An Adaptive Machine Learning Method Based on Finite Element Analysis for Ultra Low-k Chip Package Design

Weishen Chu, Paul S. Ho, *Life Fellow, IEEE*, and Wei Li

***Abstract*— Machine learning (ML) is widely used for building data-driven models that are highly useful for optimization. In this study, a finite element model-based adaptive ML method is presented for chip package reliability prediction and design optimization. This ML method employs a validated multi-scale finite element model for training data generation. An adaptive sampling scheme is developed to optimize the training process with a steepest descent algorithm. The developed method was used to optimize ultra low-k chip package design. The effects of ten key design parameters on chip packaging reliability were considered. Multiple ML algorithms were evaluated for model development. It is shown that the adaptive sampling method performs much better than existing sequential sampling methods and that the finite element-based ML model can be used to achieve improved prediction accuracy for chip package design optimization.**

***Index Terms*— Adaptive sampling, chip packaging reliability, design optimization, machine learning (ML), steepest descent algorithm, ultra low-k.**

I. Introduction

CHIP package interaction (CPI) is one of the critical reliability issues for flip-chip packaging of advanced Cu/low-k chips. Finite element analysis (FEA) has been employed to reduce the time and cost of reliability assessment and to optimize the chip design [1]–[3]. These FEA-based methods can be grouped into three categories: trial-and-error studies, factorial design of experiments, and machine learning (ML)-assisted optimization [4], [5]. Most of the FEA-based methods involve a trial-and-error approach to investigating the effect of design parameters [6]–[8]. This approach provides a basic understanding of the effect of certain individual design parameters; however, the interaction among them is often overlooked. To consider interaction effects, full factorial designs of the experiment have been conducted with FEA models [9]–[12]. Due to the complexity of advanced chip design, a multi-scale modeling technique must be employed in the FEA to model packaging features from the millimeter to the nanometer scale [13]. Considering the number of design parameters that could be involved in chip packaging, a factorial design of computer experiments can be highly time-consuming, as the computation time grows exponentially with the number of design parameters.

A recent study [14] introduced an ML method for chip package design optimization with the assistance of FEA. An experimental validated FEA model was applied to generate training and testing data with respect to three selected chip package design parameters. An artificial neural network (ANN) model was trained and validated and used to assess the thermal-residual strain in the chip package. Although it provided a novel approach to evaluating chip-package reliability, this FEA-based ML method used a procedure to generate datasets in an *ad hoc* manner, which is less efficient and will quickly become intractable when the number of chip design parameters becomes large.

It has been long understood that computer experiments do not require replicates because there is no real experimental error involved in computer simulation. Thus, a quasi-random space-filling sampling scheme, such as the latin hypercube sampling (LHS) approach, is much more efficient than the factorial design of experiment. In fact, LHS has been the preferred statistical sampling method for generating training data with computer models. Sheikholeslami *et al.* [15] developed a sequential LHS scheme to increase the space-filling density by recursively subdividing the intervals of each design parameter. It can be used for generating training datasets systematically; however, the number of samples is doubled each time after one iteration and the algorithm does not provide an optimal training solution.

In this article, we develop a novel adaptive sampling strategy for developing ML models based on a multi-scale FEA model for ultra low-k chip package design optimization. A validated multi-scale finite element model was used for training data generation according to the adaptive sampling strategy. Ten key design parameters were selected from the chip-package level to the Cu interconnect level. Different ML algorithms were evaluated for building a model that provides the best predictability. The performance of the proposed adaptive sampling method is demonstrated with design optimization results.

II. Development of the FEA-Based Adaptive ML Algorithm

Fig. 1 shows a flowchart of the adaptive ML algorithm developed in this study. A validated FEA model that predicts

Manuscript received December 3, 2020; revised June 19, 2021 and July 12, 2021; accepted July 31, 2021. Date of publication August 6, 2021; date of current version September 10, 2021. Recommended for publication by Associate Editor K. Ramakrishna upon evaluation of reviewers' comments. *(Corresponding author: Weishen Chu.)*

The authors are with Walker Department of Mechanical Engineering, the University of Texas at Austin, Austin, TX 78712 USA (e-mail: weiwli@austin.utexas.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCPMT.2021.3102891.

Digital Object Identifier 10.1109/TCPMT.2021.3102891

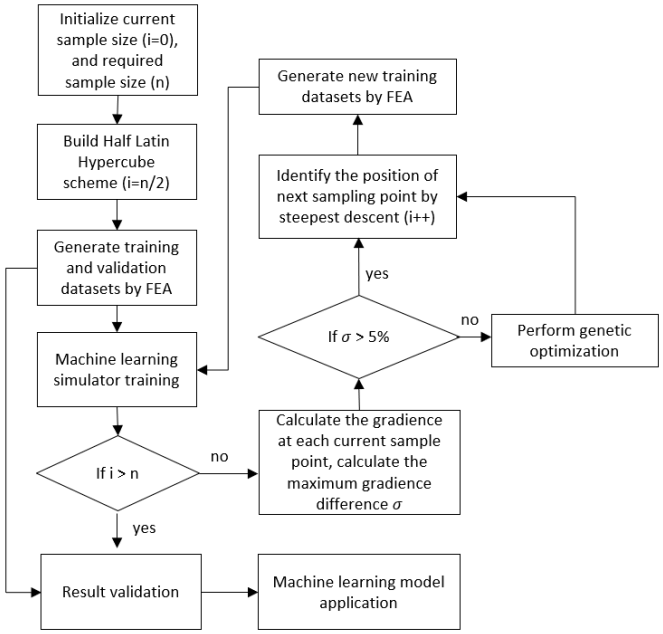

Fig. 1. FEA-based ML method with an adaptive sampling strategy.

