state leakage current perspective in 3-nm node device," *IEEE Trans. Electron Devices*, vol. 66, no. 6, pp. 2532–2537, Jun. 2019, doi: 10.1109/TED.2019.2912394.

[8] H. Kim, D. Son, I. Myeong, D. Ryu, J. Park, M. Kang, J. Jeon, and H. Shin, "Strain engineering for 3.5-nm node in stacked-nanoplate FET," *IEEE Trans. Electron Devices*, vol. 66, no. 7, pp. 2898–2903, Jul. 2019, doi: 10.1109/ted.2019.2917503.

[9] T. Ma, V. Moroz, R. Borges, K. E. Sayed, P. Asenov, and A. Asenov, "(Invited) future perspectives of TCAD in the industry," in *Proc. Int. Conf. Simul. Semiconductor Processes Devices (SISPAD)*, Sep. 2016, pp. 335–339, doi: 10.1109/sispad.2016.7605215.

[10] J. Viraraghavan, S. J. Pandharpure, and J. Watts, "Statistical compact model extraction: A neural network approach," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 12, pp. 1920–1924, Dec. 2012, doi: 10.1109/TCAD.2012.2207955.

[11] I. Stevanovic and C. C. McAndrew, "Quadratic backward propagation of variance for nonlinear statistical circuit modeling," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 9, pp. 1428–1432, Sep. 2009, doi: 10.1109/TCAD.2009.2023194.

[12] P.-H. Su and Y. Li, "A systematic approach to correlation analysis of in-line process parameters for process variation effect on electrical characteristic of 16-nm HKMG bulk FinFET devices," *IEEE Trans. Semiconductor Manuf.*, vol. 29, no. 3, pp. 209–216, Aug. 2016, doi: 10.1109/tsm.2016.2585129.

[13] M.-H. Oh, M.-W. Kwon, K. Park, and B.-G. Park, "Sensitivity analysis based on neural network for optimizing device characteristics," *IEEE Electron Device Lett.*, vol. 41, no. 10, pp. 1548–1551, Oct. 2020, doi: 10.1109/led.2020.3016119.

[14] J. Chen, M. B. Alawieh, Y. Lin, M. Zhang, J. Zhang, Y. Guo, and D. Z. Pan, "Automatic selection of structure parameters of silicon on insulator lateral power device using Bayesian optimization," *IEEE Electron Device Lett.*, vol. 41, no. 9, pp. 1288–1291, Sep. 2020, doi: 10.1109/led.2020.3013571.

[15] J. Chen, M. B. Alawieh, Y. Lin, M. Zhang, J. Zhang, Y. Guo, and D. Z. Pan, "PowerNet: SOI lateral power device breakdown prediction with deep neural networks," *IEEE Access*, vol. 8, pp. 25372–25382, 2020, doi: 10.1109/ACCESS.2020.2970966.

[16] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016, doi: 10.1109/JPROC.2015.2494218.

[17] S. Greenhill, S. Rana, S. Gupta, P. Vellanki, and S. Venkatesh, "Bayesian optimization for adaptive experimental design: A review," *IEEE Access*, vol. 8, pp. 13937–13948, 2020, doi: 10.1109/ACCESS.2020.2966228.

[18] M. J. Powell, "A view of algorithms for optimization without derivatives," *Math. Today-Bull. Inst. Math. Appl.*, vol. 43, no. 5, pp. 170–174, 2007.

[19] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, Sep. 2000, doi: 10.1109/4235.873238.

[20] D. R. Jones, C. D. Pertunnen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *J. Optim. Theory Appl.*, vol. 79, no. 1, pp. 157–181, 1993, doi: 10.1007/BF00941892.

[21] S. J. Park, B. Bae, J. Kim, and M. Swaminathan, "Application of machine learning for optimization of 3-D integrated circuits and systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 6, pp. 1856–1865, Jun. 2017, doi: 10.1109/TVLSI.2017.2656843.

[22] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Batch Bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design," in *Proc. 35th Int. Conf. Mach. Learn.* in Proceedings of Machine Learning Research, vol. 80, J. Dy and A. Krause, Eds. Stockholmsmässan, Sweden: PMLR, Jul. 2018, pp. 3306–3314. [Online]. Available: http://proceedings.mlr.press/v80/lyu18a.html

[23] *Global Optimization Tool Box in MATLAB*. Accessed: Feb. 1, 2021. [Online]. Available: http://mathworks.com/products/global-optimization.html

[24] V. Nguyen, S. Gupta, S. Rana, C. Li, and S. Venkatesh, "Regret for expected improvement over the best-observed value and stopping condition," in *Proc. 9th Asian Conf. Mach. Learn.* in Proceedings of Machine Learning Research, vol. 77, M.-L. Zhang and Y.-K. Noh, Eds. Stockholmsmässan, Sweden: PMLR, Nov. 2017, pp. 279–294. [Online]. Available: http://proceedings.mlr.press/v77/nguyen17a.html

[25] J. Zhang, A. A. Taflanidis, and J. C. Medina, "Sequential approximate optimization for design under uncertainty problems utilizing kriging metamodeling in augmented input space," *Comput. Methods Appl. Mech. Eng.*, vol. 315, pp. 369–395, Mar. 2017, doi: 10.1016/j.cma.2016.10.042.

