Physics-Informed Learning Based Multiphysics Simulation for Fast Transient TSV Electromigration Analysis

XIAOMAN YANG, Shanghai Jiaotong University, Shanghai, China
HAIBAO CHEN, Shanghai Jiaotong University, Shanghai, China
YUHAN ZHANG, University of Michigan, Ann Arbor, United States
TIANSHU HOU, Shanghai Jiao Tong University, Shanghai, China
PENGPENG REN, Shanghai Jiaotong University, Shanghai, China
RUNSHENG WANG, Peking University, Beijing, China
ZHIGANG JI, Shanghai Jiao Tong University, Shanghai, China
RU HUANG, Peking University, Beijing, China

Through Silicon Vias (TSVs) are vulnerable to electromigration (EM) degradation due to their high local current densities, thereby reducing the reliability of 3D ICs with stack dies and TSVs. Due to the broad application of 3D ICs, it is necessary to analyze the electromigration reliability of TSVs. To overcome the weakness of traditional method for EM modeling of TSVs, we propose a physics-informed learning approach for transient analysis of electromigration modeling in TSV by solving the conventional mass balance equation. The proposed method allows simultaneous consideration of atomic depletion and accumulation, effective resistance degradation, electric current evolution, and stress distribution. In particular, we propose a customized neural network to simulate the EM process in TSV without the need for fine grid meshing and temporal iteration in traditional methods. Considering that the loss function of the proposed model is a combination of different loss terms, we propose a modified self-adaptive loss balanced method to automatically adjust the weights of multiple loss terms to enhance network performance. Given the prediction uncertainty due to data randomness or model architecture constraints, Gaussian probabilistic model is constructed to define the self-adaptive weights and update the dynamic weights per epoch built on maximum likelihood estimation. Compared with the finite element method, the proposed physics informed neural network method can lead to a speedup with less than 0.1% mean square error. Experimental results also show that the proposed model achieves excellent performance over other competing methods and high robustness under values of initial weights, different numbers of hidden layers and neurons per layer.

This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant T2293704 and Grant T2293700, and in part by the National Key Research and Development Program of China under Grant 2023YFB4402204.
Authors' Contact Information: Xiaoman Yang, Shanghai Jiaotong University, Shanghai, China; e-mail: yangxm1203@sjtu.edu.cn; Haibao Chen (Corresponding author), Shanghai Jiaotong University, Shanghai, ShangHai, China; e-mail: haibaochen@sjtu.edu.cn; Yuhan Zhang, University of Michigan, Ann Arbor, Michigan, United States; e-mail: zyuhan@umich.edu; Tianshu Hou, Shanghai Jiao Tong University, Shanghai, China; e-mail: houtianshu@sjtu.edu.cn; Pengpeng Ren, Shanghai Jiaotong University, Shanghai, China; e-mail: pengpengren@sjtu.edu.cn; Runsheng Wang, Peking University, Beijing, China; e-mail: wrs@pku.edu.cn; Zhigang Ji, Shanghai Jiao Tong University, Shanghai, China; e-mail: zhigangji@sjtu.edu.cn; Ru Huang, Peking University, Beijing, China; e-mail: ruhuang@pku.edu.cn.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 1084-4309/2025/01-ART18
https://doi.org/10.1145/3706106

CCS Concepts: • **Computing methodologies** → *Model development and analysis;*

Additional Key Words and Phrases: TSV, electromigration, self-adaptive weights, multiphysics fields, physics-informed learning

ACM Reference Format:
Xiaoman Yang, Haibao Chen, Yuhan Zhang, Tianshu Hou, Pengpeng Ren, Runsheng Wang, Zhigang Ji, and Ru Huang. 2025. Physics-Informed Learning Based Multiphysics Simulation for Fast Transient TSV Electromigration Analysis. *ACM Trans. Des. Autom. Electron. Syst.* 30, 2, Article 18 (January 2025), 22 pages. https://doi.org/10.1145/3706106

1 Introduction

With the increasing demand for lower power consumption, higher performance and integration [1], 3-D packaging has been applied in **integrated circuits** (**ICs**). This approach is achieved by employing stacked dies and **through-silicon-vias** (**TSVs**), which facilitate vertical communication within the circuitry. The conjunction of TSVs with micro-bumps establishes 3D ICs as a cost-effective technology that potentially extends the "More than Moore" scaling principle into the third dimension. 3D ICs can significantly decrease both average wire delay and energy use [2] over conventional 2D chips. Despite these advancements, 3D ICs face challenges particularly with **electromigration** (**EM**) degradation, owing to substantial electric currents and high local current densities in TSVs. This vulnerability significantly threatens the overall reliability [3] of ICs, with potential risks ranging from a reduction in efficiency to short-circuit or open-circuit failures. Therefore, while the strengths of the 3D ICs technology are not to be undermined, it is crucial to comprehensively understand and mitigate these associated risks to ensure optimal functionality.

Electromigration modeling has been extensively investigated over the past few decades as a critical approach to evaluate the EM reliability of TSV. Many researchers conduct experiments to simulate the electrical, stress, thermal, and atomic concentration fields within TSV [4, 5]. Park et al. [6] conducted EM lifetime modeling of TSV structure and explored the impact of TSV-induced stress on EM lifetime of **Back-End-of-Line** (**BEOL**) wires. Advancing these studies, Zhao et al. [7] investigated transient EM modeling for TSV-based 3D connection, opting to consider grain structure in TSV EM modeling instead of modeling the entire TSV as a single crystal. Additionally, Zhou et al. [8] simulated and modeled thermal stress, with these results utilized in the reliability analysis of TSV. However, conventional methods applied in these studies require fine grid meshing and temporal iterations, leading to a considerable computation cost particularly when striving for accurate and self-consistent solutions to **partial differential equations** (**PDEs**).