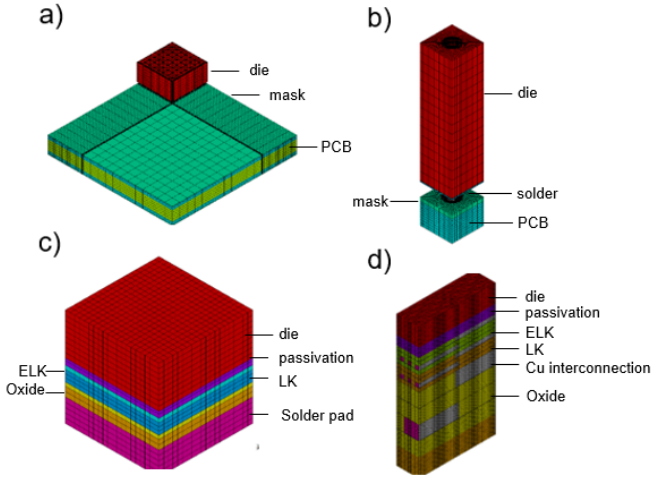

Fig. 2. Illustration of the multi-scale FEA model. (a) Level 1: chip-package level. (b) Level 2: the critical bump level with a zoomed-in view schematic of packaging design. (c) Level 3: the die-solder interface level. (d) Level 4: interconnect level where a crack is introduced between wiring layers to calculate the crack driving force [13].

the strain and stress distribution for a given chip package design serves as a data generator to produce training and validation datasets for the development of an ML model. The ML model is trained to predict the reliability of a chip package design, and further used to perform the chip design optimization. The training of the ML model relies on an adaptive sampling scheme that is developed in this study. It integrates a steepest descent optimization approach into a sequential LHS process to ensure the most efficient ML model training process. At each step of the training process, the ML model provides feedback to determine the next training dataset to generate with the FEA model. This process continues until a predetermined model accuracy criterion is achieved. The following sections discuss details of the proposed FEA-based adaptive ML method.

A. Multi-Scale FEA Model for Training Data Preparation

We applied a multi-scale FEA model [13] in ANSYS APDL with Newton–Raphson solver to serve as the training data generator for the ML model. Nonlinear analysis was performed to cover temperature-dependent cases. This FEA model, which has been validated with Moiré interferometry, assesses the reliability of chip package designs subjected to thermal cycling from 25 °C to 230 °C. The FEA simulated warpage is in good agreement with experimental measurement, and the crack energy release rate (ERR) calculation results matched with the experimental observations. Details of the FEA model validation are presented in [16]. A sub-modeling technique was used to model the chip package design from the package to Cu interconnection scales, as shown in Fig. 2. The material property data used in the FEA model are listed in Table I.

To evaluate the CPI-induced inner dielectric delamination, a crack is introduced at the interface of the extra low-k (ELK), and Cu interconnection layer on Level 4 of the model

TABLE I
MATERIAL PROPERTIES FOR THE MULTI-LEVEL FEA MODEL [13], [18]

Material	Young's Modules (GPa)	CTE (ppm/°C)	Poisson's ratio
Cu	117.0	16.7	0.34
Sn	162.5	3.2	0.29
Si	162	2.6	0.28
PCB core	33	80	0.30
PCB mask	4.1	30	0.40
passivation	66	0.57	0.18
Low-k dielectrcis	17	8.0	0.30
Ultra low-k dielectrics	4	18	0.30
IMC (Cu_3Sn)	143.0	18.2	0.30

to calculate the ERR at the crack tip using the modified virtual crack closure (MVCC) technique [17] as illustrated in Fig. 3. ERR has been considered a proper criterion for CPI reliability evaluation, and can be calculated by the tip force, tip displacement, and unit element area with

$$G_I = \frac{F_x^{(1)}\delta_x^{(2)}}{2\Delta A} \tag{1}$$

$$G_{\mathrm{II}} = \frac{F_y^{(1)}\delta_y^{(2)}}{2\Delta A} \tag{2}$$

$$G_{\mathrm{III}} = \frac{F_z^{(1)}\delta_z^{(2)}}{2\Delta A} \tag{3}$$

$$\mathrm{ERR} = G_{\mathrm{I}} + G_{\mathrm{II}} + G_{\mathrm{III}} \tag{4}$$

where G_{I}, G_{II}, and G_{III} are the x, y, and z components of the ERR, respectively, δ is the displacement of the crack tip, F is the tip force, and A is the unit element area for the corresponding components.

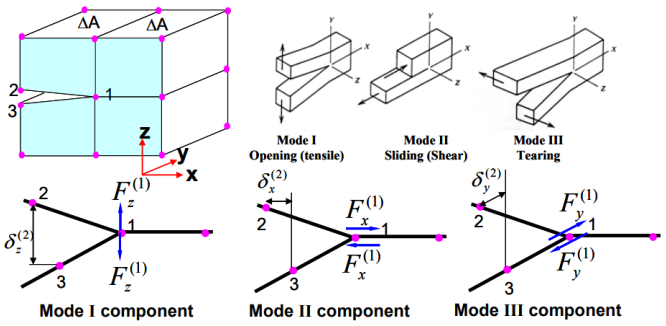

Fig. 3. Illustration of the MVCC technique to determine the ERR at an interface [19].

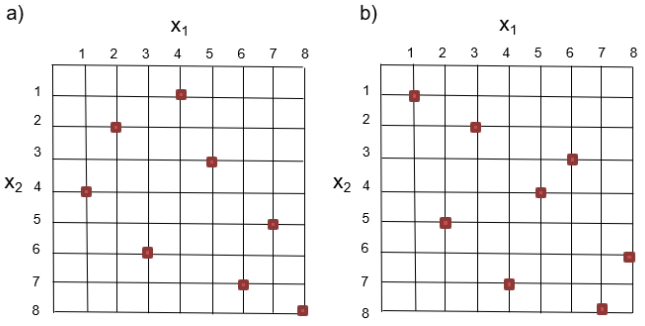

Fig. 4. Illustration of 2-D LHS with eight intervals. (a) One possible LHS scheme for two parameters with eight intervals. (b) Another possible LHS scheme for two parameters with eight intervals.

TABLE II
MINIMUM NUMBER OF SAMPLE POINTS REQUIRED BY LHS AND TWO-WAY FACTORIAL

No. of parameters	No. of samples required by LHS [21]	No. of samples required by two-way full factorial
2-7	17	4-128
8-11	33	256-2048
12-16	65	4096-65536
17-22	129	131072-4194304

B. LHS Scheme

The LHS scheme [20] was employed in this study to generate the initial datasets for the training of the ML model. According to [21], 33 samples are required by the LHS scheme for a design with ten parameters, comparing to 1024 samples if a two-way full factorial design were used. Table II shows a comparison of required samples between LHS and two-way full factorial design of the experiment.

The LHS scheme requires that each equal interval in the parameter space must be presented by a sole sample point. In a 2-D space as shown in Fig. 4, there must be exactly one sample in each row and column. The example shown in Fig. 4 is an LHS scheme for two parameters with eight intervals labeled from 1 to 8 each. One of the resulting LHS scheme has a sample set of $\{(x_1, x_2)|$ (1,4), (2,2), (3,6), (4,1), (5,3), (6,7), (7,5), (8,8)$\}$, as shown in Fig. 4(a). It is worth mentioning that the sampling scheme generated by LHS is not necessarily unique. The sample set $\{(x_1, x_2)|$ (1,1), (2,5), (3,2), (4,7), (5,4), (6,3), (7,8), (8,6)$\}$ shown in Fig. 4(b) is also a valid LHS for two parameters with eight intervals. This introduces randomness and uncertainty to the sampling scheme and further impacts the accuracy of the ML model. More details will be discussed in Section III.

C. Proposed Adaptive Sampling Approach

To increase the sampling efficiency, we introduce an adaptive sampling approach by starting with only one-half of the samples required by the LHS scheme and sequentially adding new samples one at a time with the assistance of the steepest descent optimization algorithm until the required number of samples by the LHS scheme is reached.

The initial scheme is developed following the procedure of orthogonal LHS scheme generation [21]. Instead of filling all required sample points in the orthogonal LHS scheme, our algorithm randomly selects one-half of the sample points from the orthogonal LHS scheme and then sequentially adds new sample points according to the steepest descent algorithm. In the steepest descent algorithm, the gradient at each existing sample point is calculated based on the response surface of a trained ML model. As an example of a 2-D space shown in Fig. 5(a), four sample points are initialized, upon which FEA simulations are conducted and an ML model is trained. The ML model is then used to determine the gradient at each point, as illustrated by the arrows in Fig. 5(a). The point with the largest gradient is identified, e.g., point (1, 1) in Fig. 5(a), and assigned as $P_s(x_1, x_2)$ as shown in (5). The next sample to include in the training dataset, $P_{s+1}(x_1, x_2)$, is then determined by (6), as illustrated by arrow I in Fig. 5(b). This process will be repeated as shown by arrows labeled II, III, and IV in Fig. 5(b) until the total number of sample points in the space reaches the required LHS sample size as listed in Table II.

$$P_s(x_1, x_2)\ s.f.\ \max\left(\left|\nabla f\left(x_1^i, x_2^i\right)\right|\right) \quad \text{for } i \epsilon L \tag{5}$$

$$P_{s+1}(x_1, x_2) = \left(x_1 + \frac{\frac{\partial f}{\partial x_1}(x_1, x_2)}{\max\left(\frac{\partial f}{\partial x_1}(x_1, x_2), \frac{\partial f}{\partial x_2}(x_1, x_2)\right)}\right)|e_1| \left(x_2 + \frac{\frac{\partial f}{\partial x_2}(x_1, x_2)}{\max\left(\frac{\partial f}{\partial x_1}(x_1, x_2), \frac{\partial f}{\partial x_2}(x_1, x_2)\right)}\right)|e_2| \tag{6}$$

where L is the set of sample points in the current scheme, x_n^i is the coordinate of the ith sample in dimension n, and e_n is the unit vector in dimension n.

In case the gradient values are close [22], a genetic algorithm [23] is applied to generate a new sample. For example, in Fig. 6(a) both points (2, 4) and (5, 5) have the same largest gradient magnitude. The genetic algorithm is then activated to switch the coordinates between them, resulting in two new sample points, (2, 5) and (5, 4), so that the adaptive sampling algorithm can continue.

D. ML Model

With the training data generated following the steepest descent adaptive sampling method, an ANN model is

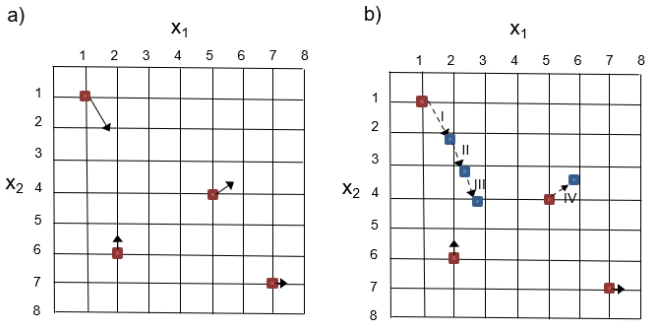

Fig. 5. Example of steepest descent adaptive sampling for a 2-D space with eight intervals. (a) Initial LHS map. (b) New sample points added by the steepest descent algorithm.

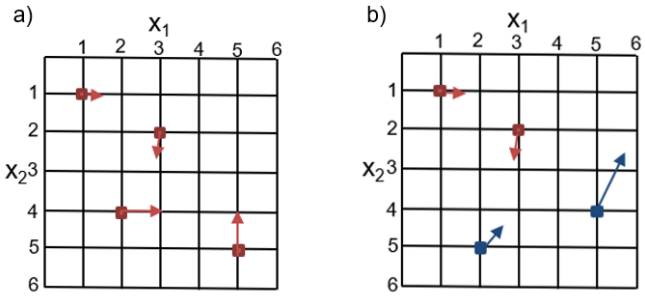

Fig. 6. Example of genetic algorithm for a 2-D space with six intervals. (a) Both points (2, 4) and (5, 5) are found to have the largest gradient magnitude. (b) New sample points (2, 5) and (5, 4) are generated by genetic algorithm.

developed to conduct ERR predictions. The structure of the ANN is shown in Fig. 7. The ANN model is trained by adjusting the weight and bias of each neuron so that the output of the model matches the expected output in the training dataset.

In this study, chip package design parameters are assigned to the input layer, ERR is assigned as the sole output to the output layer, and two hidden layers are used, each with ten neurons. Chip package design parameters are scaled to 0 to 1 before imported to the ANN model to meet the input range requirements. The sigmoid function, as shown in (7), is applied as the activation function of neurons. Standardization is employed to cover all possible ERR outputs and each hidden layer has been normalized to ensure that all inputs and outputs fit the range.

$$\varphi(x)=\frac{1}{1+e^{-x}} \tag{7}$$

where x is a linear combination of the input signals from previous neurons and $\varphi(x)$ is the output from the current neuron.

The signal transmission process from the jth hidden layer to the ith hidden layer follows [24]

$$h_k^i=\sum_{m=1}^{n}\varphi\left(w_k^i h_m^j+b_k^i\right) \tag{8}$$

where n is the number of neurons at the jth layer, h_m^i is the output signal from the mth neurons at the ith layer, w_k^i is

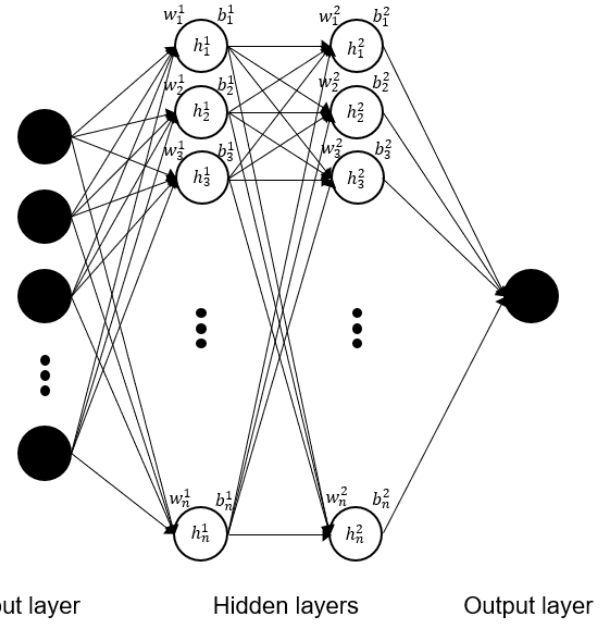

Fig. 7. Illustration of ANN with two hidden layers and single output neuron.

weighting of the kth neuron at the ith layer, and b_k^i is the bias of the kth neuron at the ith layer.