[26] D. Drignei, "An estimation algorithm for fast kriging surrogates of computer models with unstructured multiple outputs," *Comput. Methods Appl. Mech. Eng.*, vol. 321, pp. 35–45, Jul. 2017, doi: 10.1016/j.cma.2017.04.001.

[27] V. Picheny and D. Ginsbourger, "Noisy kriging-based optimization methods: A unified implementation within the DiceOptim package," *Comput. Statist. Data Anal.*, vol. 71, pp. 1035–1053, Mar. 2014, doi: 10.1016/j.csda.2013.03.018.

[28] J. Jung, J. I. Yoon, H. K. Park, J. Y. Kim, and H. S. Kim, "An efficient machine learning approach to establish structure-property linkages," *Comput. Mater. Sci.*, vol. 156, pp. 17–25, Jan. 2019, doi: 10.1016/j.commatsci.2018.09.034.

[29] T. Ueno, T. D. Rhone, Z. Hou, T. Mizoguchi, and K. Tsuda, "COMBO: An efficient Bayesian optimization library for materials science," *Mater. Discovery*, vol. 4, pp. 18–21, Jun. 2016, doi: 10.1016/j.md.2016.04.001.

[30] H. M. Torun and M. Swaminathan, "High-dimensional global optimization method for high-frequency electronic design," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 6, pp. 2128–2142, Jun. 2019, doi: 10.1109/TMTT.2019.2915298.

[31] S. Ju, T. Shiga, L. Feng, Z. Hou, K. Tsuda, and J. Shiomi, "Designing nanostructures for phonon transport via Bayesian optimization," *Phys. Rev. X*, vol. 7, no. 2, May 2017, Art. no. 021024, doi: 10.1103/physrevx.7.021024.

[32] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2007, pp. 1177–1184. [Online]. Available: https://papers.nips.cc/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf

[33] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. vol. 25. Red Hook, NY, USA: Curran Associates, 2012, pp. 2951–2959. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf

[34] Z. Yang, A. Wilson, A. Smola, and L. Song, "A la carte–learning fast kernels," in *Proc. Artif. Intell. Statist.*, 2015, pp. 1098–1106. [Online]. Available: http://proceedings.mlr.press/v38/yang15b.pdf

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[36] H. Zhang, G. Liu, T. W. S. Chow, and W. Liu, "Textual and visual content-based anti-phishing: A Bayesian approach," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1532–1546, Oct. 2011.

[37] *ATLAS User's Manual*. (2016). [Online]. Available: https://dynamic.silvaco.com/dynamicweb/jsp/downloads/DownloadManualsAction.do?req=silen-manuals&nm=atlas

[38] Y.-C. Wu and Y.-R. Jhan, "Introduction of synopsys sentaurus TCAD simulation," in *3D TCAD Simulation for CMOS Nanoeletronic Devices*. Singapore: Springer, 2018, pp. 1–17, doi: 10.1007/978-981-10-3066-6_1.

[39] C. Lombardi, S. Manzini, A. Saporito, and M. Vanzi, "A physically based mobility model for numerical simulation of nonplanar devices," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. CADICS-7, no. 11, pp. 1164–1171, Nov. 1988, doi: 10.1109/43.9186.

[40] R. H. Fowler and L. Nordheim, "Electron emission in intense electric fields," *Proc. R. Soc. Lond. A, Math. Phys. Sci.*, vol. 119, no. 781, pp. 173–181, May 1928, doi: 10.1098/rspa.1928.0091.

[41] D. D. D. Ma, C. Lee, F. C. F. Au, S. Y. Tong, and S. T. Lee, "Small-diameter silicon nanowire surfaces," *Science*, vol. 299, no. 5614, pp. 1874–1877, 2003, doi: 10.1126/science.1080313.

[42] Y. Collette and P. Siarry, *Multiobjective Optimization: Principles and Case Studies*. Springer, 2013, doi: 10.1007/978-3-662-08883-8.

BOKYEOM KIM received the B.S. degree in electronic and electrical engineering from Sungkyunkwan University, Suwon, South Korea, in 2017. He is currently pursuing the combined master's and Ph.D. degree with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea.

His current research interest includes simulation and optimization of nanoscale electronic devices.

MINCHEOL SHIN received the B.S. degree in physics from Seoul National University, Seoul, South Korea, in 1988, and the Ph.D. degree in physics from Northwestern University, Evanston, IL, USA, in 1992.

He is currently a Professor with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea.