Novel Partitioning-Based Approach for Electromigration Assessment with Neural Networks

Tianshu Hou, Farid N. Najm, *Fellow, IEEE*,
Ngai Wong, *Senior Member, IEEE* and Hai-Bao Chen, *Senior Member, IEEE*

***Abstract*—Due to continuing technology scaling, electromigration (EM) remains a prominent reliability concern in integrated circuit design. Traditional empirical methods often result in over-design in very large scale integration (VLSI) due to model inaccuracy. Recently, researchers have focused on analyzing EM susceptibility by tracking hydrostatic stress evolution in metal lines, governed by computationally expensive partial differential equations (PDEs). In this paper, we propose a partitioning-based approach using neural networks to efficiently forecast the stress evolution along interconnect trees during the void nucleation and growth phases. This approach begins by decomposing the interconnect tree into subcomponents, providing computationally efficient analytical solutions for predicting stress evolution within each subtree. Subsequently, we employ a lightweight neural network to reassemble these components with their corresponding solutions to the original structure, ensuring accurate stress prediction. This divide-and-conquer strategy can accommodate various tree structures, with offshoots at arbitrary junctions, and holds substantial promise for using NN-based methods to solve mesh-free stress evolution on much larger interconnect trees than previously possible, with reduced computational overhead and heightened accuracy. The proposed approach eliminates the need for time discretization and grid meshing typically required in numerical methods. Numerical results confirm its advantages in accuracy and computational efficiency.**

***Index Terms*—Electromigration, partitioning, interconnect tree, machine learning.**

I. Introduction

Electromigration (EM) remains a major circuit reliability concern in advanced technology. This concern arises from the reduced service lifetime of interconnects under high current density, resulting in voids, a reduction of cross-sectional area, and an increase in line resistance. Thus, EM verification is crucial for chip signoff in very large scale integration (VLSI).

Traditional EM models have predominantly relied on empirical approaches, which often lack the accuracy needed for complex multi-line structures. Based on the well-known Blech effect [1], a certain length-current density product can be used to identify EM-robust interconnects, called “immortal” wires. In additional, Black’s equation [2] provides an empirical model for the mean-time-to-failure (MTF) of isolated metal lines, as a relationship between EM failure time, current density, and temperature. However, applying these methods to multi-line metal structures implicitly assumes that individual lines may be modelled as independent isolated lines, which is not true and leads to inaccurate reliability evaluation when applied to on-chip circuits such as power grids [3]. As an alternative physical model, and building on previous work in [4]–[6], EM assessment based on Korhonen’s equation [7] effectively captures the physical kinetics. This equation describes the relationship between the current density-driven electron wind force and the diffusion gradient-driven back stress force along a wire, formulated as partial differential equations (PDEs). According to this model, tree-level analysis for EM-induced stress calculation has been proposed to consider the material flow between adjacent segments confined by diffusion barriers, rather than solely focusing on individual wires [8]–[10]. Solving the PDEs-governed physical model remains a challenge, especially as interconnect tree sizes grow. Numerical methods, such as the finite difference method (FDM) and finite element method (FEM), are typically employed for stress analysis, necessitating time discretization and mesh refinement [11], [12]. Analytical methods, on the other hand, yield closed-form solutions for hydrostatic stress via Laplace transformations [13]–[15], offering computational acceleration for specific interconnect structures. Recently, machine learning (ML) has emerged as a promising tool for modeling the evolution of EM-induced stress governed by complex PDEs.

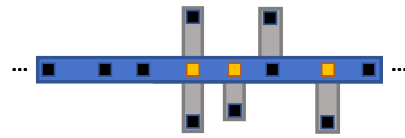


Fig. 1. Example of an interconnect tree consisting of a long stripe (blue) with occasional offshoots (grey) from certain junctions. Black blocks denote junctions in the interconnect tree; yellow blocks indicate selectable joints in the proposed approach.

To address the scalability issues in large interconnect trees, we partition the EM-induced stress problem and incorporate sub-problems. Partitioning techniques have traditionally been used in VLSI to accelerate the solution of large-scale problems by leveraging circuit locality. A large circuit can be divided into smaller sub-circuits connected at a few global boundaries. This divide-and-conquer strategy reduces computational complexity by manipulating the entire system into hierarchical tree structures after optimization, scaling, relaxation, partitioning, and assignment [16], [17].

This study focuses on EM assessment in power grids, where metal lines mostly carry unidirectional currents and are therefore susceptible to EM failure. Fig. 1 shows an example of a typical interconnect tree in modern power grids, where the tree consists of a long stripe (blue) with possibly occasional offshoots (grey) from certain junctions. We introduce a novel learning approach to calculate stress evolution along such interconnect trees with offshoots. By utilizing the partitioning-based method with neural networks and benefiting from the locality property in EM-induced stress evolution, satisfactory computational efficiency can be achieved without sacrificing accuracy. Compared with numerical methods, the proposed

This work was supported in part by the Nature Science Foundation of China under Grants T2293704 and T2293700, the Natural Sciences and Engineering Research Council of Canada (NSERC), the Theme-based Research Scheme (TRS) project T45-701/22-R of the Research Grants Council (RGC), Hong Kong SAR, and the China Scholarship Council (CSC), China. Corresponding author: Hai-Bao Chen.

T. Hou and H-B. Chen are with the Department of Micro/Nano Electronics, Shanghai Jiao Tong University.
F. N. Najm is with the Department of Electrical & Computer Engineering, University of Toronto.
N. Wong is with the Department of Electrical and Electronic Engineering, The University of Hong Kong.

model frees EM analysis from time discretization and grid meshing. Compared with analytical methods that are typically limited to interconnect trees with simple structures, the proposed method is applicable to more practical interconnect tree structures and provides closed-form expressions for stress evolution. Our main contributions include:

- We propose, for the first time, a fast partitioning-based approach to predict mesh-free stress evolution along interconnect trees during both the void nucleation and growth phases. The method is based on a neural network and requires no prior knowledge of the stress solution. Designed to align with EM analysis, the proposed framework considers physical parameters like varying segment width and random diffusivity, as well as operating conditions like time-dependent temperature and dynamic current densities caused by EM-induced metal line degradation.
- We introduce a novel divide-and-conquer strategy that enables multiple partitioning approaches for the original interconnect tree and derives computationally efficient stress solutions for the subtrees. This approach results in significant computational savings, with additional time reduction achieved through parallelization.
- We develop a learning approach to combine the stress solutions of the subtrees, leveraging the locality of EM stress evolution, which reduces the number of constraint conditions in the learning task compared with the existing methods. The proposed approach is extended to substantially improve the convergence speed by incorporating random Fourier feature embeddings.
- The proposed approach is validated using the FEM tool COMSOL, showing significant improvement in accuracy, performance, and scalability for EM assessment compared with existing methods. Further advantages are demonstrated by the numerical results.

The rest of the paper is structured as follows. Section II provides a brief overview of the related works, EM physics, and stress modeling, along with an introduction to spectral bias and random Fourier features. Section III outlines the partitioning method and derives the brick-joint solution. Section IV introduces the learning framework and extends it to solve stress evolution under dynamic current densities. Section V presents the results of the proposed method and compares its performance against competing methods. Finally, Section VI concludes the paper and the paper ends with an Appendix that includes additional technical details.

II. Preliminary

We will review several ML-based approaches for EM assessment, along with key fundamentals and preliminaries for modeling EM physics and establishing the neural network-based framework.

A. Related work

ML has shown effectiveness in modeling various physical systems, including capturing the evolution of EM-induced hydrostatic stress [18]–[23]. Recently, physics-informed neural networks (PINN) have been introduced to model physical dynamics by encoding constraints via PDEs [24], providing a novel approach to solving Korhonen's equation. Studies in [19], [20], [23] have shown promising results in ML-aided EM analysis by employing a coordinate-based neural representation to achieve stress evolution on interconnect trees with multiple segments. However, these methods encounter challenges as the number of segments increases and exhibit heavy training overhead. To mitigate this issue, a customized loss function has been proposed to accelerate training and enable stress analysis in larger-scale interconnect trees during the void nucleation phase [21], which has been further developed for the void growth phase in [22]. Although these methods can solve Korhonen's equation for interconnect trees, they remain limited by the scale of the trees.

B. EM physics and stress modeling

EM occurs within metal wires carrying significant current densities, where metal atoms are bombarded by the electron flow. This resulting atomic flow increases compressive stress at the anode and tensile stress at the cathode, presenting an opposing driving force that impedes EM progress [1]. Voids may form when stress levels surpass critical thresholds, potentially leading to circuit malfunction. The confinement of metal lines in modern dual damascene technology leads to void formation due to tensile stress, which is more likely than hillock formation. In this subsection, we first introduce the physics-based modeling describing the stress evolution and then present the concept of stress flow [14], which is employed to construct the analytical solution of stress evolution along linear interconnect trees during the void nucleation phase.

1) Physics-based modeling: The evolution of EM-induced stress can be characterized by the 1D Korhonen's equation [7], which combines the atomic flux equation tracking the atomic mass transport, and the continuity equation enforcing the mass balance. The atomic flux describing the migration of metal atoms within the confined line can be expressed as:

$$\phi(x) = \frac{DC}{k_b T}\left(\Omega \frac{\partial \sigma}{\partial x} + q^* \rho j\right), \tag{1}$$

where j and x represent the current density and location, respectively. The effective atomic diffusivity $D = D_0 \exp(-E_a/k_b T)$ depends on the activation energy E_a, the self-diffusion coefficient D_0, the Boltzmann constant k_b and the temperature T. The variables $C,\ q^*,\ \Omega$, and ρ denote the atomic concentration, effective charge, atomic lattice volume, and metal resistivity, respectively. Furthermore, the continuity equation, which accounts for the interplay between stress evolution over time and material transport, yields the diffusion equation:

$$\frac{\partial \sigma(x,t)}{\partial t} = \frac{\partial}{\partial x}\Big[\kappa\Big(\frac{\partial \sigma(x,t)}{\partial x} + G\Big)\Big], \tag{2}$$

where $\kappa = DB\Omega/k_b T$ and $G = q^* \rho j/\Omega$. The notation B is the effective bulk modulus associated with the line geometry, especially width, aspect ratio and grain morphology [25], [26].

Fig. 2 illustrates interconnect trees featuring degree-1 to 4 junctions, where the degree indicates the number of lines connecting to the junction (hollow squares). Boundary conditions (BCs) for the junctions should be established in the EM model. A preliminary step is to enforce the balance between the outgoing and incoming atomic flux at junctions with a degree greater than one (called interior junction), which we refer to as flux balance:

$$\sum_{i \in S_r} \pm W_i \cdot \kappa_i \Big(\frac{\partial \sigma_i(x,t)}{\partial x}\Big|_{x=x_r} + G_i\Big) = 0, \tag{3}$$

where the atomic flux term is positive in the incoming direction. W_i is the branch width of the i-th segment, S_r is the index set of all the segments connected to the r-th interior junction and Eq. (3) demonstrates the net atomic flux entering the r-th interior junction is zero. Moreover, the stress continuity

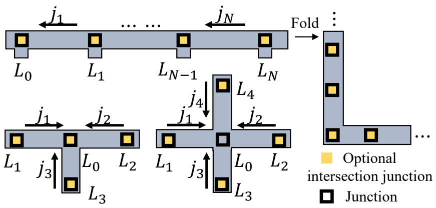


Fig. 2. The general straight multi-segment, T-shaped, and cross-shaped interconnect trees. A straight multi-segment brick can be transformed by folding the segments at the junctions. Notations L_k and j_i represent the location of the k-th node and the current density loaded within the i-th segment. L_0 is equal to zero. The hollow square represents the junction of the interconnect tree. The yellow square indicates an optional intersection junction within the partitions.

condition is enforced at the interior junctions, as the stress evolution must remain continuous along the tree:

$$\sigma_{S_r[1]}(x_r, t) = \cdots = \sigma_{S_r[m]}(x_r, t). \tag{4}$$

BC at degree-1 junctions (called terminal junction) and the initial condition for interconnect trees can be categorized into two cases: a) void nucleation phase when no EM-induced voiding occurs in metal lines and b) void growth phase when voids are created and develop over time:

- **void nucleation:** before void occurs, the atomic flux ϕ at the diffusion barrier equals zero. BC at terminals follows:

$$\kappa_t\Big(\frac{\partial\sigma_t^n(x,t)}{\partial x}\Big|_{x=x_t} + G_t\Big) = 0, \tag{5}$$

where σ^n shows the stress evolution during the void nucleation phase as indicated by the superscript n, and x_t denotes the location of the terminal. For each segment, the initial condition can be written as:

$$\sigma_i^n(x, 0) = \sigma_T, \tag{6}$$

where σ_T denotes the residual stress at the initial time, which is uniform as a result of stress migration occurring immediately after chip manufacturing.

- **void growth:** once a void occurs, typically after the stress evolution reaches a critical level defined as σ_{crit} (this moment is called the void nucleation time t_{nuc}), the normal component of stress at the void surface is constrained to zero due to the absence of confinement at the surface. However, the stress nearby cannot change immediately and will gradually decrease over time. The infinitesimally small distance between the zero stress and the remaining stress is referred to as the effective thickness of the void interface, denoted as δ_s. In [27], this value is set to $\delta_s = 1$nm. The tree is subsequently divided into several subtrees based on the void formation location. Typically, the voids will be situated at the terminal junctions of the decomposed trees [10], [11], [28], as it is supported by the observation that flux divergence is higher around junctions and especially around vias. Assuming a linear stress distribution across the void interface, the stress gradient follows $d\sigma_v^g(x_v,t)/dx = \pm\sigma_v^g(x_v,t)/\delta_s$ where g denotes the void growth phase and x_v is the location of the void surface. For simplicity, we rewrite the BC at the voided terminal as [15]:

$$\sigma_v^g(x_v, t) = 0, \tag{7}$$

where σ^g is the stress evolution during the void growth phase. The initial stress along the interconnect tree, excluding the void surface, remains the same as the stress evolution at t_{nuc}. Therefore, we have:

$$\sigma_i^g(x, 0) = \sigma_i^n(x, t_{nuc}). \tag{8}$$

2) Stress flows: For a straight multisegment interconnect, stress flow components are introduced in [14] to demonstrate the contribution of sources located at terminals and intersections of the interconnect trees to the stress at location x. These stress flows are the fundamental terms to build up the series in the stress solution, and the corresponding traveled distance, defined as $X_i^k(x)$, follows:

$$\begin{aligned}
X_i^0(x) &= \begin{cases} iL_N + x, & \text{i is even,} \\ (i+1)L_N - x, & \text{i is odd.} \end{cases} \\
X_i^N(x) &= \begin{cases} (i+1)L_N + x, & \text{i is even,} \\ iL_N - x, & \text{i is odd.} \end{cases} \\
X_i^k(x) &= \begin{cases} L_k + iL_N - x, & \text{i is even, to the left,} \\ L_k + (i-1)L_N + x, & \text{i is odd, to the left,} \\ -L_k + iL_N + x, & \text{i is even, to the right,} \\ -L_k + (i+1)L_N - x, & \text{i is odd, to the right.} \end{cases}
\end{aligned} \tag{9}$$

where k represents the distance from the k-th junction to x and satisfies $k = 1, \cdots, N-1$. The stress flow reflects at the boundaries and i represents the i-th reflection. Left and right flows are created for the source at interior junctions and travel sideways from the source. The 0-th reflection distance from the source of interior junctions called fundamental traveled distances, are combined from the left and right sides at $x = L_k$ into the distance $X_0^k(x) = |L_k - x|$. The stress flow decays exponentially with distance traveled and is obtained by [29]:

$$g(x,t) = 2\sqrt{\frac{\kappa t}{\pi}}e^{-\frac{x^2}{4\kappa t}} - x \times \texttt{erfc}\{\frac{x}{2\sqrt{\kappa t}}\}, \tag{10}$$

demonstrating the relationship between the stress flow, traveled distance, and time. The notation erfc represents the complementary error function. Finally, the stress flow under the i-th reflection is defined as $\pm G_k g(X_i^k(x), t)$ for $k = 0, N$, and $(G_{k+1} - G_k)g(X_i^{k,l/r}(x), t)/2$ for $0 < k < N$ where $X_i^{k,l}(x)$ and $X_i^{k,r}(x)$ are the left and right traveled distance in $X_i^k(x)$.

C. Spectral Bias Analysis through Neural Tangent Kernel and Random Fourier Feature Embedding

The recently developed neural tangent kernel (NTK) theory [30], [31] bridges the spectral bias analysis with the training behavior of fully-connected neural networks [32]. Given a dataset consisting of input $X_{train} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ and expected output $Y_{train} = [y_1, \cdots, y_n]$, and considering a network $f(\cdot)$ with unknown weights θ trained by minimizing the loss function MSE $= ||f(\theta; X_{train}) - Y_{train})||_2^2$, the entries of NTK, the dot-product kernel, are given by [30], [33]:

$$K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle\frac{df(\theta;\boldsymbol{x}_i)}{d\theta}, \frac{df(\theta;\boldsymbol{x}_j)}{d\theta}\rangle, \tag{11}$$

demonstrating the gradient descent dynamics under an infinitely small learning rate. The convergence rate of network training is shown to be determined by the eigenvalues of the kernel [31], [32]. It is suggested in [34] that a conventional fully-connected neural network is incapable of learning high-frequency tasks, where the eigenvalues of the kernel decrease sharply and monotonically with increasing frequency of the eigenvectors, a phenomenon termed "spectral bias".

To overcome this problem, [35] and [32] introduce Random Fourier Feature (RFF) embeddings into the neural network framework, defined as:

$$\gamma(\boldsymbol{B}, \boldsymbol{x}) = [\cos(\boldsymbol{B}\boldsymbol{x}), \sin(\boldsymbol{B}\boldsymbol{x})]^T, \tag{12}$$

where $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m]^T \in \mathbb{R}^{m\times d}$ is sampled from a Gaussian distribution $\mathcal{N}(0, \nu^2)$ and ν is a fixed hyper-parameter. The inputs $\boldsymbol{x}$ are transformed through the Fourier mapping $\gamma(\cdot)$, followed by a fully connected neural network. It has been demonstrated in [32] that the kernel of a two-layer linear and bias-free neural network with RFF embeddings follows $K_\gamma(x,x') = \cos(b(x-x'))$, featuring non-zero eigenvalues $\lambda = (1 \pm \sin b/b)/2$ by setting $\boldsymbol{B} = b \in \mathbb{R}$ when $d = m = 1$. The corresponding eigenfunction is $g(x) = C_1\cos(bx) + C_2\sin(bx)$, where C_1 and C_2 are constants. The finding illustrates that training behavior with larger ν would yield higher-frequency eigenfunctions and smaller eigenvalue gaps, potentially mitigating the spectral bias issue and accelerating convergence speed.

III. Novel Partitioning-based approach

Although previous work [21], [22] has successfully approximated stress evolution along interconnect trees using neural networks, the performance degrades as the number of segments increases, particularly in interconnect trees with offshoots. This degradation can be explained as the reduced approximation ability of neural networks in learning tasks penalized by the increasingly complex physics-constraint loss function. To address this issue, we propose a partitioning method to simplify the loss function and capture stress along larger-scale trees with improved accuracy and reduced computational time. In this method, an interconnect tree is first divided into several portions where the stress evolution on each portion can be directly solved. Subsequently, a learning-based method is employed to combine the stress on each portion, providing accurate stress solutions across the entire interconnect tree.

We start with a general interconnect tree with the goal of selecting certain junction nodes to be removed, which disconnects the tree, and refer to these nodes as "*joints*". This approach draws inspiration from node tearing technology [36]. For example, the yellow junctions in Fig. 1 can be selected as joints. Having removed the joints, we duplicate them and restore the edges of separated trees originally connected to the joints. These duplicated joints are called "*intersection junctions*". The multiple generated subtrees are referred to as "*bricks*" and specifically categorized as structures in Fig. 2. Specifically, we refer to a brick with degree-2 junctions as a straight multi-segment brick, a degree-3 junction as a T-shaped brick, and a degree-4 junction as a cross-shaped brick. A straight multi-segment brick can be transformed by folding the segments at the junctions. In our work, the terminal and interior junctions of straight multi-segment and T-shaped bricks, and the terminal junctions of the cross-shaped brick, as shown in the yellow squares from Fig. 2, can be treated as intersection junctions.

In the following, Section III-A presents the analytical solutions of stress evolution along different types of bricks as shown in Fig.2. Section III-B proposes modifications to these analytical solutions by considering the combination of stress on bricks, the transition between EM phases in each brick, and the impact of dynamic current density within each segment and time-dependent temperature on stress evolution. Finally, Section III-C concludes with the brick and joint solutions, and provides truncation strategies for fast calculation.

A. Analytical stress solutions for individual bricks

Our first step is to customize the analytical stress solutions for individual bricks. It should be noted that the stress solutions during the void nucleation phase are, in this section, under the assumption of zero initial stress. This will be extended to the general case in the next section. The schematic of each brick, introducing the notations L_k and j_i, and the positive current direction, is illustrated in Fig. 2. We denote the initial time in each EM phase by t_0, which is set to 0 during the void nucleation phase and t_{nuc} during the void growth phase. The notation $\mathcal{T} = t - t_0$ is then introduced to represent the evolution time, where t denotes the observation time instance.

1) Straight multi-segment brick: During the void nucleation phase, [14] demonstrates that substituting Eqs. (9) and (10) into:

$$\mathrm{g}^n_{i,k}(x,t) \triangleq \begin{cases} g(X^k_i(x),t), k=0,N, \\ \dfrac{g(X^{k,l}_i(x),t)+g(X^{k,r}_i(x),t)}{2}, k=1,\cdots,N-1, \end{cases} \tag{13}$$

where, as before, $X^{k,l}_i(x)$ and $X^{k,r}_i(x)$ are the left and right traveled distances of $X^k_i(x)$ from the k-th junction during the i-th reflection, the analytical stress evolution on a straight multi-segment brick follows:

$$\sigma^n_{2,j}(x,t) = \sum_{i=0}^{+\infty}\sum_{k=0}^{N} A_k \mathrm{g}^n_{i,k}(x,\mathcal{T}), L_{j-1} \le x \le L_j, \tag{14}$$

where the subscript $2,j$ represents the j-th segment of a straight N-segment brick (in which the maximum degree of a junction is '2'), while A_k equals G_1 for $k=0$, $-G_N$ for $k=N$ and $(G_{k+1}-G_k)$ for $k=1,\cdots,N-1$.

During the void growth phase, for $t > t_{nuc}$, since the analytical solution has already been published in [15], we summarize the solution below. Let $f_k(x) = \sigma^n_{2,k}(x,t_{nuc})$ where $x \in [L_{k-1}, L_k]$. The influence of initial stress along the k-th segment can be computed as [15]:

$$b_k(m,x,t) = \frac{1}{L_N} e^{-M_b} \int_{L_{k-1}}^{L_k} f_k(u)\sin(z_m(u-x))du, \tag{15}$$

where $z_m = (m-0.5)\pi/L_N$ is the m-th zeros obtained from the Residue Theorem and we have $M_b = z_m^2 \kappa_k t$. Consistent with a decay phenomenon, Eq. (15) is close to zero as time increases to infinity. The overall impact of initial stress on stress evolution along the j-th segment at $t > t_{nuc}$ follows:

$$\begin{aligned} \mathcal{B}^g_{2,j}(x,t) = \sum_{m=0}^{+\infty}(-1)^m \Big\{ &\sum_{k=1}^{N} b_k(m,x^-,\mathcal{T}) \\ &- \sum_{k=1}^{j-1} b_k(m,-x^-,\mathcal{T}) + \sum_{k=j}^{n} b_k(m,x^+,\mathcal{T}) \Big\}, \end{aligned} \tag{16}$$

where $x^+ = L_N + x$ and $x^- = L_N - x$. Then we establish

$$\mathrm{g}^g_{i,k}(x,t) \triangleq \begin{cases} (-1)^i g_0, k=0, \\ \dfrac{g_0 - g(\eta^{j-}_{N,k}(i,x),t) + g(\xi^{j+}_{N,k}(i,x),t)}{2\times(-1)^i}, 1 \le k < j, \\ \dfrac{g_0 + g(\eta^{j-}_{N,k}(i,x),t) - g(\xi^{j+}_{N,k}(i,x),t)}{2\times(-1)^i}, j \le k < N, \end{cases} \tag{17}$$

where $g_0 = -g(\xi^{j-}_{N,k}(i,x),t) + g(\eta^{j+}_{N,k}(i,x),t)$ and the detailed notations $\eta^{j-}_{N,k}$, $\eta^{j+}_{N,k}$, $\xi^{j-}_{N,k}$, $\xi^{j+}_{N,k}$ can be found in [15], representing the traveled distances. Stress evolution along the j-th segment of a straight interconnect tree satisfies [15]:

$$\hat{\sigma}^g_{2,j}(x,t) = \sigma^g_{2,j}(x,t) + \mathcal{B}^g_{2,j}(x,t) = \sum_{i=0}^{+\infty}\sum_{k=0}^{N} A_k \mathrm{g}^g_{i,k}(x,\mathcal{T}) + \mathcal{B}^g_{2,j}(x,t), \tag{18}$$

where $A_N = 0$ and the void formation is assumed at the right terminal ($x = L_N$).

2) T-shaped and cross-shaped bricks: In the following, the Laplace transform is used to derive the analytical stress solutions for bricks with a degree-M junction, where $M=3$ for the T-shaped brick and $M=4$ for the cross-shaped brick. A detailed derivation process for stress evolution along a T-shaped brick during the void nucleation phase is given in Appendix A. In Algorithm 1 of Appendix A, we provide a set of vectors including $\mathbf{L}^{i,M}$, $\boldsymbol{\mathcal{H}}^{C}$ and $\boldsymbol{\mathcal{H}}^{*}$, where i is a non-negative integer related to the number of stress flow reflections. We define $\boldsymbol{\mathcal{H}}^{C} \triangleq \boldsymbol{\mathcal{H}}(C)$ as a transformation that maps C, a set of indices of segments, to a vector used for calculating stress evolution. These vectors are then employed to obtain the traveled distance:

$$
\begin{aligned}
&\xi^{j+}_{p,q}(i,x) = \mathbf{L}^{i,M}_{p} - \boldsymbol{\mathcal{H}}^{\{j\}}_{q} - L_\Sigma - L_j + x,\\
&\xi^{j-}_{p,q}(i,x) = \mathbf{L}^{i,M}_{p} - \boldsymbol{\mathcal{H}}^{\{j\}}_{q} - L_\Sigma + L_j - x,\\
&\eta^{j+}_{p,q,k}(i,x) = \begin{cases} \mathbf{L}^{i,M}_{p} - \boldsymbol{\mathcal{H}}^{\{j,k\}}_{q} - L_\Sigma - L_j + x, \ j \neq k,\\ \mathbf{L}^{i,M}_{p} + \boldsymbol{\mathcal{H}}^{*}_{q} - L_\Sigma + x, \ j = k,\end{cases}\\
&\eta^{j-}_{p,q,k}(i,x) = \begin{cases} \mathbf{L}^{i,M}_{p} - \boldsymbol{\mathcal{H}}^{\{j,k\}}_{q} - L_\Sigma + L_j - x, \ j \neq k,\\ \mathbf{L}^{i,M}_{p} - \boldsymbol{\mathcal{H}}^{*}_{q} - L_\Sigma - x, \ j = k,\end{cases}
\end{aligned} \tag{19}
$$

where $x \in [L_0, L_j]$, $L_\Sigma = \sum_{j=1}^{M} L_j$, $j \in [1, M]$ and $k \in [0, M]$. The subscripts p and q are vector indices that traverse the entire vectors. At location x on the j-th segment of the brick, we employ Eq. (20) to calculate the stress flow component from the source of the terminal junction on the k-th segment:

$$
\mathrm{g}^{n/g}_{i,j,k}(x,t,M,\boldsymbol{n}) = \begin{cases} \dfrac{\sum_{p,q}\mathbf{K}^{i,M}_{p}\boldsymbol{h}^{\{j\}}_{q}\left(g(\xi^{j+}_{p,q}(i,x),t) + \boldsymbol{n}_1 g(\xi^{j-}_{p,q}(i,x),t)\right)}{M}, k=0,\\ \dfrac{\sum_{p,q}\mathbf{K}^{i,M}_{p}\boldsymbol{h}^{*}_{q}(\boldsymbol{n}_2 g(\eta^{j+}_{p,q,j}(i,x),t) + g(\eta^{j-}_{p,q,j}(i,x),t))}{M}, k=j,\\ \dfrac{2\sum_{p,q}\mathbf{K}^{i,M}_{p}\boldsymbol{h}^{\{j,k\}}_{q}\left(g(\eta^{j+}_{p,q,k}(i,x),t) + \boldsymbol{n}_3 g(\eta^{j-}_{p,q,k}(i,x),t)\right)}{M}, k\neq 0,j,\end{cases} \tag{20}
$$

where $k=0$ corresponds to the degree-M interior junction. The vectors $\mathbf{K}^{i,M}$, $\boldsymbol{h}^{*}$ and $\boldsymbol{h}^{C}$ can be calculated through Algorithm 1 of Appendix A where $\boldsymbol{h}^{C} \triangleq \boldsymbol{h}(C)$ represents a transformation from the set C to a vector, and $\boldsymbol{n} = [n_1, n_2, n_3]$ varies from different EM phases. The stress evolution on the j-th segment of the brick with a degree-M junction follows:

$$
\sigma^{n/g}_{M,j}(x,t) = \sum_{i=0}^{+\infty}\sum_{k=0}^{M} A_k \mathrm{g}^{n/g}_{i,j,k}(x,\mathcal{T},M,\boldsymbol{n}), L_0 \le x \le L_j, \tag{21}
$$

where $A_k = \sum_{i=1}^{M} G_i$ for $k=0$, and $A_k = -G_k$ for $k = 1,\cdots,M$, and $\boldsymbol{n} = [1,1,1]$ when estimating $\sigma^{n}_{M,j}(x,t)$ during the void nucleation phase (cf. Appendix A).

During the void growth phase, we assume the void is created at the terminal junction on the v-th segment. Let $f_k(x) = \sigma^{n}_{M,k}(x, t_{nuc})$. By utilizing the Residue Theorem, the initial unit function describing the influence of the initial stress along the k-th segment on the stress along the j-th segment yields:

$$
b_{j,k,M}(m,x,t) = \begin{cases} \dfrac{4e^{-M_b}\beta_j(z_m x)\prod_{\substack{i=1,\\ i\neq k,j}}^{M}\beta_i(z_m L_i)\int_{L_0}^{L_k} f_k(u)\beta_k(z_m(L_k-u))du}{Z}, j\neq k,\\ \dfrac{4e^{-M_b}\alpha_j(z_m x)\int_{L_0}^{L_k} f_k(u)\beta_k(z_m(L_k-u))du}{Z}, j=k,\end{cases} \tag{22}
$$

where $\alpha_k(\cdot)$ and $\beta_k(\cdot)$ are $\cos(\cdot)$ and $\sin(\cdot)$ for $k=v$, and $\sin(\cdot)$ and $\cos(\cdot)$ for $k \neq v$. The notation z_m is the m-th zeros of $p_M(z) = M\cos(L_\Sigma z) + (M-2)\left(\sum_{\substack{i=1,\\ i\neq v}}^{M}\cos((L_\Sigma - 2L_i)z) - \cos((L_\Sigma - 2L_v)z)\right)$. Due to the periodicity of $p_M(z)$, the number of zeros is infinite. We have $Z = -dp(z)/dz|_{z=z_m}$ and $M_b = z_m^2\kappa_k t$. The overall impact on stress evolution along the j-th segment takes the form:

$$
\begin{aligned}
\mathcal{B}^{g}_{M,j}(x,t) = \sum_{m=0}^{+\infty}\Big\{&\sum_{k=1}^{M}[n_k b_{j,j,M}(m,x_k,\mathcal{T}) + b_{j,j,M}(m,x^{+},\mathcal{T})]\\
&+2^{M-1}\sum_{\substack{k=1,\\ k\neq j}}^{M} b_{j,k,M}(m, L_j - x, \mathcal{T})\Big\},
\end{aligned} \tag{23}
$$

where $x^{+} = x + \sum_{\substack{i=1,\\ i\neq j}}^{M} L_i$. Notations x_k and n_k are $x - \sum_{\substack{i=1,\\ i\neq j}}^{M} L_i$ and $2-M$ for $k=v$, and $x^{+} - 2L_k$ and $M-2$ for $k \neq v$. Finally, during the void growth phase, the stress evolution along the j-th segment can be obtained:

$$
\hat{\sigma}^{g}_{M,j}(x,t) = \sigma^{g}_{M,j}(x,t) + \mathcal{B}^{g}_{M,j}(x,t), \tag{24}
$$

where in Eq. (21) $\boldsymbol{n} = [-1, 0, -1]$ for $j = v$ and $\boldsymbol{n} = [1, -1, 1]$ for $j \neq v$. It should be noted that A_v in Eq. (21) turns out to be zero. Furthermore, to solve for the stress evolution along interconnect trees with multiple voids, the interconnect tree can be partitioned into non-voided bricks and bricks containing a single void.

B. Modified analytical stress solutions for bricks

In this section, we first expand the analytical solutions to account for the combination of stress on different bricks, satisfying the BCs at intersection junctions. The stress solution is then modified to be constrained by the non-zero initial condition during the void nucleation phase for the stress assessment of each brick experiencing the transition between EM phases. Consequently, stress prediction under time-dependent current density within each segment and dynamic temperature is proposed.

By utilizing a linear transformation, EM-induced stress evolution, denoted by $\hat{\sigma}^{n/g}_{M,j}(x,t)$, can be converted to the sum of two stress components satisfying Korhonen's equation. For example, during the void growth phase, the first component, represented by $\sigma^{g}_{M,j}(x,t)$ in Eqs. (18) and (24), describes the stress evolution where the initial condition Eq. (8) is converted to $\sigma^{g}_{M,j}(x, t_{nuc}) = 0$, as the stress under zero initial condition. The second component, represented by $\mathcal{B}^{g}_{M,j}(x,t)$ in Eqs. (16) and (23), illustrates the stress evolution where G_i in Eq. (3) is set to be zero when $t > t_{nuc}$, characterizing the decay of initial stress.

According to Eqs. (14), (18), (21) and (24), we summarize the first stress component on individual N-segment bricks in which the maximum degree is M ($M = 2,3,4$):

$$
\sigma^{n/g}_{M,j}(x,t) = \sum_{i,k}\hat{A}_k \mathrm{g}^{n/g}_{i,k}(x, t-t_0), t > t_0, \tag{25}
$$

where $i \in [0,\infty)$ and $k \in [0,N]$ are integers, $\mathrm{g}^{n/g}_{i,k}(\cdot)$ follows Eqs. (13), (17) and (20) for different M. Let S_k be a set of indices of the segments connected to the k-th junction. The BC of Eq. (25) regarding atomic flux at junctions is governed by $\sum_{i\in S_k} \pm\partial\sigma^{n/g}_{M,i}(x,t)/\partial x|_{x=L_k} = \hat{A}_k$, where $\hat{A}_k$ represents the sum of EM driving forces flowing toward the k-th junction. The spatial derivative of stress at junctions, expressed as $\partial\sigma(x,t)/\partial x|_{x=L_k}$, is defined as the stress gradient, the sum of which can be computed by the flux balance condition Eq. (3). We refer to the stress gradients in the same brick as local

This article has been accepted for publication in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2025.3567885

stress gradients. For bricks without intersection junctions, $\hat{A}_k$ is consistent with A_k in Section III-A. As indicated in [15], when $\hat{A}_k$ is time-dependent, Eq. (25) turns to:

$$\sigma^{n/g}_{M,j}(x,t) = \sum_{i,k}\{\hat{A}_k(t_0)\mathrm{g}_{i,k}(x,t-t_0) + \hat{A}'_k(t) * \mathrm{g}_{i,k}(x,t-t_0)\}, \tag{26}$$

where $t > t_0$, and $\hat{A}'_k(t)$ represents the time derivative of $\hat{A}_k(t)$ and $*$ is the temporal convolution $a(t) * b(t-t_0) = \int_{t_0}^{t} a(\tau)b(t-t_0-\tau)d\tau$.

1) Combination of stress on bricks: Assuming the l-th junction is the intersection junction, the sum of local stress gradients toward the junction cannot be directly calculated due to the unknown stress and atomic flux from the neighboring brick. In this work, we regard $\hat{A}_l$ as a certain time-dependent variable that meets the physical constraints in Eqs. (3) and (4) at the intersection junction, considering the interaction between stress on adjacent bricks. Compared with our previous work [21] that proposed a method to combine stress on intersected single metal lines, we develop a novel approach to enable the combination of stress on versatile interconnect structures by introducing the well-tailored $\hat{A}_k(t)$:

$$\hat{A}_k(t) = \begin{cases} A_k, \ t=0, k \neq l, \\ m_c \dfrac{N_A W_n \kappa_n \ + A_k W \kappa}{m_n W_n \sqrt{\kappa_n \kappa} + m_c W \kappa}, \ t=0, k=l, \\ \hat{A}_k(0) + \displaystyle\int_0^t \hat{A}'_k(t)dt, \ t>0, \end{cases} \tag{27}$$

where A_k is provided in Section III-A, the second term is motivated by [21] and the third term presents a method to compute $\hat{A}_k(t)$ when t is positive. Notations m_c and m_n are the number of segments connected with the intersection junction in the current and neighboring bricks. The notations W, κ and W_n, κ_n are the width, diffusivity of the metal branch of the current and neighboring brick, respectively. N_A equals the sum of EM driving forces toward the intersection junction in the neighboring brick and for the current brick, the temporal derivative term $\hat{A}'_k(t)$ is zero when $k \neq l$ and the current density within each brick is constant. However, the derivative term $\hat{A}'_l(t)$ remains unknown when $t > 0$, which can be addressed using the powerful capabilities of neural networks. Since the equivalent EM driving force at $t > t_{nuc}$ in the second stress component, which represents the decay of initial stress, is zero, the combination of stress will not affect the value of $\mathcal{B}^{n/g}_{M,j}(x,t)$.

2) Transition between EM phases: Once an interconnect tree turns into the void growth phase, not all of its bricks undergo this phase simultaneously. Bricks that remain in the void nucleation phase will exhibit the stress evolution at t_{nuc} as the initial condition. Since the stress evolution can be decomposed into two components, the first stress component, with the initial condition $\sigma^g_{M,j}(x,t_{nuc}) = 0$ at the initial time $t_0 = t_{nuc}$ can be computed by Eq. (26). Furthermore, to estimate the second stress component $\mathcal{B}^n_{M,j}(x,t)$, the term $\hat{A}_k(t)$ in the void nucleation phase becomes $\hat{A}^-_k(t) = \hat{A}_k(t)(1-u(t-t_{nuc}))$ where $u(t)$ is the continuous-time unit step signal, used to set G_k to zero after t_{nuc}. In this EM phase, $\hat{A}^{-'}_l(t)$ is a determined function learned from the simulation in the void nucleation phase. Then the derivative term yields:

$$\frac{d\hat{A}^-_k(t)}{dt} = \hat{A}^{-'}_k(t) - \hat{A}_k(t_{nuc})\delta(t-t_{nuc}), t>0, \tag{28}$$

where $\hat{A}^{-'}_k(t) = \hat{A}'_k(t)(1-u(t-t_{nuc}))$. Substituting $t_0 = 0$ into Eq. (26), and replacing $\hat{A}_k(t_0)$ by $\hat{A}^-_k(t)$ applied at $t=0$ and $\hat{A}'_k(t)$ by the right-hand side of Eq. (28), the stress decay $\mathcal{B}^n_{M,j}(x,t)$ can be obtained. This result is then added to the first stress component, achieving stress evolution during the void nucleation phase considering the initial stress at t_{nuc}:

$$\begin{aligned}\hat{\sigma}^n_{M,j}(x,t) = \sum_{k,i}\{\hat{A}^-_k(0)\mathrm{g}^n_{i,k}(x,t) + \int_0^t \hat{A}^{-'}_k(\tau)\mathrm{g}^n_{i,k}(x,t-\tau)d\tau \\ + \int_{t_{nuc}}^t \hat{A}'_k(\tau)\mathrm{g}^n_{i,k}(x,t-t_{nuc}-\tau)d\tau\}, t>t_{nuc},\end{aligned} \tag{29}$$

where $\hat{A}'_l(t)$ is unknown for $t > t_{nuc}$. When the interconnect tree is in the void nucleation phase, stress solution $\hat{\sigma}^n_{M,j}(x,t)$ can be obtained by substituting $t_0 = 0$ into Eq. (26) where $\hat{A}'_l(t)$ remains unknown.

3) Stress solution under dynamic current density and temperature: When the current density within each segment is not constant, the dynamic EM driving force can be represented by $G_k(t)$ and used to calculate the time-dependent A_k. By substituting A_k into Eq. (27), the influence of dynamic current density can be considered where $\hat{A}'_k(t)$ $(k \neq l)$ becomes non-zero. The impact of time-dependent temperature on stress can be addressed using the method in [15].

C. Bricks and joints building

1) Brick solution: We first employ $\hat{A}^{+'}_l(t)$ to represent the undetermined temporal derivative of stress gradients in Eqs. (26) and (29). Specifically, we set $\hat{A}^{+'}_l(t) = \hat{A}'_l(t)$ for $t \leq t_{nuc}$ and $\hat{A}^{+'}_l(t) = \hat{A}'_l(t)u(t-t_{nuc})$ for $t > t_{nuc}$. For bricks in the void nucleation phase, omitting $\hat{A}^{+'}_l$ in (26) and (29) results in an approximation as the brick solution:

$$\sigma^n_{M,j}(x,t) = \begin{cases} \displaystyle\sum_{i=0}^{+\infty}\{\sum_{k=0}^{N}[\hat{A}_k(0)\mathrm{g}^n_{i,k}(x,t)] + \sum_{\substack{k=0,\\k\neq l}}^{N}[\int_0^t \hat{A}'_k(\tau)\mathrm{g}^n_{i,k}(x,t-\tau)d\tau]\}, t \leq t_{nuc}, \\ \displaystyle\sum_{i=0}^{+\infty}\{\sum_{k=0}^{N}[\hat{A}^-_k(0)\mathrm{g}^n_{i,k}(x,t) + \int_0^t \hat{A}^{-'}_k(\tau)\mathrm{g}^n_{i,k}(x,t-\tau)d\tau] \\ \qquad + \displaystyle\sum_{\substack{k=0,\\k\neq l}}^{N}[\int_{t_{nuc}}^t \hat{A}'_k(\tau)\mathrm{g}^n_{i,k}(x,t-t_{nuc}-\tau)d\tau]\}, t > t_{nuc}, \end{cases} \tag{30}$$

where N is the number of segments in the brick.

For the void growth phase, we substitute $t_0 = t_{nuc}$ into Eq. (26) to compute $\sigma^g_{M,j}(x,t)$ under dynamic current density within each segment. The result is then substituted into Eqs. (18) and (24). By dropping the terms with $\hat{A}^{+'}_l(t)$, an approximation of stress evolution yields:

$$\begin{aligned}\sigma^g_{M,j}(x,t) = \mathcal{B}^g_{M,j}(x,t) + \sum_{i=0}^{+\infty}\{\sum_{k=0}^{N}[\hat{A}_k(t_{nuc})\mathrm{g}^g_{i,k}(x,t-t_{nuc})] \\ + \sum_{\substack{k=0,\\k\neq l}}^{N}\int_{t_{nuc}}^t \hat{A}'_k(\tau)\mathrm{g}^g_{i,k}(x,t-t_{nuc}-\tau)d\tau\}.\end{aligned} \tag{31}$$

The brick solutions in Eqs. (30) and (31) consist of terms that can be precomputed before neural network training and treated as constants, thereby reducing the computational overhead of gradient computation during back-propagation. The terms with $\hat{A}^{+'}_l(t)$ are used to construct the joint solutions.

Given the properties of the bricks, such as the current density, topology of the interconnects, and the prior knowledge $\hat{A}^{-'}_l(t)$ learned in previous simulation steps, Eqs. (30)

This article has been accepted for publication in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2025.3567885

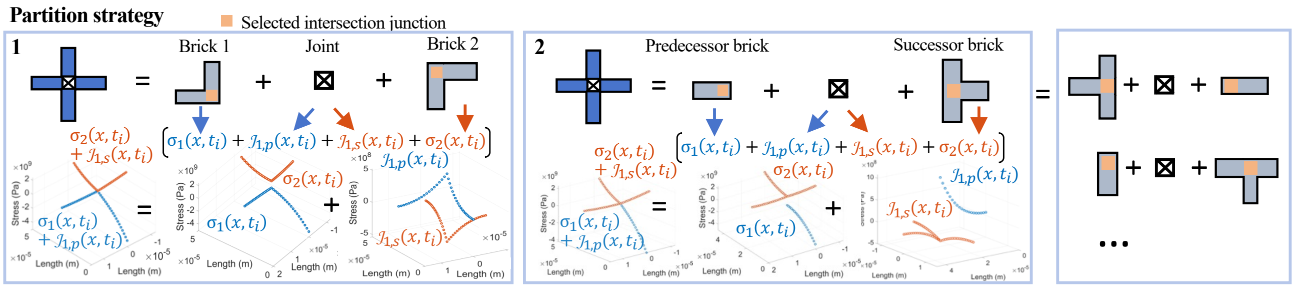


Fig. 3. Partitioning strategy in EM assessment.

and (31) can be easily calculated. This computation can be parallelized, as the solution on distinct bricks and segments can be independently solved.

2) Joint solution: Fig. 3 presents several partitioning strategies and two detailed scenarios for calculating stress evolution along a cross-shaped interconnect tree. In these scenarios, the partitioning-based approach divides the interconnect tree into two bricks. Specifically, one configuration consists of two folded straight multi-segment bricks, while the other comprises a straight single-segment brick and a T-shaped brick. The resulting bricks are referred to as the predecessor and successor bricks, with their respective stress distributions subsequently integrated. In the original interconnect tree, the bricks are connected at a joint, which is duplicated to produce the intersection junctions on the adjacent bricks, thereby exhibiting impacts on the stress of both bricks. This influence is controlled by $\hat{A}_l(t)$ and $\hat{A}_l'(t)$, and we further denote the undetermined term $\hat{A}_l^{+'}$(t) as $\hat{A}_{j,p}^{+'}(t)$ and $\hat{A}_{j,s}^{+'}(t)$ corresponding to the intersection junctions duplicated from the j-th joint, on the predecessor and successor bricks, respectively. Subsequently, we introduce a compact Multilayer Perception (MLP) with RFF embedding $\gamma(\cdot)$ in Eq. (12), expressed as $\boldsymbol{y} = f(\gamma(\boldsymbol{B}, \boldsymbol{x}); \theta)$, where $\boldsymbol{x}$ and θ are the inputs and trainable weights in the neural network $f(\cdot)$. The output $\boldsymbol{y}$ varies from the different locations of joints and time instances, and thus can be rewritten as $y_{j,t}$ to illustrate the approximation of $\hat{A}_{j,p/s}^{+'}(t)$ for the j-th joint at time t. Based on Eq. (3), we have:

$$\hat{A}_{j,p/s}^{+'}(t) = \vec{n}_{j,p/s} \cdot y_{j,t}, \tag{32}$$

where $\vec{n}_{j,p} = -1/(W\kappa)$ and $\vec{n}_{j,s} = 1/(W\kappa)$. The impact of the j-th joint on stress along each brick during the void nucleation and growth phases, defined as "*joint solution*", is constructed based on Eq. (26):

$$\mathcal{J}_{j,p/s}(x,t) = \sum_{i=0}^{+\infty} \int_{t_0}^{t} \hat{A}_{j,p/s}^{+'}(\tau) \mathrm{g}_{i,j,p/s}(x, t - t_0 - \tau) d\tau, \tag{33}$$

where t_0 is set to 0 for interconnect trees in the void nucleation phase and t_{nuc} in the void growth phase. Eq. (33) illustrates the locality property of EM-induced stress, where the solution reaches its maximum magnitude at the joint and rapidly decreases along both sides. This phenomenon will be discussed in Section V-A2. As demonstrated in Fig. 3, during the void nucleation phase and along the whole interconnect tree, the stress evolution of the original tree at $t = t_i$ follows $\hat{\sigma}(x,t_i) = \sigma_1(x,t_i) + \mathcal{J}_{1,p}(x,t_i) + \mathcal{J}_{1,s}(x,t_i) + \sigma_2(x,t_i)$ where $\sigma_1(x,t)$ and $\sigma_2(x,t)$ are solutions of brick 1 and brick 2. The spatial coordinates x are first converted to relative coordinates within the bricks, as the inputs for brick and joint solutions. The solutions are then restored to their locations in the original interconnect tree, as illustrated by the stress curves plotted in Fig. 3. Finally, for a general interconnect tree broken into N_b bricks by N_j joints, the stress evolution follows:

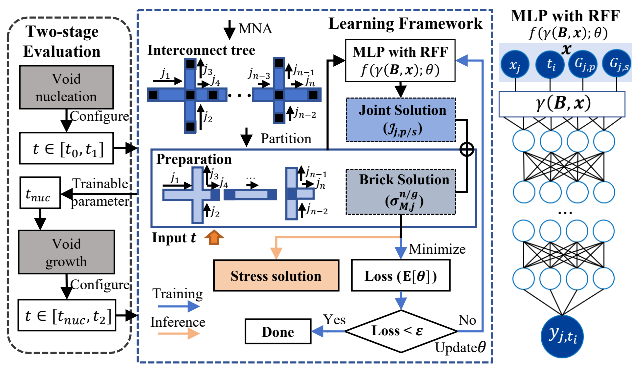


Fig. 4. Framework of the proposed method.

$$\hat{\sigma}(x,t) = \sum_{i=1}^{N_b} \sigma_i(x,t) + \sum_{j=1}^{N_j} \left(\mathcal{J}_{j,p}(x,t) + \mathcal{J}_{j,s}(x,t)\right), \tag{34}$$

where $\sigma_i(x,t)$ is the i-th brick solution.

3) Compact solution: To facilitate EM-induced stress calculation, a truncation strategy is required since the stress evolution solutions comprise an infinite number of terms. According to the decay property of $g(x,t)$, the value diminishes quickly when the traveled distance increases. As a result, we establish a distance threshold $x_{\text{crit}} = 4\sqrt{\kappa t}$, beyond which the value of $g(x,t)$ can be disregarded. Additionally, the convergence of $\mathcal{B}_{M,j}^{g}(x,t)$ has been proven in [15] and the series $b_k(\cdot)$ and $b_{j,k,M}(\cdot)$ can be truncated following the criteria configured as $M_b = 3$. For the temporal convolution operation in Eqs. (30), (31) and (33), we employ an eight-point Gauss–Legendre quadrature technology, enhancing computational efficiency while keeping promising accuracy.

IV. Proposed Learning Framework

The general framework of the proposed method is shown in Fig. 4 consisting of the learning framework and a two-stage evaluation.

A. Learning framework

1) Preparation: Given an interconnect tree with applied current sources at varying junctions, we first calculate the node voltages and current within each segment using modified nodal analysis (MNA). Based on the brick types outlined in Fig. 2, we partition the interconnect tree into individual bricks and compute the current density within each segment based on the results obtained through MNA.

The neural network input is structured as a collection $\boldsymbol{x} = \{x_j, G_{j,p}, G_{j,s}, t_i\}$ and the corresponding output y_{j,t_i} is employed to approximate $\hat{A}^{+'}_{j,p/s}$ in Eq. (32). Here, we define x_j as the location of the j-th joint in the original interconnect tree, and $G_{j,p}$ and $G_{j,s}$ as the average incoming EM driving forces toward the j-th joint on the predecessor and successor bricks, respectively. These forces can be calculated by $\sum G/m_c$, where m_c and $\sum G$ are the number of segments connected to the junction and the sum of EM driving forces on adjacent segments toward the junction. The temporal instances t_i are randomly sampled within the appropriate range using Latin hypercube sampling and logarithmic equidistant sampling strategies as described in [23], with the number of sampling points defined as N_t. The location, EM driving force, time instance, diffusion coefficient and stress are normalized to regular magnitudes following [15], where $x = 10^5 x_o,\ t = 10^{-8} t_o,\ \kappa = 10^{18}\kappa_o,\ G = 10^{-14} G_o$ and $\sigma = 10^{-9}\sigma_o$. The network input is subsequently normalized to the range $[-1, 1]$. Furthermore, before training, the brick solutions in Eqs. (30) and (31) can be precomputed for each location and time, and treated as constants to alleviate the computational burden during the training process.

2) Training and inference: During the training process, following the preparation step, the RFF embedding, as defined in Eq. (12), is applied to the inputs before passing them into a compact MLP, as illustrated on the right side of Fig. 4. The matrix $\boldsymbol{B}$ is randomly sampled from a Gaussian distribution and remains fixed throughout the training process. The output y_{j,t_i}, corresponding to different joint locations and N_t time instances, can be regarded as a stress source at the j-th joint. The parameters in the MLP with RFF framework are shared when predicting different y_{j,t_i}, enabling efficient mapping from the various joint locations (x_j), time instances (t_i), and EM driving forces ($G_{j,p}, G_{j,s}$) to the corresponding y_{j,t_i}. This output is then propagated outward from the joint and decays over distance. We quantify this propagation as the joint solution, where the resulting $\hat{A}^{+'}_{j,p/s}$ is continuously adjusted by the neural network to build the stress evolution along each brick respecting Korhonen's physical constraints. Substituting the brick solutions and joint solutions into Eq. (34), the loss function can be established:

$$E[\theta] = \frac{1}{N_t N_j} \sum_{i=1}^{N_t} \sum_{j=1}^{N_j} |\hat{\sigma}_{j,p}(x_j, t_i) - \hat{\sigma}_{j,s}(x_j, t_i)|^2, \qquad (35)$$

where $\hat{\sigma}_{j,p}(x,t)$ and $\hat{\sigma}_{j,s}(x,t)$ are the stress solutions in the predecessor and successor bricks connected with the j-th joint located at x_j, respectively. Equation (35) is minimized to penalize the deviations of Eq. (4) to approximate joint solutions in Eq. (33). Thanks to the well-tailored analytical solutions in Eqs. (30), (31) and (33) that meet constraints in Eqs. (2)-(3) and Eqs. (5)-(8), the optimization goal of the neural network is simplified to the stress continuity condition at certain junctions, eliminating the need for derivative operations and resulting in significant computational savings compared with traditional PINNs. Furthermore, compared with the methods proposed in [21], [22] which divide N-segment interconnect trees into N partitions, our approach notably reduces the number of neural network outputs, especially at intersection junctions with degrees greater than two, and the number of terms in Eq. (35). This reduction is achieved by utilizing a diverse set of brick structures as partitioning units, therefore leveraging a more compact representation of the interconnect structure.

After convergence, the proposed model can achieve the accurate approximations of $\hat{A}^{+'}_{j,p/s}$. The location x_i within the interconnect tree and the observation time instance t_i in the training range are then substituted into Eq. (34) to obtain the stress solution. Finally, the magnitude scaling is restored to achieve the unnormalized stress evolution.

B. Two-stage EM evaluation

The proposed two-stage stress evolution in Fig. 4 is divided by the switching of the EM phase. Initially, all the bricks partitioned from the interconnect tree are in the void nucleation phase and the temporal inputs of the learning framework are obtained through random sampling within the time range $[t_0, t_1]$. After convergence of the learning framework, if the stress evolution exceeds the critical level at t_1, the void nucleation time $t_{nuc} \in [t_0, t_1]$ becomes a trainable parameter. The objective is to adjust this parameter so that the maximum stress at the junctions equals the critical level. In this evaluation, voids are assumed to occur at junctions, as no pre-existing voids and internal voids are considered in the stress modeling [10]. Once the interconnect tree enters the void growth phase, the second stage configures the training time range as $[t_{nuc}, t_2]$. To compute the brick solutions Eqs. (30) and (31) in this stage, we load the previous learning model to determine the coefficient $\hat{A}^{-'}_{l}$ through outputs at different locations and specific time instances, and $\mathcal{B}^{g}_{M,j}$ through the approximated stress evolution at t_{nuc}. Following a training process similar to that of the first stage, the learning framework can provide the post-voiding stress evolution during the inference phase.

V. Experimental Results

In this section, we present the stress evolution results obtained using the proposed approach during the void nucleation and void growth phases. Firstly, we validate the accuracy of the brick solution across various interconnect shapes, and provide a detailed procedure for partitioning an interconnect tree and combining stress solutions along the subcomponents. The stress prediction on interconnect trees is extended to consider varying physical parameters and operating conditions. We then evaluate the training acceleration achieved by incorporating RFF embeddings into a fully connected neural network, compared with utilizing a single MLP as described in [21], [22]. The neural network structure is customized through a detailed accuracy analysis under different numbers of layers and neurons per layer. Finally, we demonstrate the scalability of the proposed approach and the performance enhancements over existing methods. The proposed model and the stress model in [21] are carried out in Python 3.9.5 using PyTorch 1.12 and Tensorflow 1.14, respectively. The competing semi-analytical method in [37] is implemented in MATLAB. The experiments are conducted on a Linux machine equipped with an Intel Core i5-10505 CPU with 16GB of RAM, and an Nvidia RTX 3090 GPU with a 24GB buffer. The reference solution is obtained from FEM simulations conducted by the COMSOL Multiphysics software [38].

In the experiments, the RFF embedding ($m = 30$, $\nu = 3$) is followed by a fully connected MLP with three 60-neuron hidden layers. We utilize the Adam optimizer for neural network training and implement an early stopping mechanism when the loss drops below 5×10^{-6}. Xavier initialization is used to initialize the parameters of the MLP. The activation function is $\tanh$, and the training batch size is configured to $N_t = 1,000$.

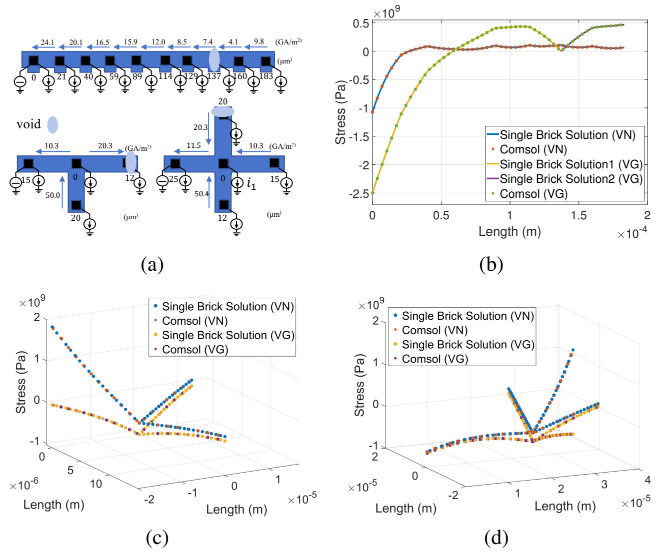


Fig. 5. a) Configuration of segment lengths and current densities on different single bricks, and the comparison of stress evolution on a b) straight 9-segment line, c) T-shaped, d) cross-shaped brick at $t = 10^8$s in the void nucleation (VN) and void growth (VG) phase.

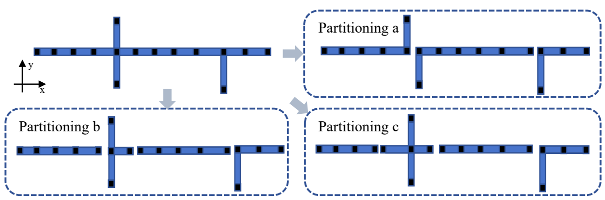


Fig. 6. Three partitioning methods for the 14-segment interconnect tree.

The values of parameters in the physical model are $k = 1.3806 \times 10^{-23} J/K,\ e = 1.609 \times 10^{-19} C,\ Z^* = 10,\ E_a = 1.1eV,\ B = 1 \times 10^{11} Pa,\ D_0 = 7.5 \times 10^{-5} m^2/s,\ \rho = 2.2 \times 10^{-8} \Omega \cdot m,\ \Omega = 8.78 \times 10^{-30} m^3,\ T = 350K,\ \sigma_{crit} = 5 \times 10^8 Pa$. Interconnect trees with the number of offshoots exceeding 10% of the total number of segments are tested in the experiments.

A. Accuracy Analysis

1) Single brick solution: Considering a single brick as a complete interconnect tree, we calculate the current distribution based on the configured current/voltage sources. Fig. 5(a) provides the length and current density configuration in the straight 9-segment line, T-shaped, and cross-shaped interconnect trees. Along interconnect trees with varying structures, the comparison of stress solutions obtained using brick solutions and COMSOL simulations are shown in Figs. 5(b)-5(d), showing agreement between the stress curves. The stress evaluation encompasses both the void nucleation (VN) and void growth (VG) phases, presenting the stress evolution at 10^8s from the beginning of each EM phase. With a computation time of less than 0.013s, the relative errors of stress along a straight nine-segment, T-shaped and cross-shaped interconnect trees are 0.03%, 0.34%, 0.89% during the void nucleation phase, and 0.21%, 0.19%, 0.88% during the void growth phase. The stress evolution along the straight nine-segment interconnect tree is divided into two single brick solutions by the void located at the interior junction during the void growth phase.

2) Brick-joint solution: Given a 14-segment interconnect tree with three offshoots, Fig. 6 lists three strategies for

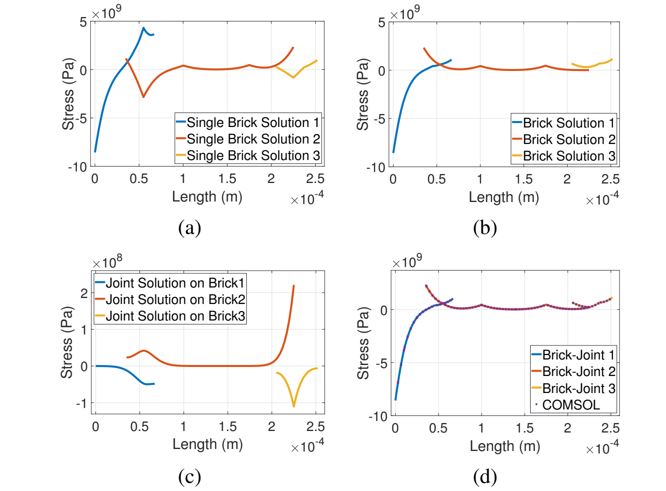


Fig. 7. Stress evolution on a 14-segment interconnect tree at $t = 10^8$s under partitioning method a in Fig. 6: a) single brick solutions, b) brick solutions, c) joint solutions, d) the summation of b) and c) yields the overall brick-joint solution. To enhance the visualization, each 2-D spatial coordinate (x_i, y_i) located in the interconnect tree is transformed into a 1-D coordinate $x_i + y_i$.

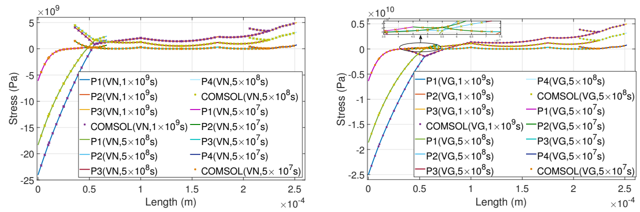
partitioning the interconnect tree into a) three straight multi-segment bricks, b) three straight multi-segment bricks and one T-shaped brick, c) three straight multi-segment and one cross-shaped brick. This partitioning is achieved by dividing the interconnect tree into bricks with specific structures.

As illustrated in the first partitioning method, we utilize the proposed brick-joint method to compute the stress evolution at $t = 10^8$s during the void nucleation phase. To enhance the visualization and facilitate a more straightforward comparison of stress curves derived from varying approaches, we convert the 2-D coordinates of the interconnect trees into a 1-D representation, which has been used in [39]. Specifically, each spatial coordinate (x_i, y_i) is transformed into a single coordinate $x_i + y_i$. Initially, we present the single brick solutions in Fig. 7(a), demonstrating that although Korhonen's diffusion kinetics are observed in the individual stress curves, no constraints are enforced at the intersection junctions. Moreover, to modify the stress solution, $\hat{A}_k(t)$ at intersection junctions is fine-tuned according to Eq. (27), resulting in brick solutions as depicted in Fig. 7(b), where a reduction in bias between stress solutions at intersection junctions across different connecting bricks can be observed. The stress solutions in Fig. 7(b) show compatibility with the flux balance constraint in Eq. (3), yet further enhancements are required to fully satisfy the stress continuity condition in Eq. (4). Compared with the single brick solution, the complete brick solution shows progress toward approximating the desired stress curves. Neural network training on joint solutions is then conducted, as plotted in Fig. 7(c), illustrating the influence of the combination operation on stress evolution near the joints. This influence reaches maximum magnitude at intersection junctions, and decreases along both sides, thereby demonstrating the locality property of stress evolution in the partition strategy. Finally, the prediction obtained by the brick-joint solution is shown in Fig. 7(d), accompanied by the overlapping stress solution calculated from COMSOL.

The performance of the proposed method is reported in Table I where $t_{pre},\ t_{tra}$ are the preparation time for dataset generation and brick solution calculation, and the training time, respectively. The notations $t_{tot},\ t_{inf}$ correspond to the total time required to achieve a prediction model satisfying $t_{tot} = t_{pre} + t_{tra}$ and the inference time for stress assessment. Errors

TABLE I
PERFORMANCE OF THE PROPOSED METHOD UTILIZING PARTITIONING METHODS IN FIG. 6 DURING THE VOID NUCLEATION AND VOID GROWTH PHASES.

Method	**Void Nucleation**						**Void Growth**					
	Training			*Inference*	*Error*		*Training*			*Inference*	*Error*	
	t_{pre} (s)	t_{tra} (s)	t_{tot} (s)	t_{inf} (s)	δ_{rel} (%)	δ_{max} (%)	t_{pre} (s)	t_{tra} (s)	t_{tot} (s)	t_{inf} (s)	δ_{rel} (%)	δ_{max} (%)
a	0.61	0.21	0.82	0.009	0.17	0.16	0.18	0.28	0.46	0.016	0.54	0.22
b	0.63	0.41	1.04	0.012	0.18	0.15	0.49	0.44	0.93	0.026	0.34	0.17
c	0.65	0.66	1.31	0.012	0.18	0.12	0.56	0.96	1.52	0.028	0.32	0.16



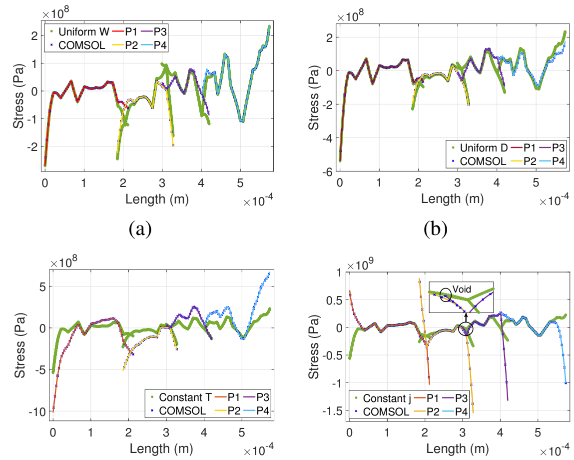
(a) (b)

Fig. 8. Comparison of stress evolution on a 14-segment interconnect tree during a) void nucleation phase, b) void growth phase where we employ three different partitioning methods shown in Fig. 6. Pi represents the proposed brick-joint solution on the i-th brick.

δ_{rel}, δ_{max} are the relative errors between the approximations and results from COMSOL, and the maximum relative errors calculated as $\max|\sigma_{\text{Proposed}} - \sigma_{\text{COMSOL}}|/\max|\sigma_{\text{COMSOL}}|$. The errors are computed based on the stress evolution along the interconnect tree at 10 equidistantly sampled time points within the time range [10^6s, 10^8s]. Three different partitioning methods shown in Fig. 6 are conducted for the performance evaluation, indicating that higher accuracy can be attained by utilizing brick types with a higher junction degree, albeit with an increase in computation time. The stress prediction at 5×10^7s, 5×10^8s, and 10^9s are plotted in Fig. 8, which are consistent with the results from COMSOL.

3) Physical parameters and operating conditions: In this section, we configure a 40-segment interconnect tree to analyze the impact of varying physical parameters, such as segment width (W) and effective atomic diffusivity (D), as well as operating conditions, including dynamic temperature (T) and current density (j), on EM-induced stress evolution. Figs. 9(a)-9(b) depict stress evolution curves at $t = 10^8$s, obtained after assigning different values of W and D to specific partitions. The widths of the partitions are set to $0.2\mu m$, $0.1\mu m$, $0.15\mu m$, and $0.1\mu m$, with corresponding effective atomic diffusivity of $9.24\times10^{-21}m^2/s$, $7.43\times10^{-21}m^2/s$, $9.28\times10^{-21}m^2/s$, and $9.90\times10^{-21}m^2/s$. The relative errors of stress results within $t = 10^8$s, as obtained by the proposed approach, are 0.17% and 0.14%, respectively. Additionally, the stress evolution under constant width ($W = 0.1\mu m$) and constant effective atomic diffusivity ($D = 9.24\times10^{-21}m^2/s$) is included in the figures to highlight the significance of incorporating variability in interconnect width and diffusivity driven by stochastic effects in EM assessments. In practical applications, a Monte Carlo (MC) simulation loop is employed to generate random effective atomic diffusivity, introducing the key stochastic parameter κ. A subsequent MC simulation is performed to model the randomness of critical stress levels, which govern transitions between distinct EM phases.

By setting the dynamic temperature as $T = 350+30\sin(2\times10^{-8}\pi t)$ K, the stress evolution at $t = 10^8$s during the void nucleation phase is plotted in Fig. 9(c), with a relative error of 0.21%. Moreover, the figure includes stress evolution under a constant temperature of 350 K, highlighting the acceleration and deceleration effects of temperature fluctuations on stress



(a) (b) (c) (d)

Fig. 9. Comparison of stress evolution on a 40-segment interconnect tree at $t = 10^8$s considering a) varying segment width, b) different effective atomic diffusivity D and c) dynamic temperature during the void nucleation phase, and d) dynamic current density during the void growth phase.

build-up. Subsequently, a linear current density is configured for each segment during the void growth phase and the EM driving force is set as $G(t) = G_0(1 + 5\times10^5 t)A/m^2$, where G_0 represents the initial EM driving force varying from segments. The stress comparison shown in Fig. 9(d) illustrates the capability of the proposed approach in addressing EM-induced stress considering dynamic current conditions, with a relative error of 0.22%. Stress evolution at 10^8s under constant j is plotted in Fig. 9(d). This highlights the potential for the partitioning-based method to assess EM failure in the void growth phase, considering dynamic current densities due to global effects such as the dynamic IR drop in EM degradation.

B. Performance Analysis

MLP architectures are commonly used in tasks associated with solving EM-induced stress evolution due to their simplicity and lightweight nature [21], [22]. These approaches draw inspiration from PINN, which initially employed MLPs as network structures. In this study, an RFF embedding is introduced before passing inputs into a conventional fully connected MLP, resulting in a notable improvement in training speed. Fig. 10 shows the training performance on a 300-segment interconnect tree, reporting the loss and relative error over iterations for a conventional MLP and an MLP with RFF embeddings ($\nu = 1 \sim 7$). Both models are fully connected networks with three hidden layers of 60 neurons each. In the conventional MLP, the weights from the input layer to the first hidden layer are undefined, while in the MLP with RFF embeddings, the mapping follows Eq. (12). As shown in Fig. 10, incorporating RFF embeddings (dotted lines) accelerates the decline in loss and relative error during training compared with the conventional MLP (solid lines).

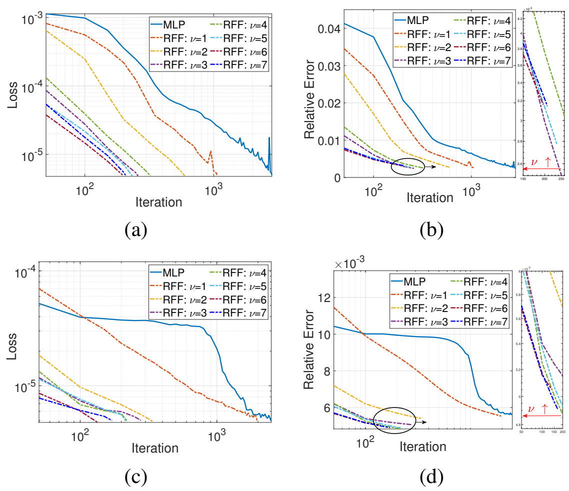


Fig. 10. Loss and relative error versus iterations by employing a conventional MLP [21], [22] and an MLP with RFF embeddings under different ν during a) and b) void nucleation phase, and c) and d) void growth phase. Training is early stopped when the loss reaches 5×10^{-6}.

The RFF embedding serves as a simple nonlinear transformation for the input, leading to negligible additional training burden. As illustrated in Eq. (12), the preferred learning frequency of an RFF-enhanced MLP is controlled by the hyperparameter ν, demonstrated through seven ablation experiments under $\nu = 1 \sim 7$. Fig. 10(a) shows that increasing ν reduces the training iterations required for the loss to reach 5×10^{-6} during the void nucleation phase. While a general improvement in convergence speed is noted as ν increases from 3 to 7, this gain comes at the expense of accuracy in the predicted stress solutions. One potential explanation for the reduction in accuracy is that the neural network tends to interpolate the training data with higher frequency oscillations than those present in the target solution [32]. This trend is also noted during the void growth phase. Across both EM phases, the model with $\nu = 6$ demonstrates the fastest convergence to the loss threshold. However, achieving the same loss level, the models with $\nu = 3$ and $\nu = 4$ exhibit the highest accuracy in stress predictions during the void nucleation and the void growth phases, respectively. In subsequent experiments, $\nu = 3$ is selected for the RFF embedding. To further accelerate convergence as the number of partitions increases, ν can be adjusted to higher values but should be bounded to avoid frequency mismatches between the network and the target solution. In this study, the upper bound for ν is set to 7.

Moreover, we investigate the influence of network structure on training efficiency. Specifically, we set the length of RFF embedding output, defined as $2m$ in Eq. (12), equal to the number of neurons per layer in the cascaded MLP. Table II provides the number of training iterations and the corresponding training time required for achieving a loss of 5×10^{-6} under varying widths and depths of the MLP, and different lengths of the RFF embedding output in the learning framework. As the neural network structure expands, the number of iterations decreases, with a rising training overhead per iteration. This trend remains consistent across various lengths of RFF embedding outputs. The neural network structure should be tailored, considering the trade-off between the number of iterations and the training time per iteration. In the following experiments, with a sufficient number of neurons in the model, we utilize an RFF embedding configured with $m = 30$, connected to a 3-hidden-layer MLP with 60 neurons per layer.

TABLE II
NUMBER OF TRAINING ITERATIONS/TRAINING TIME (S) TO ACHIEVE THE LOSS OF 5×10^{-6} UNDER DIFFERENT NUMBERS OF HIDDEN LAYERS AND NEURONS PER LAYER IN THE LEARNING FRAMEWORK DURING THE VOID NUCLEATION PHASE. THE LENGTH OF THE RFF EMBEDDING OUTPUT EQUALS THE NUMBER OF NEURONS PER LAYER.

Layers \ Neurons	20	40	60	80
1	2765/7.20	1057/3.54	733/3.03	488/3.94
2	1715/5.21	660/2.78	441/2.42	278/2.86
3	1228/4.34	465/2.35	352/2.73	241/3.02
4	618/5.74	381/4.07	253/3.26	206/3.07

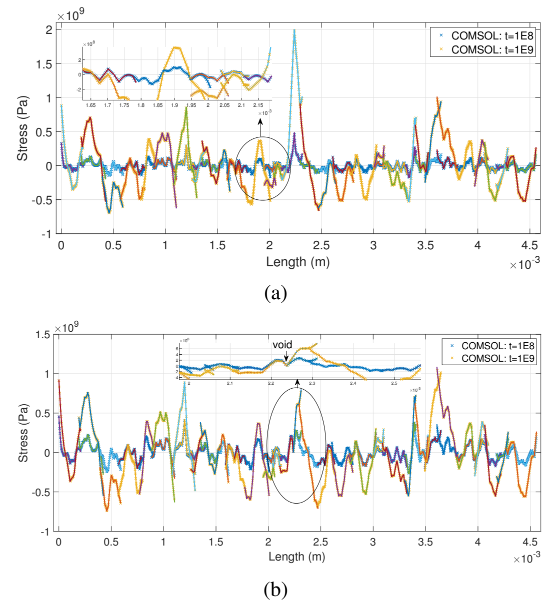


Fig. 11. Stress evolution along a 300-segment interconnect tree at 10^8s and 10^9s during a) the void nucleation phase, b) the void growth phase. Stress solutions obtained from different bricks are represented in various colors.

C. Scalability Analysis

In this section, we evaluate the scalability of the proposed method on interconnect trees with increasing numbers of segments. Fig. 11 illustrates the stress evolution along a 300-segment interconnect tree at $t = 10^8$s and $t = 10^9$s obtained from the proposed model and COMSOL. The brick-joint solutions of different bricks are depicted in various colors. The stress curves at the same time instances are indistinguishably overlapped, indicating promising stress predictions in the proposed model. Additionally, the position of void generation is shown in Fig. 11(b).

Table III presents the performance comparison between the proposed approach and COMSOL for interconnect trees with junctions of degree 2-4. In COMSOL simulations, the Finer mesh setting is utilized, with memory consumption ranging from 1.2 to 4.0 GB of RAM, and the runtime required to calculate stress solutions in each EM phase is defined as t_{com}. With early stopping, the proposed method takes 33.04s to train a model approximating stress evolution along a 1000-segment interconnect tree, with relative errors of less than 0.94%, where the scale of interconnect trees that can be solved exceeds those mentioned in [21], [22]. The relative errors of stress solutions within 10^8s are below 1.08% in all test cases and the total training overhead illustrates speedups ranging from 35× to 132× over COMSOL. The memory usage of the proposed approach is $O(n)$, and the training time grows as $O(\beta N_j)$, where β and N_j denote the number of training iterations

TABLE III
SCALABILITY PERFORMANCE COMPARISON BETWEEN COMSOL AND THE PROPOSED METHOD ON INCREASING N-SEGMENTED INTERCONNECT TREES DURING THE VOID NUCLEATION AND THE VOID GROWTH PHASE.

n-segment	**COMSOL**	**Void Nucleation**						**Void Growth**					
	t_{com} (s)	*Training*			Speedup vs COMSOL	*Inference*	*Error*	*Training*			Speedup vs COMSOL	*Inference*	*Error*
		t_{pre} (s)	t_{tra} (s)	t_{tot} (s)		t_{inf} (s)	δ (%)	t_{pre} (s)	t_{tra} (s)	t_{tot} (s)		t_{inf} (s)	δ (%)
50	56	0.74	0.88	1.62	35×	0.01	0.54	0.28	1.32	1.60	35×	0.02	0.38
100	113	1.00	0.55	1.55	73×	0.03	0.44	0.75	1.57	2.32	49×	0.04	0.54
200	229	1.15	1.13	2.28	100×	0.09	0.71	0.66	1.08	1.74	132×	0.09	0.80
400	604	2.06	3.29	5.35	113×	0.16	0.43	1.59	4.82	6.41	94×	0.16	1.08
600	754	2.87	3.67	6.54	115×	0.23	0.54	2.24	4.20	6.44	117×	0.25	0.84
800	1027	4.25	15.60	19.85	52×	0.25	0.59	3.83	15.61	19.44	53×	0.33	0.96
900	1263	4.38	15.56	19.94	63×	0.31	0.57	3.88	19.34	23.22	54×	0.35	0.43
1000	1456	4.52	20.95	25.47	57×	0.34	0.60	4.15	28.89	33.04	44×	0.42	0.94

and the number of joints. During the inference phase, the proposed approach can be viewed as a closed-form expression for stress solutions and demonstrates satisfactory computation speed, requiring less than 0.42s to achieve stress distribution at 20 equivalent sampled locations per segment. For stress assessment on interconnect trees with 2,000, 4,000, and 6,000 segments, the proposed method requires 57s, 198s, and 485s to converge to a loss of 5×10^{-6}, demonstrating scalability for large-scale interconnects. In these experiments, ν is gradually increased to 6, and the neural network is expanded to 80 neurons per layer, with an RFF embedding size of $m=40$.

During the void nucleation phase, the relative errors of the model proposed by [21] for solving stress evolution along the 50, 100, and 200-segment interconnect trees, as depicted in Table III, are 1.61%, 18.82%, and 3.99% after total training times of 77.41s, 124.44s, and 400.71s, respectively. The method in [21] employs an MLP to approximate stress dynamics at all junctions. In the 100-segment interconnect tree test case, the stress distributions at all junctions exhibit the highest variance among the three cases, and thus cannot be accurately captured. Furthermore, in the proposed partitioning-based approach, degree-i ($i=3,4$) junctions are more likely to be regarded as joints, which exhibit a narrower stress distribution compared with junctions of $i=1,2$. Thereby, in addition to benefiting from the reduction in the size of the training dataset induced by partitioning, the proposed method further enhances its approximation ability by narrowing the distribution of desired outputs, achieved by focusing on stress at degree-i ($i=3,4$) junctions. The proposed approach demonstrates a speedup of up to $87\times$ over the model from [21] in the three test cases, with higher accuracy in stress analysis. Furthermore, we compare the proposed approach with the semi-analytical method [37] on 100- and 500-segment interconnect trees. The semi-analytical method requires 1.38s and 67s to calculate the EM-induced stress during the void nucleation phase, whereas the proposed approach requires 0s, 0.27s, and 0.39s for training the stress model after dividing the interconnect trees into 1, 2, and 3 partitions in the 100-segment case, and 0s, 0.28s, and 0.41s in the 500-segment case. It should be noted that in cases where partitioning the interconnect tree is unnecessary, the proposed method is reduced to analytical expressions. The corresponding inference times are 0.002s, 0.006s, and 0.008s for the 100-segment case and 0.002s, 0.007s, and 0.009s for the 500-segment case, respectively, increasing with the number of partitions and showing a more significant impact on runtime than the segment scale. The proposed approach achieves runtime reductions in both training and inference phases compared with the semi-analytical method.

VI. CONCLUSION

This work introduces a novel partitioning-based approach to predict stress evolution along interconnect trees during the void nucleation and void growth phases, utilizing a lightweight neural network. Initially, we present a partitioning strategy to divide a complex interconnect tree into individual bricks through specific joints and derive analytical solutions for each brick. The learning framework is then employed to fine-tune the solutions along bricks, particularly at the intersection junctions connected with neighboring bricks. By simplifying the physics-based constraints through partitioning, the required amount of training data for the learning task is significantly reduced. Moreover, we incorporate random Fourier features before passing inputs to a multilayer perceptron to enhance the capacity of the neural network to learn features under customized frequencies. The proposed method can be extended to account for the influence of varying physical parameters and operating conditions on stress evolution during EM phases. Experimental results indicate significant computational savings while maintaining competitive accuracy compared with the existing methods. The proposed approach captures continuous multi-order stress while considering dynamic current density, offering a practical alternative to discretization-dominant numerical methods. Although current density simulation is not included here, the promising characteristics of the proposed approach suggest potential expansion for comprehensive electromigration assessments, including void event prediction related to IR drop.

APPENDIX

A. Derivation of Stress Evolution along T-shaped Bricks

In this section, we derive the analytical solution of stress evolution along the T-shaped interconnect trees during the void nucleation phase. Based on Eq. (2), we establish the trial solution for the j-th segment in the Laplace domain:

$$\sigma_j(x,s) = A_j e^{mx} + B_j e^{-mx}, \tag{36}$$

where $m=\sqrt{s/\kappa}$ and $\kappa_1=\kappa_2=\kappa_3=\kappa$. We establish the following BCs for a T-shaped interconnect tree

$$\begin{aligned}\kappa\Big(\frac{\partial\sigma_1(x,s)}{\partial x}|_{x=L_0}+G_1\Big)+\kappa\Big(\frac{\partial\sigma_2(x,s)}{\partial x}|_{x=L_0}+G_2\Big)\\ +\kappa\Big(\frac{\partial\sigma_3(x,s)}{\partial x}|_{x=L_0}+G_3\Big)=0,\end{aligned} \tag{37}$$

$$\sigma_1(x,s)|_{x=L_0}=\sigma_2(x,s)|_{x=L_0}=\sigma_3(x,s)|_{x=L_0}, t>0, \tag{38}$$

$$\begin{aligned}\kappa\Big(\frac{\partial\sigma_1(x,s)}{\partial x}|_{x=L_1}+G_1\Big)=0,\\ \kappa\Big(\frac{\partial\sigma_2(x,s)}{\partial x}|_{x=L_2}+G_2\Big)=0,\\ \kappa\Big(\frac{\partial\sigma_3(x,s)}{\partial x}|_{x=L_3}+G_3\Big)=0,\end{aligned} \tag{39}$$

Algorithm 1 Vectors used for stress evolution assessment.

Input: Max iteration number i_{max}, index of segment j, index of junction where void occurs v (v = NA during void nucleation phase), number of segments M, length of the i-th segment L_i, a set of indices of segments C;

Output: Vectors $\boldsymbol{\mathcal{H}}^C$, $\boldsymbol{\mathcal{H}}^*$, $\boldsymbol{h}^C$, $\boldsymbol{h}^*$, $\mathbf{K}^{i,M}$, $\mathbf{L}^{i,M}$.

Initialize empty $\boldsymbol{\mathcal{H}}^C = [\ ]$, $\boldsymbol{h}^C = [\ ]$.

Initialize $\mathbf{L}^{0,M} = [0]$, $\mathbf{L}^{1,M} = [2\sum_{i=1}^{M} L_i]$,

$\mathbf{K}^{0,M} = [1]$ and $\boldsymbol{\mathcal{H}}^* = [\sum_{\substack{i=1,\\ i\neq j}}^{M} L_i, -\sum_{\substack{i=1,\\ i\neq j}}^{M} L_i]$.

if $v ==$ **NA then**

  $\mathbf{K}^{1,M} = [1]$, $\boldsymbol{h}^* = [M, -\lfloor M/2 \rfloor]$,

else

  $\mathbf{K}^{1,M} = [-1]$, $\boldsymbol{h}^* = [M, \lfloor M/2 \rfloor]$,

end if

for $c = 1 \rightarrow M$ **do**

  if $c \notin C$ **then**

    $\boldsymbol{\mathcal{H}}^C = \mathcal{F}((\boldsymbol{\mathcal{H}}^C)^T + [c, -c])$,

    if $c == v$ **then**

      $\boldsymbol{h}^C = \mathcal{F}((\boldsymbol{h}^C)^T \cdot [1, -1])$,

    else

      $\boldsymbol{h}^C = \mathcal{F}((\boldsymbol{h}^C)^T \cdot [1, 1])$,

    end if

  end if

  if $c \neq j$ **then**

    $\boldsymbol{\mathcal{H}}^* = \boldsymbol{\mathcal{H}}^* \| -2L_c + \sum_{\substack{i=1,\\ i\neq j}}^{M} L_i$,

    if $c == v$ **then**

      $\boldsymbol{h}^* = \boldsymbol{h}^* \| -\lfloor M/2 \rfloor$,

    else

      $\boldsymbol{h}^* = \boldsymbol{h}^* \| \lfloor M/2 \rfloor$,

    end if

  end if

  $\mathbf{L}^{1,M} = \mathbf{L}^{1,M} \| 2L_c \| 2\sum_{\substack{i=1,\\ i\neq n}}^{M} L_i$,

  if $v ==$ NA **then**

    $\mathbf{K}^{1,M} = \mathbf{K}^{1,M} \| \frac{2-M}{M} \| \frac{M-2}{M}$,

  else if $c == v$ **then**

    $\mathbf{K}^{1,M} = \mathbf{K}^{1,M} \| \frac{M-2}{M} \| \frac{M-2}{M}$,

  else

    $\mathbf{K}^{1,M} = \mathbf{K}^{1,M} \| \frac{2-M}{M} \| \frac{2-M}{M}$,

  end if

end for

for $i = 2 \rightarrow i_{max}$ **do**

  $\mathbf{L}^{i,M} = \mathcal{F}((\mathbf{L}^{1,M})^{\mathrm{T}} + \mathbf{L}^{i-1,M})$,

  $\mathbf{K}^{i,M} = \mathcal{F}((\mathbf{K}^{1,M})^{\mathrm{T}} \cdot \mathbf{K}^{i-1,M})$.

end for

according to constraints in Eqs. (3), (4) and (5), respectively. Substituting Eq. (36) into Eqs. (37)-(39), we then obtain a linear matrix equation $\boldsymbol{E} \cdot \boldsymbol{\alpha} = \boldsymbol{\beta}$, where E is a 6×6 matrix, $\boldsymbol{\alpha} = [A_1, B_1, A_2, B_2, A_3, B_3]^{\mathrm{T}}$ describing unknown coefficients, $\boldsymbol{\beta} = [-G_1, -G_2, -G_3, 0, -G_\Sigma]^T$, and $G_\Sigma = \sum_{i=1}^{3} G_i$. Solving this equation gives A_j and B_j as follows:

$$
\begin{aligned}
A_j &= \frac{-G_j \sum_q \{\boldsymbol{h}_q^* e^{m\boldsymbol{\mathcal{H}}_q^*}\} - 2\sum_{\substack{k=1,\\ k\neq j}}^{3} \{G_k \sum_q H_{q,k}^-\} + G_\Sigma \sum_q H_q^-}{3m(1-\sum_p \mathbf{K}_p^{1,3} e^{-m\mathbf{L}_p^{1,3}}) e^{-m\sum_{i=1}^{3} L_i}}, \\
B_j &= \frac{-G_j \sum_q \{\boldsymbol{h}_q^* e^{-m\boldsymbol{\mathcal{H}}_q^*}\} - 2\sum_{\substack{k=1,\\ k\neq j}}^{3} \{G_k \sum_q H_{q,k}^+\} + G_\Sigma \sum_q H_q^+}{3m(1-\sum_p \mathbf{K}_p^{1,3} e^{-m\mathbf{L}_p^{1,3}}) e^{-m\sum_{i=1}^{3} L_i}},
\end{aligned}
\tag{40}
$$

where

$$
\begin{aligned}
&H_{q,k}^- = \boldsymbol{h}_q^{\{j,k\}} e^{m(\boldsymbol{\mathcal{H}}_q^{\{j,k\}} - L_j)}, H_{q,k}^+ = \boldsymbol{h}_q^{\{j,k\}} e^{m(\boldsymbol{\mathcal{H}}_q^{\{j,k\}} + L_j)}, \\
&H_q^- = \boldsymbol{h}_q^{\{j\}} e^{m(\boldsymbol{\mathcal{H}}_q^{\{j\}} - L_j)}, H_q^+ = \boldsymbol{h}_q^{\{j\}} e^{m(\boldsymbol{\mathcal{H}}_q^{\{j\}} + L_j)},
\end{aligned}
\tag{41}
$$

and using Algorithm 1 we determine the vectors $\mathbf{L}^{1,3}$, $\mathbf{K}^{1,3}$, $\boldsymbol{\mathcal{H}}^C$, $\boldsymbol{\mathcal{H}}^*$, $\boldsymbol{h}^C$ and $\boldsymbol{h}^*$, where q and p are indices that traverse the entire vectors. $\boldsymbol{\mathcal{H}}^C \triangleq \boldsymbol{\mathcal{H}}(C)$ and $\boldsymbol{h}^C \triangleq \boldsymbol{h}(C)$ are transformations mapping C, a set of indices of segments, to vectors in Eq. (41). In Algorithm 1, $\lfloor x \rfloor$ represents the greatest integer less than or equal to x. The calculation operator $A^{l\times 1} + / \cdot B^{1\times l}$ represents the sum/dot product of two matrices $A^{l\times l}$ and $B^{l\times l}$, where the vectors are converted to matrices of uniform size by duplicating specified dimensions of the input vectors. The reshape transform $\mathcal{F}$ follows $\mathcal{F}(X) : X \in \mathbb{R}^{l\times p} \rightarrow X \in \mathbb{R}^{1\times lp}$. The operator $a \| b$ represents the concatenation of a vector $a \in \mathbb{R}^n$ and an element $b \in \mathbb{R}$, resulting in a new vector $c \in \mathbb{R}^{n+1}$.

The coefficients A_j and B_j can be converted into infinite Taylor series, resulting in the infinite terms in Eq. (21). Using Eqs. (36) and (40), the stress evolution can be obtained through the inverse Laplace transform, by introducing the basic function (10). This ends the derivation of stress evolution along a T-shaped interconnect tree during the void nucleation phase. The derivation of stress evolution along bricks with a degree-4 junction follows a process similar to that used for a degree-3 junction. Stress analysis during the void growth phase can be conducted following the method in [15] which utilizes the Residue Theorem.

References

[1] I. A. Blech, "Electromigration in thin aluminum films on titanium nitride," *Journal of Applied Physics*, vol. 47, no. 4, pp. 1203–1208, 1976.

[2] J. Black, "Electromigration failure modes in aluminum metallization for semiconductor devices," *Proceedings of the IEEE*, vol. 57, no. 9, pp. 1587–1594, 1969.

[3] V. Sukharev *et al.*, "Electromigration check: Where the design and reliability methodologies meet," *IEEE Transactions on Device and Materials Reliability*, vol. 18, no. 4, pp. 498–507, 2018.

[4] R. Rosenberg *et al.*, "Void Formation and Growth During Electromigration in Thin Films," *Journal of Applied Physics*, vol. 42, no. 13, pp. 5671–5679, 1971.

[5] M. Shatzkes *et al.*, "A model for conductor failure considering diffusion concurrently with electromigration resulting in a current exponent of 2," *Journal of Applied Physics*, vol. 59, no. 11, pp. 3890–3893, 1986.

[6] J. J. Clement *et al.*, "Numerical investigations of the electromigration boundary value problem," *Journal of Applied Physics*, vol. 71, no. 4, pp. 1729–1731, 1992.

[7] M. A. Korhonen *et al.*, "Stress evolution due to electromigration in confined metal lines," *Journal of Applied Physics*, vol. 73, no. 8, pp. 3790–3799, 1993.

[8] S. P. Hau-Riege *et al.*, "Experimental characterization and modeling of the reliability of interconnect trees," *Journal of Applied Physics*, vol. 89, no. 1, pp. 601–609, 2001.

[9] X. Huang *et al.*, "Physics-based electromigration assessment for power grid networks," in *51st Design Automation Conference (DAC)*, 2014, pp. 1–6.

[10] S. Chatterjee *et al.*, "Power grid electromigration checking using physics-based models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 7, pp. 1317–1330, 2018.

[11] C. Cook *et al.*, "Fast electromigration stress evolution analysis for interconnect trees using Krylov subspace method," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 26, no. 5, pp. 969–980, 2018.

[12] H. Zhao *et al.*, "Postvoiding FEM analysis for electromigration failure characterization," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 26, no. 11, pp. 2483–2493, 2018.

[13] H.-B. Chen *et al.*, "Analytical modeling and characterization of electromigration effects for multibranch interconnect trees," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 11, pp. 1811–1824, 2016.

[14] M. A. Al Shohel *et al.*, "Analytical modeling of transient electromigration stress based on boundary reflections," in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2021, pp. 1–8.

[15] T. Hou *et al.*, "Analytical post-voiding modeling and efficient characterization of EM failure effects under time-dependent current stressing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 12, pp. 4959–4972, 2023.

[16] C.-K. Cheng *et al.*, "Module placement based on resistive network optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 3, no. 3, pp. 218–225, 1984.

This article has been accepted for publication in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. This is the author's version which has not been fully edit
content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2025.3567885

[17] S.-J. Chen *et al.*, "Tutorial on VLSI partitioning," *VLSI design*, vol. 11, no. 3, pp. 175–218, 2000.

[18] W. Jin *et al.*, "Emgraph: Fast learning-based electromigration analysis for multi-segment interconnect using graph convolution networks," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, 2021, pp. 919–924.

[19] W. Jin *et al.*, "Data-driven electrostatics analysis based on physics-constrained deep learning," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021, pp. 1382–1387.

[20] T. Hou *et al.*, "A space–time neural network for analysis of stress evolution under dc current stressing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 12, pp. 5501–5514, 2022.

[21] T. Hou *et al.*, "Multilayer perceptron-based stress evolution analysis under DC current stressing for multisegment wires," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 2, pp. 544–557, 2023.

[22] S. Lamichhane *et al.*, "Postpinn-em: Fast post-voiding electromigration analysis using two-stage physics-informed neural networks," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2023, pp. 1–9.

[23] T. Hou *et al.*, "A deep learning framework for solving stress-based partial differential equations in electromigration analysis," *ACM Transactions on Design Automation of Electronic Systems*, vol. 28, no. 4, 2023.

[24] M. Raissi *et al.*, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[25] V. M. Dwyer, "An investigation of electromigration induced void nucleation time statistics in short copper interconnects," *Journal of Applied Physics*, vol. 107, no. 10, p. 103 718, 2010.

[26] S. P. Hau-Riege *et al.*, "The effects of the mechanical properties of the confinement material on electromigration in metallic interconnects," *Journal of Materials Research*, vol. 15, no. 8, pp. 1797–1802, 2000.

[27] V. Sukharev *et al.*, "Postvoiding stress evolution in confined metal lines," *IEEE Transactions on Device and Materials Reliability*, vol. 16, no. 1, pp. 50–60, 2015.

[28] F. N. Najm, "Equivalent circuits for electromigration," *Microelectronics Reliability*, vol. 123, p. 114 200, 2021.

[29] H.-B. Chen *et al.*, "Analytical modeling and characterization of electromigration effects for multibranch interconnect trees," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 11, pp. 1811–1824, 2016.

[30] A. Jacot *et al.*, "Neural tangent kernel: Convergence and generalization in neural networks," in *Advances in Neural Information Processing Systems*, vol. 31, 2018, pp. 8571–8580.

[31] J. Lee *et al.*, "Wide neural networks of any depth evolve as linear models under gradient descent," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 8572–8583.

[32] S. Wang *et al.*, "On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks," *Computer Methods in Applied Mechanics and Engineering*, vol. 384, p. 113 938, 2021.

[33] S. Arora *et al.*, "On exact computation with an infinitely wide neural net," in *Advances in Neural Information Processing Systems*, 2019, pp. 8141–8150.

[34] N. Rahaman *et al.*, "On the spectral bias of neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97, 2019, pp. 5301–5310.

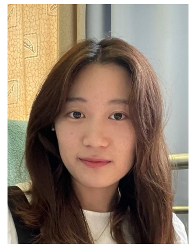
[35] M. Tancik *et al.*, "Fourier features let networks learn high frequency functions in low dimensional domains," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 7537–7547.

[36] A. Sangiovanni-Vincentelli *et al.*, "Node-tearing nodal analysis," Tech. Rep. UCB/ERL M582, 1976. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/1976/29034.html.

[37] X. Wang *et al.*, "Fast physics-based electromigration analysis for full-chip networks by efficient eigenfunction-based solution," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 3, pp. 507–520, 2021.

[38] "COMSOL multiphysics," https://www.comsol.com.

[39] M. A. Al Shohel *et al.*, "Frequency-domain transient electromigration analysis using circuit theory," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2023, pp. 1–8.



Tianshu Hou received the B.S. degree in Electronic Information Science and Technology from Sichuan University, Sichuan, China, in 2019, and the Ph.D. degree in Electronic Science and Technology from Shanghai Jiao Tong University, Shanghai, China, in 2025. She was a visiting scholar at The University of Hong Kong, Hong Kong, in 2023, and at the University of Toronto, Ontario, Canada, in 2024. She is currently a postdoctoral research fellow in the Department of Computer Science and Engineering at The Chinese University of Hong Kong. Her research interests include electromigration reliability modeling, assessment, and optimization, as well as resistive random access memory (RRAM) modeling and assessment.

Farid N. Najm (Fellow, IEEE)) is a Professor in the Edward S. Rogers Sr. Department of Electrical and Computer Engineering (ECE) at the University of Toronto, Ontario, Canada. He received the B.E. degree in electrical engineering from the American University of Beirut in 1983 and the Ph.D. degree in ECE from the University of Illinois at Urbana-Champaign (UIUC) in 1989. From 1989 to 1992, he worked with Texas Instruments, Dallas, Texas. He then joined the ECE department at UIUC as an Assistant Professor and became Associate Professor in 1997. In 1999, he joined the ECE Department at the University of Toronto, where he became Professor in 2001, then served as Department Vice-Chair from 2004 to 2007 then as Department Chair from 2009 to 2019. His research is on Electronic Design Automation (EDA) for VLSI, with a focus on circuit level issues related to power, reliability, timing, and variability.

Dr. Najm is a Fellow of the IEEE and of the Canadian Academy of Engineering. He was Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS from 1997 to 2002, and Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS from 2001 to 2009. He has received an IEEE Transactions on CAD Best Paper Award, an NSF Research Initiation Award, an NSF CAREER Award, the Design Automation Conference (DAC) Prolific Author Award, and several Best Paper Awards from the International Conference on Computer-Aided Design (ICCAD). He served on the executive committee of the International Symposium on Low-Power Electronics and Design (ISLPED) 1999-2013 and has served as TPC Chair and General Chair for ISLPED. In 2010, Dr. Najm authored the book *Circuit Simulation* (John Wiley and Sons, New York).

Ngai Wong (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees in electrical and electronic engineering from The University of Hong Kong (HKU), Hong Kong. He was a Visiting Scholar with Purdue University, West Lafayette, IN, USA, in 2003. He is currently an Associate Professor with the Department of Electrical and Electronic Engineering, HKU. He was the Associate Editor of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems from Jan 2014-Jun 2022, and has served as the track chair and member in the technical program committees (TPCs) of premier EDA conferences every year including DAC, ICCAD and ASP-DAC. He co-founded the IEEE Council on EDA (CEDA), Hong Kong Chapter in 2016, and was the Chair in 2018/19. His research interests include compact neural network design, compute-in-memory (CIM) AI chips, electronic design automation (EDA) and tensor algebra.

Hai-Bao Chen (Senior Member, IEEE) received the B.S. degree in information and computing sciences and the M.S. and Ph.D. degrees in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2006, 2008, and 2012, respectively. He then joined Huawei Technologies, where he focused on cloud computing and big data. He was a Post-Doctoral Research Fellow with the Department of Electrical Engineering, University of California, Riverside, CA, USA, from 2013 to 2014. He is currently an Associate Professor with the Department of Micro/Nano-Electronics, Shanghai Jiao Tong University, Shanghai, China. He has authored or coauthored about 100 papers in scientific journals and conference proceedings. His current research interests include VLSI reliability, machine learning and neuromorphic computing, numerical analysis and modeling for VLSIs, and integrated circuits for signal and control systems. He received one Best Paper Award nomination from Asia and South Pacific Design Automation Conference (ASP-DAC) in 2015. He has been serving as an Associate Editor for Integration—The VLSI Journal since 2016.