The training process is evaluated by a cross-entropy cost function [25] expressed in

$$\text{cost}(W, B)=-(y\ln(\hat{y})-(1-y)\ln(1-\hat{y})) \tag{9}$$

where W is the matrix of weights and B is the matrix of biases of hidden layers, y is the output from the training dataset, and $\hat{y}$ is predicted output by the ML model.

Werbos' backpropagation algorithm [26] is applied to recursively update the weight and bias to minimize the cross-entropy cost.

III. Validation of the Adaptive ML Method

A. Performance of the Proposed Adaptive LHS Scheme

We first compared the performance of the proposed adaptive LHS scheme with that of the traditional sequential LHS scheme [15] using the same training and validation datasets for an ANN model development. The sequential LHS scheme has been proven to be a more efficient and robust sampling strategy compared to full-factorial design [15]. Results are shown in Fig. 8. The root-mean-squared error (RMSE) of training is used as an evaluation criterion. The entire training and validation process was conducted twice for the traditional sequential LHS method. Due to the randomness involved in the traditional sequential LHS scheme, different RMSE results were found.

The results in Fig. 8 indicate that the performance of the adaptive sampling algorithm is more consistent than the sequential LHS sampling method. The final RMSE decreased from 2.20 to 1.96 for the adaptive sampling algorithm, suggesting that the feedback mechanism in the proposed method is able to add suitable sample points for the FEA model to

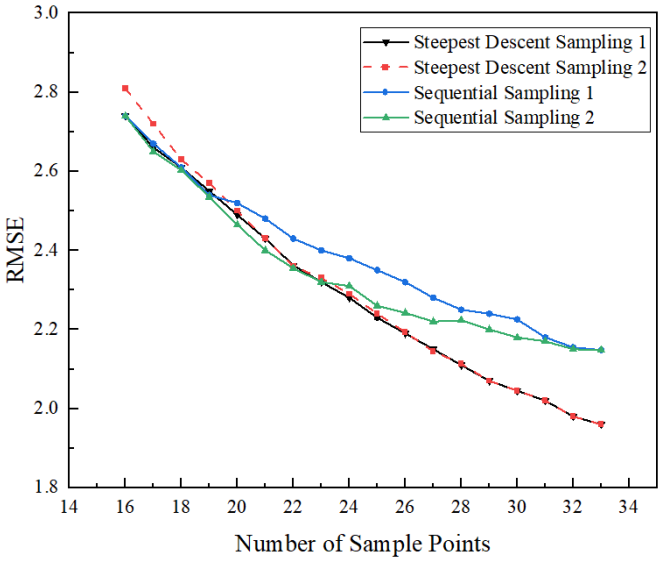

Fig. 8. Comparison between steepest descent and sequential sampling with respect to RMSE.

generate the most efficient training data at each step during the training process.

B. Evaluation of Different ML Models

To validate the efficiency of the neural network model, we examined two other major ML algorithms, including support vector machine (SVM) and random forest, for suitability of chip-package design optimization. The performances of the ML models were evaluated by *R*-squared values and RMSE. The sample points generated from the multi-scale FEA model according to the adaptive LHS scheme were applied as the training and validation dataset. We calculated RMSE with respect to the entire training datasets and the validation dataset for each of the ML algorithms, and the results are listed as RMSE-training and RMSE-validation in Table III. 18% of the entire dataset was used for validation and the other was used for training. Additional six cases with random input parameters that are independent of the training data were also created as the test dataset to evaluate the ML models. The results are listed as RMSE-test in Table III.

We found the ANN model has the lowest RMSE and highest R^2 for both the training and testing datasets, indicating that it is the most suitable model for this study. This is because chip package ERR modeling is a highly non-linear regression problem, while SVM is more suitable for unsupervised learning and the random forest method requires a larger number of training datasets and input features.

C. Comparison of ANN Model Prediction With FEA Results

The developed ANN ML model was applied to predict the ERR of different mixed-signal chip package designs. Table IV shows the first 16 cases generated by the multi-scale FEA model following the conventional LHS scheme. Those 16 cases from the FEA model were used for the initial training dataset. The input parameters of the FEA model have been scaled to a range from -16 to 16 in the LHS scheme. The corresponding predictions from the ML model are also included in the table. The two sets of prediction results agreed well.

TABLE III
PERFORMANCE EVALUATION ON ML MODELS

Algorithm	R^2	RMSE-training	RMSE-validation	RMSE-test
ANN	0.90	1.96	2.05	2.16
SVM	0.84	2.59	2.88	2.84
Random Forest	0.87	2.11	2.50	2.51

We also performed a parametric study on the effect of intermetallic compound (IMC) formation using the multi-scale FEA model and compared the results with the ones predicted by the ML model. As shown in Fig. 9, the thickness of IMC was varied from 0 to 1.25 μm for two different cases, i.e., 20% and 30% metal density in the LK layer. The ERR values were obtained from both the FEA and ML model.

The predicted ERR from the ML model is in good agreement with the out-of-sample FEA results. The maximum discrepancy is below 5%, indicating that the trained ML model is capable of performing ERR prediction as well as the FEA model. However, it should be noted that the computational efficiency of the ML model is dramatically higher than the multi-scale FEA simulation. This advantage of the ML model enables real-time design optimization.

IV. DESIGN OPTIMIZATION USING THE FEA-BASED ADAPTIVE ML MODEL

A. Optimization Methodology

The developed FEA-based adaptive ML method was used to optimize the design of an ultra low-k chip package. Considering the complexity of advanced low-k chip designs, materials, and process limitations, e.g., the minimum linewidth that can be achieved in fabrication, the sequential quadratic programming (SQP) technique, one of the most effective methods for nonlinear constrained optimization problems [27], was adopted in this study to conduct the design optimization. The SQP algorithm is provided below [28].

Algorithm 1 Sequential Quadratic Programming

Given initial x_0, λ_0, iterate for $k = 0, 1, 2, \ldots,$.
(a) *Compute* $\nabla f(x_k), c(x_k)$.
(b) *Solve the following inequality constrained QP to obtain* s_k
$min(s): \nabla f(x_k)^T s + \frac{1}{2} s^T (\nabla^2_{xx}(f(x) - \lambda^T c(x)))s$
subject to $\nabla c(x_k)^T + c(x_k) = 0$.
(c) *Set* $x_{k+1} \geq x_k + s_k$.
(d) *Return* (x_{k+1}, λ_{k+1}) *if satisfactory.*

where x_k and λ_k are unknown variables to solve, $f(x_k)$ is the objective function, $c(x_k)$ is the constrain function, and s is iteration step length.

Based on previous studies [29], we selected ten key design parameters that have the most significant impact on the chip packaging reliability and constrained the input range of each

TABLE IV
INPUT PARAMETERS AND ERR CALCULATIONS FROM THE MULTI-SCALE FEA AND THE ML MODEL

	Chip thickness	Chip dimension	PCB dimension	IMC thickness	Solder height	Metal Density_ LK	Metal Density_ Gap	Metal Density_ Analog	Metal Density_ Digital	Crack length	ERR-FEA J/m^2	ERR-Machine learning J/m^2
Range	0.1-5 mm	0.1-2 mm	1-10 mm	0-2 μm	10-100 μm	2-10%	2-10%	15-25%	15-25%	10-100 nm		
Case1	1	4	-8	-16	3	16	16	1	5	5	8.11	7.47
Case2	-1	3	-7	-15	-4	-8	15	5	14	6	9.49	10.14
Case3	4	-2	-6	-13	-1	5	14	-8	13	-7	6.98	7.82
Case4	-3	-1	-5	-14	2	-6	13	7	15	8	10.21	10.20
Case5	6	8	4	-12	7	3	-11	-2	16	-1	7.74	7.10
Case6	-5	7	3	-11	-8	-4	-12	1	10	-2	4.61	4.97
Case7	8	-6	2	-10	-5	-1	-9	-4	9	3	3.69	4.67
Case8	-7	-5	1	-9	6	2	-10	3	11	4	4.99	5.28
Case9	10	12	-16	8	11	-15	7	-13	12	-14	13.89	13.33
Case10	-9	11	-15	7	-12	-16	8	14	6	13	10.75	10.74
Case11	12	-10	-13	6	-9	-14	5	-15	8	-15	11.83	11.54
Case12	-11	-9	-14	5	10	-13	6	16	7	-16	13.48	13.19
Case13	13	16	12	4	15	11	-3	-10	2	9	5.81	4.41
Case14	-14	15	11	3	-16	-12	-4	9	1	10	6.45	6.39
Case15	16	-13	10	2	14	9	-1	-12	4	-11	6.95	6.57
Case16	-15	-14	9	1	13	-10	-2	11	3	12	13.62	13.63

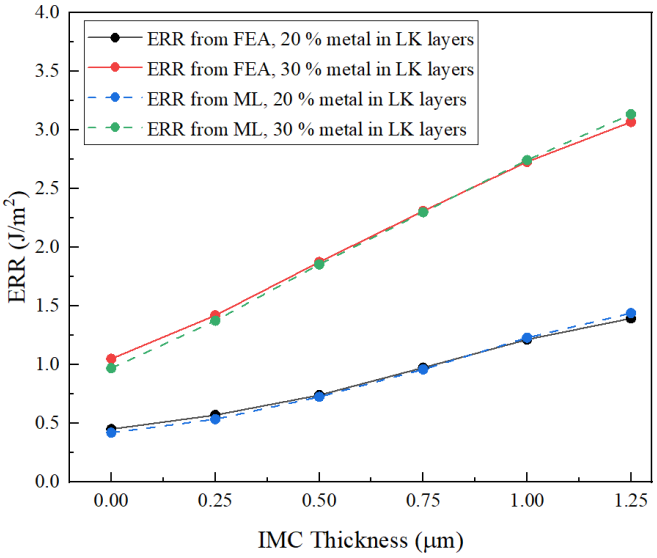

Fig. 9. Comparison of calculated ERR from the FEA model versus the ML model for cases with different metal densities.

TABLE V
RESULTS OF DESIGN OPTIMIZATION USING THE ADAPTIVE ML MODEL

Input parameter	Optimal value
Chip thickness	0.998 mm
Chip dimension	0.997 mm
PCB dimension	1.00 mm
IMC thickness	0 μm
Solder height	99.98 μm
Metal Density_LK	9.9%
Metal Density_Gap	9.9%
Metal Density_Analog	24.98%
Metal Density_Digital	24.98%
Crack length	10.0 nm
Optimized ERR	2.85 J/m^2

design parameter. The ranges of the design parameters are also listed in Table IV.

B. Design Optimization and Validation

Our FEA-based adaptive ML model was used to make ERR predictions and the SQP algorithm was performed to minimize the ERR for design optimization. The optimized results are shown in Table V.

The optimized ERR under the given constraints is 2.85 J/m^2, comparing to the FEA model prediction using the same input parameters as listed in Table V, which is 2.71 J/m^2. The optimal values obtained from the design optimization procedure suggest that the chip and PCB dimension should be as small as possible in order to reduce the ERR value. Similar findings have been reported by Yuan *et al.* [29], who have shown that strain and stress monotonically increase with the chip and PCB dimension. The optimization results also suggest that the metal density differences between analog, digital, and the gap portion of the chip should be minimized. This also agrees with the result of a previous study [13] where the local material mismatch is found to be the cause of high ERR values and should be avoided.

V. CONCLUSION

A finite element model-based adaptive ML method is developed for ultra low-k chip package reliability prediction and design optimization. An adaptive LHS scheme is proposed to optimize the training data generation process with a steepest descent algorithm. The method is validated with a complex chip design example, where ten key design parameters on chip packaging reliability were optimized. It is shown that the adaptive sampling method performs much better than existing sequential sampling methods. The ANN model performed better than other ML models such as

the SVM and random forest models. The proposed finite element-based ML model can be used to achieve accurate ERR prediction and enable real-time complex chip package design optimization.

REFERENCES

[1] L. Zhang, I. C. Ume, J. Gamalski, and K. P. Galuschki, "Study of flip chip solder joint cracks under temperature cycling using a laser ultrasound inspection system," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 32, no. 1, pp. 120–126, Mar. 2009.

[2] X. Zhang, S.-K. Ryu, R. Huang, P. S. Ho, J. Liu, and D. Toma, "Impact of process induced stresses and chip-packaging interaction on reliability of air-gap interconnects," in *Proc. Int. Interconnect Technol. Conf.*, Jun. 2008, pp. 135–137, doi: 10.1109/IITC.2008.4546947.

[3] W. D. van Driel, G. Q. Zhang, J. H. J. Janssen, and L. J. Ernst, "Response surface modeling for nonlinear packaging stresses," *J. Electron. Packag.*, vol. 125, no. 4, pp. 490–497, Dec. 2003, doi: 10.1115/1.1604149.

[4] F. Gao *et al.*, "AD-NET: Age-adjust neural network for improved MCI to AD conversion prediction," *NeuroImage, Clin.*, vol. 27, Jun. 2020, Art. no. 102290, doi: 10.1016/j.nicl.2020.102290.

[5] Z. Cui, C. Li, W. Dai, L. Zhang, and Y. Wu, "A hierarchical teaching-learning-based optimization algorithm for optimal design of hybrid active power filter," *IEEE Access*, vol. 8, pp. 143530–143544, 2020, doi: 10.1109/ACCESS.2020.2995716.

[6] C. J. Zhai, S. S. Too, and R. N. Master, "Reliability modeling of lidded flip chip packages," in *Proc. 57th Electron. Compon. Technol. Conf.*, May 2007, pp. 1091–1096, doi: 10.1109/ECTC.2007.373933.

[7] P. Lianto *et al.*, "CPI parametric investigation of UBM-Al interface for Cu pillar flip-chip application," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 6, no. 7, pp. 1120–1126, Jul. 2016, doi: 10.1109/TCPMT.2016.2571059.

[8] T. Sinha, S. Li, K. Tunga, J. A. Zitz, and K. K. Sikka, "Geometric optimization and mechanical risk mmitigation in 2.5 D flip-chip packages using parametric finite element analysis (FEA) simulations," in *Proc. 15th IEEE Intersociety Conf. Thermal Thermomechanical Phenomena Electron. Syst. (ITherm)*, May 2016, pp. 51–57, doi: 10.1109/ITHERM.2016.7517527.

[9] M. C. Yew, C. Yuan, C. Han, C. Huang, W. Yang, and K. N. Chiang, "Factorial analysis of chip-on-metal WLCSP technology with fan-out capability," in *Proc. 13th Int. Symp. Phys. Failure Anal. Integr. Circuits*, Jul. 2006, pp. 223–228, doi: 10.1109/IPFA.2006.251035.

[10] S. Raghavan, I. Schmadlak, G. Leal, and S. K. Sitaraman, "Study of chip–package interaction parameters on interlayer dielectric crack propagation," *IEEE Trans. Device Mater. Rel.*, vol. 14, no. 1, pp. 57–65, Nov. 2013, doi: 10.1109/TDMR.2013.2288255.

[11] L. Lin, J. Wang, L. Wang, and W. Zhang, "The stress analysis and parametric studies for the low-k layers of a chip in the flip-chip process," *Microelectron. Rel.*, vol. 65, pp. 198–204, Oct. 2016, doi: 10.1016/j.microrel.2016.09.001.

[12] J. S. Kang, "Parametric study of warpage in PBGA packages," *Int. J. Adv. Manuf. Technol.*, vol. 107, no. 9, pp. 4213–4219, 2020, doi: 10.1007/s00170-020-05256-3.

[13] W. Chu, L. Spinella, D. R. Shirley, and P. S. Ho, "Effects of wiring density and pillar structure on chip package interaction for advanced Cu low-k chips," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Apr. 2020, pp. 1–4, doi: 10.1109/IRPS45951.2020.9128333.

[14] C. C. A. Yuan and C.-C. Lee, "Solder joint reliability modeling by sequential artificial neural network for glass wafer level chip scale package," *IEEE Access*, vol. 8, pp. 143494–143501, 2020, doi: 10.1109/ACCESS.2020.3014156.

[15] R. Sheikholeslami and S. Razavi, "Progressive Latin hypercube sampling: An efficient approach for robust sampling-based analysis of environmental models," *Environ. Model. Softw.*, vol. 93, pp. 109–126, Jul. 2017, doi: 10.1016/j.envsoft.2017.03.010.

[16] W. Chu, T. Jiang, and P. S. Ho, "Effect of wiring density and pillar structure on chip packaging interaction for mixed-signal Cu low k chips," *IEEE Trans. Device Mater. Rel.*, early access, May 19, 2021, doi: 10.1109/TDMR.2021.3082043.

[17] R. Krueger, "Virtual crack closure technique: History, approach, and applications," *Appl. Mech. Rev.*, vol. 57, no. 2, pp. 109–143, Apr. 2004, doi: 10.1115/1.1595677.

[18] Y. Wang, "Electromigration and chip-package interaction reliability of flip chip packages with Cu pillar bumps," M.S. thesis, Dept. Mech. Eng., Univ. Texas Austin, Austin, TX, USA, 2011, vol. 78712.

[19] G. Wang, "Thermal deformation of electronic packages and packaging effect on reliability for copper/low-k interconnect structures," Ph.D. dissertation, Dept. Mech. Eng., Univ. Texas Austin, Austin, TX, USA, 2004, vol. 78712.

[20] S. S. Garud, I. A. Karimi, and M. Kraft, "Design of computer experiments: A review," *Comput. Chem. Eng.*, vol. 106, pp. 71–95, Nov. 2017, doi: 10.1016/j.compchemeng.2017.05.010.

[21] K. Q. Ye, "Orthogonal column Latin hypercubes and their application in computer experiments," *J. Amer. Statist. Assoc.*, vol. 93, no. 444, pp. 1430–1439, 1998, doi: 10.1080/01621459.1998.10473803.

[22] A. Nuchitprasittichai and S. Cremaschi, "An algorithm to determine sample sizes for optimization with artificial neural networks," *AIChE J.*, vol. 59, no. 3, pp. 805–812, Mar. 2013, doi: 10.1002/aic.13871.

[23] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks*. Cham, Switzerland: Springer, 2019, pp. 43–55, doi: 10.1007/978-3-319-93025-1_4.

[24] J. Zupan, "Introduction to artificial neural network (ANN) methods: What they are and how to use them," *Acta Chim. Slovenica*, vol. 41, p. 327, Jan. 1994.

[25] K. Gajowniczek, "Generalized entropy cost function in neural networks," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2017, pp. 128–136, doi: 10.1007/978-3-319-68612-7_15.

[26] S. Leven, "The roots of backpropagation: From ordered derivatives to neural networks and political forecasting," *Neural Netw.*, vol. 9, no. 3, pp. 543–544, Apr. 1996.

[27] R. Tauler and B. Walczak, *Comprehensive Chemometrics: Chemical and Biochemical Data Analysis*. Amsterdam, The Netherlands: Elsevier, 2009, doi: 10.1016/b978-044452701-1.09002-5.

[28] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numer.*, vol. 4, pp. 1–51, Jan. 1995, doi: 10.1017/S0962492900002518.

[29] C. A. Yuan, C. N. Han, and K. N. Chiang, "Design and analysis of novel glass WLCSP structure," in *Proc. 5th Int. Conf. Thermal Mech. Simul. Exp. Microelectron. Microsystems*, May 2004, pp. 279–285, doi: DOI:.10.1109/ESIME.2004.1304051.