Recent advancements in **machine learning** (**ML**) [9], especially **physics-informed neural networks** (**PINNs**) [10], have shown significant promise in addressing the challenges associated with solving PDEs [11]. In comparison to traditional neural network approaches, PINNs are trained to undertake supervised learning tasks while concurrently adhering to the established laws of physics. A key feature of PINNs is the application of **automatic differentiation** (**AD**) technology [12], which facilitates the computation of derivative terms. This approach markedly enhances the accuracy of solution over traditional numerical methods, which predominantly depend on discretization techniques. In recent years, there are a number of PINN-based solution libraries. For example, Lu et al. has developed an innovative Python library, DeepXDE [13], dedicated to optimizing the application of PINNs. PINNs have been applied in many engineering fields [14–16], such as electromagnetic simulation, heat transfer and so on. There are many recent studies that have focused on the application of machine learning and PINN-based methods to EM analysis. Hou et al. [17] introduced a **space–time physics-informed neural network**

(**STPINN**), specifically tailored for solving Korhonen Equation in multisegment interconnect trees under various thermal conditions in void nucleation phase. Jin et al. [18] developed HierPINN-EM, a pioneering hierarchical PINN approach aimed at accelerating the analysis of EM-induced stress in multi-segment interconnects. Additionally, Hou et al. [19] explored the use of **multilayer perceptrons** (**MLPs**) for predicting the evolution of stress within interconnect trees during void nucleation stage. Lamichhane [20] introduced PostPINN-EM for addressing PDEs governing the stress evolution in confined metal multisegment interconnect trees throughout post-voiding phase.

The loss function of PINN consists of several key components: the observed data, the boundary and initial conditions, and the PDE residuals. It is noted that the performance and convergence speed of PINN are significantly influenced by the weights attributed to the various components of the loss function. Therefore, many researches focused on adaptively balancing the interplay among different loss components to improve the performance of PINN [21]. Kim et al. [22] proposed the **dynamic pulling method** (**DPM**) to dynamically adjust the gradients of loss terms during optimization. Wang et al. [23] explored and analyzed a fundamental failure mechanism inherent to PINN and introduced a learning rate annealing algorithm to harmonize the interactions among different elements of loss functions during training process. However, given the lack of theoretical basis specific to the training dynamics of PINN within this learning rate annealing strategy, Wang et al. [24, 25] proposed the **Neural Tangent Kernel** (**NTK**) Perspective to understand the gradient propagation. Through this perspective, the allocation of weights to loss terms was informed by the NTK, which is adjusted per epoch during optimization. Furthermore, Chen et al. [26] introduced the **Gradient Normalization** (**GradNorm**) algorithm, designed to auto-regulate training intensity across deep multitask models by dynamically adjusting the magnitudes of gradients. Heydari [27] proposed the SoftAdapt method, which modified loss weights within multi-part loss functions in response to real-time performance metrics of the individual loss components.

In this article, we propose a physic-informed neural network for transient modeling of electromigration in TSV. The key advancement of this research lies in employing PINN for transient EM modeling of TSV, surpassing conventional approaches that suffer from drawbacks. While the Analytical Model offers simplicity, efficiency, and parallelism, its precision is constrained by a limited parameter set. The **Finite Difference Method** (**FDM**), versatile across accuracy levels and parallel-friendly, struggles with irregular geometries and grid adaptability. The **Finite Element Method** (**FEM**), renowned for its ability to tackle irregular shapes and boundaries, demonstrates strong grid adaptability but comes at the cost of high computational demands and memory usage. To overcome limitations of traditional methods in TSV modeling, PINN are introduced as an alternative and can model the rapid electromigration of TSV with both accuracy. The main contributions are as follows:

— We propose a physics-informed learning method to predict the evolution of atomic concentration, stress distribution, and electric current within TSV. To the best of our knowledge, this is the *first study* that utilizes the learning capabilities of physics-informed neural network for multiphysics analysis of EM modeling in TSV.
— We propose a neural network for solving the traditional mass balance equation in 3-D axissymmetric geometry. This well-tailored framework demonstrates consistency with multiphysics kinetics in EM modeling of TSV.
— We improve the self-adaptive loss balanced method and employ it in the neural network to enhance the performance of structure. We construct a Gaussian probabilistic model to calculate the self-adaptive weights and utilize the maximum likelihood estimation to automatically update the weights.

The remainder of this article is organized as follows. Section 2 describes the fundamental PDEs for transient EM modeling in TSV, taking into account electrical, atomic concentration and stress fields. In Section 3, we present the customized physics-informed learning framework for EM modeling in TSV and introduce the modified self-adaptive loss balanced method utilized in this framework. Experimental results are presented in Section 4, and the conclusion is provided in Section 5.

2 Transient EM Modeling in TSV

2.1 Fundamental PDEs

For the analysis of EM process in TSV, it is crucial to understand the dynamics of atomic movements within structures. A set of well-established PDEs can be employed to model the complex, transient evolution of atomic concentration, denoted as N, at any given point (x, y, z) and time t. These equations serve as the foundation for simulating how atoms distribute themselves over time. The conventional mass balance equation is used to simulate the EM process in TSV, given by

$$\frac{\partial N}{\partial t} + \nabla \cdot \mathbf{J} = 0, \tag{1}$$

Equation (1) encapsulates the law of conservation of mass in TSV, where $\mathbf{J}$ is the total atomic flux described by

$$\mathbf{J} = \mathbf{J_C} + \mathbf{J_S} + \mathbf{J_T} + \mathbf{J_N}. \tag{2}$$

In Equation (2), each term represents a distinct mechanism driving the movement of atoms: $\mathbf{J_C} = \frac{N}{kT} eZ^* \rho \mathbf{j} D$ is the current-induced atomic flux, propelled by electrical forces; $\mathbf{J_S} = \frac{N\Omega}{kT} D \nabla \sigma_H$ is the stress-induced atomic flux, driven by mechanical stresses within the material; $\mathbf{J_T} = -\frac{NQ^*}{kT^2} D \nabla T$ is the temperature-gradient-induced atomic flux, a consequence of thermal gradients; $\mathbf{J_N} = -D \nabla N$ is the atomic flux due to atom diffusion, resulting from the natural tendency of atoms to move from regions of high to low concentration. Here, e is electrical charge, Z^* is effective valence charge, ρ is electrical resistivity, k is Boltzmann constant, Ω is atomic volume, Q^* is heat of transport, $\mathbf{j}$ is current density, T is temperature, and $D = D_0 \exp(-\frac{E_A}{kT})$ is the atomic diffusion coefficient where D_0 is self-diffusion coefficient, E_A is activation energy. It should be noted that $\mathbf{J_S}$ is governed by the the gradient of the hydrostatic stress σ_H. The hydrostatic stress primarily involves the back flow stress defined as $\sigma_{bf} = B\frac{N - N_0}{N_0}$, where B is the back flow stress modulus. From the experiment results [7], we find that thermal expansion stress has a minor effect on the overall electromigration atomic transport. Therefore, we only consider the back flow stress as the component of the hydrostatic stress. The TSV-based 3D connection can be influenced by the following three types of stress:

— Back flow stress occurs due to electromigration in TSV and is determined by the distribution of atomic concentration. As atoms accumulate at the anode and deplete at the cathode, it causes the stress gradient throughout TSV. Back flow stress is usually co-simulated with EM process and will increase gradually.
— As a result of differential thermal expansion or contraction, thermal residual stress arises during the fabrication process [28] and remain in the material after the temperature has returned to normal. This kind of stress arising from runtime temperature fluctuation is generally ignored [7, 29] in simulation.
— Thermal stress arises from coefficient of thermal expansion mismatch among different materials [30] and the presence of elevated thermal gradients [31]. Mechanical expansion or contraction can occur at different rates, attributable to the temperature gradients that is caused by uneven local heating.

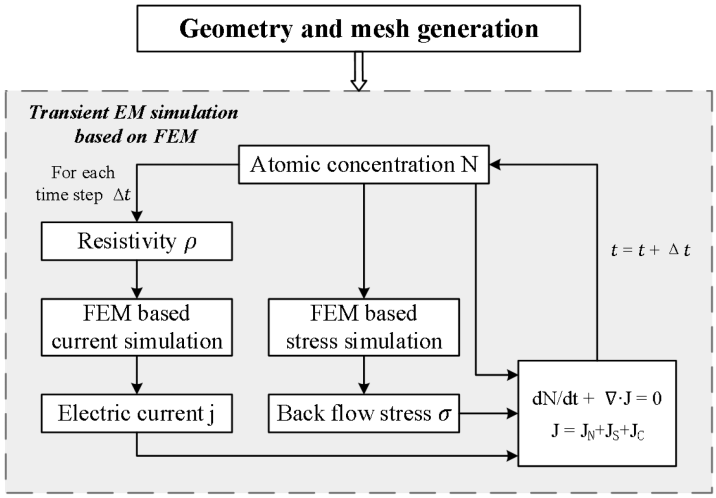


Fig. 1. Flow of transient EM simulation in TSV.

Electromigration refers to the movement of metal ions within a conductor due to the momentum transfer [32] of electrons under high current densities, which can lead to the accumulation of metal ions at certain points and the depletion at others. The changes of atom concentration due to electromigration can cause significant structural changes within the TSV, such as the formation of voids [33] at the cathode and hillocks or whiskers at the anode. These structural changes alter the cross-sectional area available for current flow through the TSV, directly impacting its electrical resistance. When voids form within the TSV, the effective cross-sectional area decreases, leading to an increase in electrical resistance. Conversely, the formation of hillocks might temporarily decrease resistance, which can cause short-circuit failures if these protrusions contact with adjacent structures. Due to the presence of metal barrier layers, open-circuit failures caused by voids occur more frequently than short-circuit failures arising from hillocks. Therefore, our analysis primarily focuses on voids-related failures in TSV. For simplicity, we choose the strategy in work [7] to model the TSV resistivity as a function of atomic concentration to estimate the resistance of TSV. In our work, we assume that the formation of voids will occur in TSV if the atom concentration is less than $0.85N_0$. When the atom concentration is more than $0.9N_0$, we assume the resistivity of TSV to be ρ_0. From the experiment of measuring resistivity in [7], we assign a resistivity value of $32\rho_0$ to voids and make a linear approximation of the resistivity between $0.85N_0$ and $0.9N_0$. The simplified model for TSV resistivity is presented as follows:

$$\rho = \begin{cases} 32\rho_0, \ 0 < N < 0.85N_0, \\ 32\rho_0 - 620 \times (\frac{N}{N_0} - 0.85)\rho_0, \ 0.85N_0 \leq N \leq 0.9N_0, \\ \rho_0, \ 0.9N_0 < N < 5N_0. \end{cases} \tag{3}$$

2.2 FEM-based Multiphysics Simulation

The FEM analysis tool, COMSOL Multiphysics software, can be employed for transient EM modeling and simulation in TSV. The procedural flow of this simulation process is illustrated in Figure 1. This process starts with the definition and construction of TSV geometry. The prototypical TSV model involves a copper TSV with $5\mu m$ diameter and $25\mu m$ height and landing pads with $6\mu m \times 10\mu m$ and $1\mu m$ thick. In this test case, the top of the TSV is configured as the voltage input, while the bottom is established as ground. The transient EM simulation of TSV is

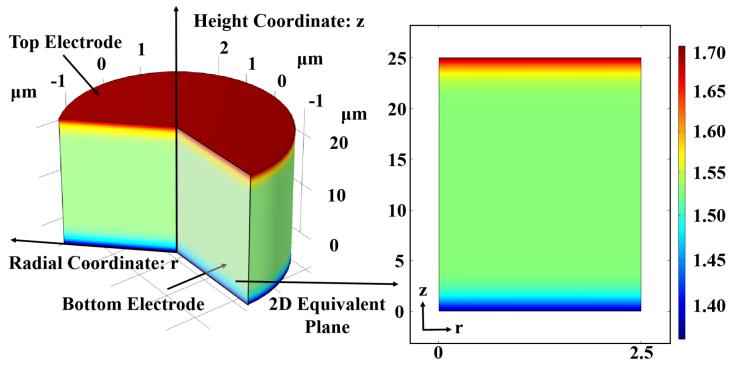


Fig. 2. Converting the 3-D symmetrical physics-based model into 2-D model by mapping from the Cartesian to the Cylindrical coordinate system.

then performed as an iterative loop. At each time step Δt, the atomic concentration is utilized to calculate the resistivity distribution based on the predefined resistivity function. Concurrently, the back flow stress is determined by the equation $\sigma = B\frac{N-N_0}{N_0}$. Then, the FEM analysis tool can be used to perform the simulation of electric current and calculate the current density distribution $\mathbf{j}$. Subsequent to these results, the process advances to mesh generation which is critical for spatial discretization in the simulation. Next, the atomic fluxes are updated: (1) Flux $\mathbf{J}_C$ arising from $\mathbf{j}$, N, and ρ; (2) Flux $\mathbf{J}_S$ resulting from the gradient of σ; and (3) Flux $\mathbf{J}_N$ caused by the gradient of N. Upon updating these fluxes, the FEM analysis tool proceeds to solve the PDEs, thereby determining the atomic concentration at time $t + \Delta t$. This simulation flow allows us to derive the evolution of atomic concentration, electric current, stress distribution and effective resistance degradation in the TSV structure. In our study, we employ the model from [7] for analysis, leveraging its insight into copper TSVs' minimal thermal gradient due to copper's exceptional thermal conductivity. In [7], joule heating is simulated with 60mA, yielding a marginal temperature range of 349.84 K to 349.90 K, and resulting in a negligible 0.04 K gradient across the TSV. Notably, initial compressive stress from thermal expansion emerges at the 3D connection, but the stress gradient primarily manifests at TSV-wire interfaces, remaining stable over time. Hence, thermal expansion stress has a minor impact on EM atomic transport. Consequently, the atomic flux $\mathbf{J}_\mathrm{T}$ induced by temperature gradients is not considered in the proposed model, and a constant temperature T of 350 K is assumed for the simulation. However, we still recognize the importance of the thermo-mechanical stress and mass flux model in [8] and plan to integrate this aspect in future work.

Figure 2 depicts a three-dimensional, symmetrical, physics-based model. By extracting a cross-sectional slice of TSV, this two-dimensional representation can encapsulate all pertinent information from the three-dimensional model due to the model's axisymmetric characteristics. Furthermore, Figures 3 and 4 illustrate the cross-sectional views of the current density, atomic concentration, and back flow stress within the TSV. Specifically, Figure 3 shows the distribution of current density along r and z coordinates at $t = 0$s and $t = 5\times10^6$s. The current density along r coordinate **ec.Jr** is small since we apply voltage ψ at the top pad. However, the electromigration phenomenon has intensified over time and **ec.Jr** gradually increases. **ec.Jz** will change when the formation of voids occurs due to atomic depletion in TSV. We conduct simulation of time-dependent atomic movements and specifically examine the atomic concentration at both the top and bottom TSV-wire interfaces at $t = 1 \times 10^5$s and $t = 1 \times 10^6$s, with the findings depicted in Figure 4(a) and (b), respectively. The experimental results distinctly indicate the influence of electric current on

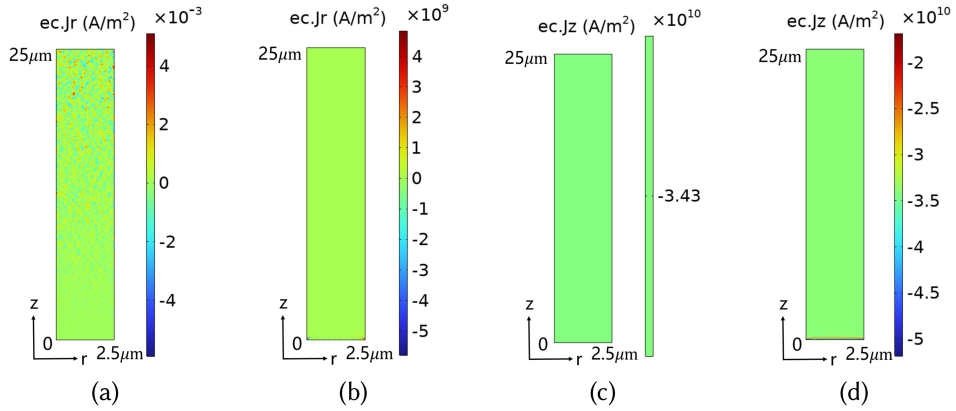


Fig. 3. The distribution of current density along r and z coordinates in a TSV-wire structure: (a) current density along r coordinate ec.Jr at the time $t = 0$s; (b) current density along r coordinate ec.Jr at the time $t = 5 \times 10^6$s; (c) current density along z coordinate ec.Jz at the time $t = 0$s; (d) current density along z coordinate ec.Jz at the time $t = 5 \times 10^6$s.

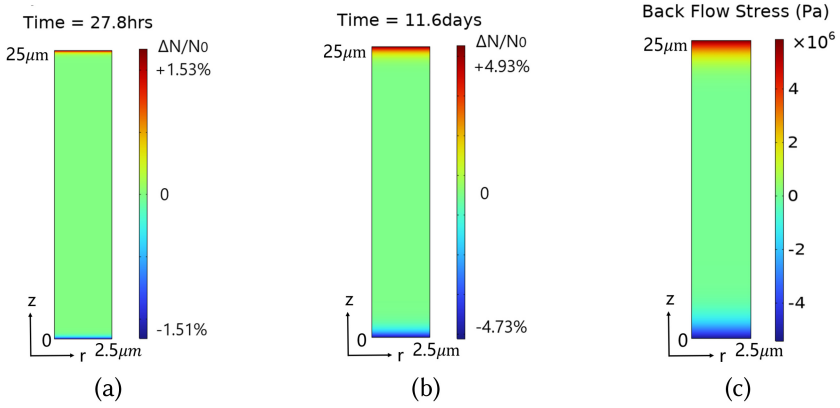


Fig. 4. The change of atomic concentration on TSV structure and the simulation of back flow stress: (a) the change of atomic concentration at the time $t = 1 \times 10^5$s; (b) the change of atomic concentration at the time $t = 1 \times 10^6$s; (c) the simulation of back flow stress at the time $t = 5 \times 10^6$s.

atomic flux. Given the direction of electron flow from the top to the bottom, copper atoms are observed to migrate upwards, propelled by the momentum exchange subsequent to electron collisions. Specifically, the phenomenon of current crowding, significantly observed at the top and bottom TSV-wire interfaces [34], results in atom depletion ($\Delta N/N < 0$) at the bottom interface and atom accumulation ($\Delta N/N > 0$) at the top interface.

The experiment conducted by Zhao et al. [7] provides empirical validation that the thermal gradient within copper TSV is negligible, attributable to copper's high thermal conductivity. Consequently, in our EM analysis of TSV, the time-dependent temperature variation is not considered as a significant factor. The comparative analysis [7] of "No Stress" versus "Thermal Stress" conditions reveals that the ratio $\Delta N/N_0$, indicative of atomic concentration change, remains largely unaffected by thermal stress. This observation suggests that thermal expansion stress has a slight effect on the dynamics of EM-induced atomic transport. Hence, our EM simulation of TSV mainly focuses on the back flow stress. Illustrated in Figure 4(c), the simulation results

depict that the top TSV-wire interface experiences compressive stress as a consequence of atomic accumulation, whereas the bottom TSV-wire interface experiences tensile stress due to atomic depletion.

3 Learning Framework for EM Modeling in TSV

3.1 Problem Formation

As shown in Figure 2, we transform the 3-D symmetrical physics-based model into 2-D model by mapping from the Cartesian to the Cylindrical coordinate system, given by

$$(\nabla \cdot \Gamma)_{(x,y,z)\to(r,z)} = \frac{1}{r}\frac{\partial(r\Gamma_r)}{\partial r} + \frac{\partial \Gamma_z}{\partial z}, \tag{4}$$

$$\nabla \cdot (N\nabla N) = (\nabla N)\cdot(\nabla N) + N(\nabla \cdot \nabla N). \tag{5}$$

Equation (5) can be utilized to optimize the form of the stress-induced atomic flux in Equations (1) and (2). We denote the radial coordinate of the axis of TSV by $r = 0$ and the radius of TSV by $r = R_{tsv}$, respectively. The height coordinates of the **top electrodes** (**TE**) and **bottom electrodes** (**BE**) are set to be $z = h$ and $z = 0$, respectively. Based on Equations (1), (2), (4), and (5), the evolution of atomic concentration in the TSV model can be rewritten as follows:

$$\begin{aligned}
&\frac{\partial N(r,z,t)}{\partial t} + \frac{\partial \mathbf{J_r}}{\partial r} + \frac{\partial \mathbf{J_z}}{\partial z} + \frac{\mathbf{J_r}}{r} = 0,\\
&\mathbf{J_r} = -D\frac{\partial N(r,z,t)}{\partial r} + \mathbf{A_{1r}}DN + A_2 DN\frac{\partial N(r,z,t)}{\partial r},\\
&\mathbf{J_z} = -D\frac{\partial N(r,z,t)}{\partial z} + \mathbf{A_{1z}}DN + A_2 DN\frac{\partial N(r,z,t)}{\partial z},\\
&\mathbf{A_{1r}} = \frac{eZ^*\rho \mathbf{j_r}}{kT}, \mathbf{A_{1z}} = \frac{eZ^*\rho \mathbf{j_z}}{kT}, A_2 = \frac{\Omega B}{kTN_0},
\end{aligned} \tag{6}$$

where $r \in [0, R_{tsv}]$, $z \in [0, h]$, $t > 0$. The **boundary conditions** (**BCs**) corresponding to Equation (6) can be expressed as follows:

$$(\mathbf{n}\cdot\mathbf{J_r})\big|_{r=R_{tsv}} = 0, (\mathbf{n}\cdot\mathbf{J_z})\big|_{z=0,h} = 0. \tag{7}$$

Equation (7) represents that the flux ($\mathbf{J_N} + \mathbf{J_C} + \mathbf{J_S}$) is expected to be zero at the boundaries $z = 0$, $z = h$, and $r = R_{tsv}$. The initial atomic concentration of TSV is set to be N_0, and the IC can be expressed as

$$N(r,z,0) = N_0, \tag{8}$$

To obtain the current densities along r and z coordinates, $\mathbf{j_r}$ and $\mathbf{j_z}$, we need to derive the following PDEs describing the electric field:

$$\nabla \cdot \left(-\frac{1}{\rho}\nabla V(r,z,t)\right) = 0, \tag{9}$$

$$V(r,z,t)\,|_{z=0} = 0,\ V(r,z,t)\,|_{z=h} = \psi, \tag{10}$$

$$\frac{\partial V(r,z,t)}{\partial r}\Big|_{r=R_{tsv}} = 0, \tag{11}$$

where ρ is the resistivity of TSV and calculated by Equation (3). It should be noted that Equation (9) can be derived by $\nabla \cdot \mathbf{j} = \mathbf{0}$, $\mathbf{j} = \mathbf{E}/\rho$ and $\mathbf{E} = -\nabla V$. Equation (10) represents that the top electrodes are the electrical potential input of ψ and the bottom electrodes are ground. Equation (11) indicates electrical insulation on the boundary. The current density along r and z coordinates in Equation (6)

can be calculated by

$$\mathbf{j_r} = -\frac{1}{\rho}\frac{\partial V(r,z,t)}{\partial r}, \quad \mathbf{j_z} = -\frac{1}{\rho}\frac{\partial V(r,z,t)}{\partial z}. \tag{12}$$

3.2 Physics-informed Neural Network

Given an input point in the time and space domain, PINN is capable of generating an approximate solution at that point of a differential equation after training [12]. Fundamentally, the PINN algorithm is a mesh-free technique that is adopted to obtain solutions to PDEs. This is achieved by converting the problem of directly solving the dominant equations into an optimization problem of loss function as traditional **feedforward neural network** (**FNN**) does. Notably, this technique has an advantage in breaking the curse of dimensionality, a prevalent challenge in high-dimensional spaces [9]. The core idea of PINN is to approximate the solution $u(\boldsymbol{x}, t)$ by $\hat{u}(\boldsymbol{x}, t; \theta)$ and the approximation is constrained by the relevant physical equations. Here, θ symbolizes a set of network parameters that define the structure of neural network. Typically, the architecture of PINN is a feed-forward, multi-layer neural network, which comprises an input layer, $M-1$ hidden layers, and an output layer. This configuration is strategically designed to effectively capture and model the complexities inherent in the physical phenomena being studied. In order to derive an approximate solution, it is imperative to restrict the neural network $\hat{u}$ to satisfy the laws of physics as well as the corresponding boundary and initial conditions. This necessitates the construction of loss functions that quantitatively measure the deviation between the predictions of the neural network and the realistic values. The optimization process then involves an iterative search for an optimal set of network parameters θ, and in this process, these loss functions are minimized.

3.3 The Modified Self-adaptive Loss Balanced Method

The core concept of the modified self-adaptive loss balanced method involves developing a multi-task loss function aimed at optimizing the Gaussian likelihood of homoscedastic uncertainty [35, 36]. We improve the self-adaptive loss balanced method by adjusting the coefficients of certain terms and employing the L-BFGS and Adam optimizers to enhance the optimization effect. In our proposed neural network, we first discuss the optimization of $\mathcal{L}_{data}$. We construct a Gaussian probabilistic model, in which the output is denoted as u, the uncertainty parameter is denoted as ε_{data}, the likelihood is defined as a Gaussian with mean provided by the solution of PINN $\hat{u}(r, z, t; \theta)$. The likelihood function p is defined by the following form:

$$p(u|\hat{u}(r,z,t;\theta)) = N\left(\hat{u}(r,z,t;\theta), \varepsilon_{data}^2\right), \tag{13}$$

where $N\left(\hat{u}(r,z,t;\theta), \varepsilon_{data}^2\right)$ represents Gaussian distribution with variance ε_{data}^2. Through maximum likelihood inference, we can minimize the negative log-likelihood function to adjust the parameter ε_{data}. The negative log-likelihood function $-\log p$ is presented as follows:

$$-\log p(u|\hat{u}(r,z,t;\theta)) = \frac{1}{2\varepsilon_{data}^2}\mathcal{L}_{data}(\theta) + \log \varepsilon_{data}. \tag{14}$$

To stabilize the training process, we implement exponential mapping on the likelihood function to obtain $s_{data} = \log \varepsilon_{data}^2$ and then transform Equation (14) into the following form:

$$-\log p(u|\hat{u}(r,z,t;\theta)) = \frac{1}{2}\exp(-s_{data})\mathcal{L}_{data}(\theta) + \frac{1}{2}s_{data}. \tag{15}$$

Furthermore, we consider adding PDE residual of aforementioned Equation (6) into the constructed Gaussian probabilistic model with output (f,u) and uncertainty parameter $\varepsilon_f, \varepsilon_{data}$. The

likelihood function comprises the loss terms of data and PDE residual, $\mathcal{L}_{data}$ and $\mathcal{L}_f$, respectively. The likelihood function p is represented by

$$\begin{aligned} p(f,u|\hat{u}(r,z,t,\theta)) &= p(f|\hat{u}(r,z,t,\theta)) \cdot (u|\hat{u}(r,z,t,\theta)) \\ &= N(\hat{u}(r,z,t,\theta),\varepsilon_f^2) \cdot N(\hat{u}(r,z,t,\theta),\varepsilon_{data}^2). \end{aligned} \tag{16}$$

The negative log-likelihood function $-\log p$ of the multi-output model is given by

$$\begin{aligned} -\log p(f,u|\hat{u}(r,z,t;\theta)) &\propto \frac{1}{2\varepsilon_f^2}\mathcal{L}_f(\theta) + \frac{1}{2\varepsilon_{data}^2}\mathcal{L}_{data}(\theta) + \log \varepsilon_f \varepsilon_{data} \\ &= \frac{1}{2}\left(\exp(-s_f)\mathcal{L}_f(\theta) + \exp(-s_{data})\mathcal{L}_{data}(\theta) + s_f + s_{data}\right), \end{aligned} \tag{17}$$

where $s_f = \log \varepsilon_f^2$. Similarly, we add the loss term of PDE residual in Equation (9) and the loss term of BC, $\mathcal{L}_e$ and $\mathcal{L}_{bc}$, to the constructed Gaussian probabilistic model. The likelihood function of the model with output (f,u,e,b) and uncertainty parameters $\varepsilon_f, \varepsilon_{data}, \varepsilon_e, \varepsilon_{bc}$ can be presented as follows:

$$\begin{aligned} -\log p(f,u,e,b|\hat{u}(r,z,t;\theta)) &\propto \frac{1}{2\varepsilon_f^2}\mathcal{L}_f(\theta) + \frac{1}{2\varepsilon_{data}^2}\mathcal{L}_{data}(\theta) + \frac{1}{2\varepsilon_e^2}\mathcal{L}_e(\theta) + \frac{1}{2\varepsilon_{bc}^2}\mathcal{L}_{bc}(\theta) + \log \varepsilon_f \varepsilon_{data} \varepsilon_e \varepsilon_{bc} \\ &= \frac{1}{2}\big(\exp(-s_f)\mathcal{L}_f(\theta) + \exp(-s_{data})\mathcal{L}_{data}(\theta) + \exp(-s_{bc})\mathcal{L}_{bc}(\theta) + \exp(-s_e) \\ &\quad \mathcal{L}_e(\theta) + s_f + s_{data} + s_e + s_{bc}\big), \end{aligned} \tag{18}$$

where $s_e = \log \varepsilon_e^2$, $s_{bc} = \log \varepsilon_{bc}^2$. Furthermore, the implement of exponential mapping enables the unconstrained minimization of the loss function. As this mapping ensures that $\exp(-s^*)$ resolves to a positive domain, the adaptive weight s^* is prevented from converging prematurely to zero. Such a design enhances the numerical stability of the training process, making it more robust and reliable in handling the complexities in EM modeling of TSV. In our methodology, we utilize both the Adam and L-BFGS optimizers to find the optimized model weights θ^* and to obtain the final adaptive weights s^* through the minimization of the loss function. Within this model, it is imperative to understand the role of adaptive weights: a larger adaptive weight s^* enhances the influence of its corresponding loss term, whereas a smaller adaptive weight diminishes its impact and imposes a lower penalty on the model. This mechanism serves as a pivotal strategy for the automatic adjustment of the adaptive weight associated with each loss component.

3.4 Scaling Scheme and Data Preparation

In this section, we propose a scaling scheme utilized to address the limitation that the related coefficients differ by a few orders of magnitude. First, we rewrite the conventional mass balance equation $\frac{\omega_N \partial \hat{N}}{\omega_t \partial \hat{t}} + \frac{\partial \hat{\mathbf{j}}_\mathbf{r}}{\omega_x \partial \hat{r}} + \frac{\partial \hat{\mathbf{j}}_\mathbf{z}}{\omega_x \partial \hat{z}} + \frac{\hat{\mathbf{j}}_\mathbf{r}}{\omega_x \hat{r}} = 0$, where $N = \omega_N \hat{N}$, $r = \omega_x \hat{r}$, $z = \omega_x \hat{z}$, $t = \omega_t \hat{t}$ are the original data without the scaling in the transient EM modeling, whereas $\hat{r}, \hat{z}, \hat{t}, \hat{N}$ are the scaled location, time and atomic concentration. The scaling factors are set to be $\omega_N = N/\hat{N} = 1 \times 10^{46}$, $\omega_{\boldsymbol{x}} = r/\hat{r} = z/\hat{z} = 1 \times 10^{-6}$, and $\omega_t = t/\hat{t} = 1 \times 10^6$. The scaled back flow stress $\hat{\sigma}$ can be calculated based on $\hat{\sigma} = B\frac{\hat{N}-\hat{N}_0}{\hat{N}_0}$, where $\omega_\sigma = \sigma/\hat{\sigma} = 1 \times 10^6$. Then, we rewrite the atomic fluxes $\hat{\mathbf{J}}_\mathbf{r} = (-D\frac{\omega_N \partial \hat{N}}{\omega_x \partial \hat{r}}) + \hat{\mathbf{A}_{1\mathbf{r}}} D\omega_N \hat{N} + A_2 D\omega_N \hat{N}\frac{\omega_N \partial \hat{N}}{\omega_x \partial \hat{r}}$ and $\hat{\mathbf{J}}_\mathbf{z} = (-D\frac{\omega_N \partial \hat{N}}{\omega_x \partial \hat{z}}) + \hat{\mathbf{A}_{1\mathbf{z}}} D\omega_N \hat{N} + A_2 D\omega_N \hat{N}\frac{\omega_N \partial \hat{N}}{\omega_x \partial \hat{z}}$, where $\hat{\mathbf{A}_{1\mathbf{r}}} = \omega_\mathbf{j}\frac{eZ^*\rho\hat{\mathbf{j}_\mathbf{r}}}{kT}$, $\hat{\mathbf{A}_{1\mathbf{z}}} = \omega_\mathbf{j}\frac{eZ^*\rho\hat{\mathbf{j}_\mathbf{z}}}{kT}$, $\omega_\mathbf{j} = \mathbf{j}_\mathbf{r}/\hat{\mathbf{j}_\mathbf{r}} = \mathbf{j}_\mathbf{z}/\hat{\mathbf{j}_\mathbf{z}} = 1 \times 10^{10}$, $\hat{\mathbf{j}_\mathbf{r}}$, $\hat{\mathbf{j}_\mathbf{z}}$ are the scaled current densities. To obtain the current density $\hat{\mathbf{j}_\mathbf{r}}$ and $\hat{\mathbf{j}_\mathbf{z}}$, we let $\omega_V = V/\hat{V} = 1 \times 10^{-2}$ and $\hat{V}$ are the scaled electric potential. The scaled Equation (9) is $-\frac{1}{\rho}(\frac{\omega_V \partial^2 \hat{V}}{\omega_x^2 \partial \hat{r}^2} + \frac{\omega_V \partial^2 \hat{V}}{\omega_x^2 \partial \hat{z}^2} + \frac{1}{\omega_x \hat{r}}\frac{\omega_V \partial \hat{V}}{\omega_x \partial \hat{r}}) = 0$. Then,

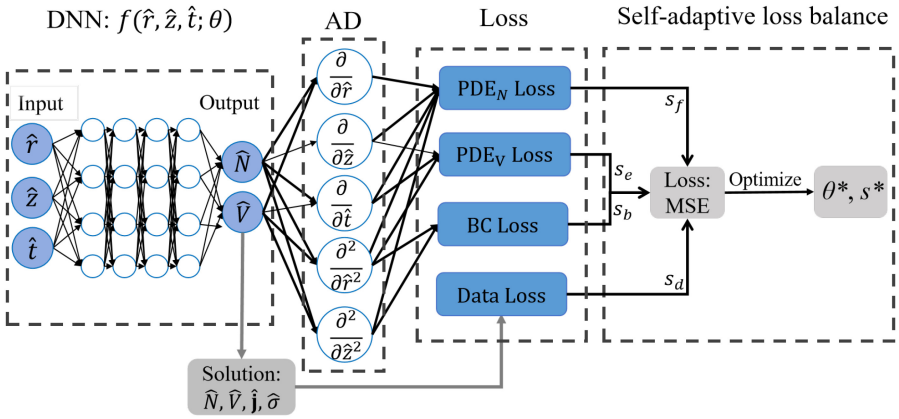


Fig. 5. The Framework of the proposed neural network, where AD represents automatic differentiation.

Equation (6) can be rewritten as follows:

$$
\begin{aligned}
&\frac{\partial\hat{N}}{\partial\hat{t}}+\frac{\omega_t}{\omega_x\omega_N}[\frac{\partial\hat{\mathbf{J}}_\mathbf{r}}{\partial\hat{r}}+\frac{\partial\hat{\mathbf{J}}_\mathbf{z}}{\partial\hat{z}}+\frac{\hat{\mathbf{J}}_\mathbf{r}}{\hat{r}}]=0,\\
&\hat{\mathbf{J}}_\mathbf{r}=\frac{\omega_N}{\omega_x}(-D\frac{\partial\hat{N}}{\partial\hat{r}})+\omega_N\hat{\mathbf{A}_{1\mathbf{r}}}D\hat{N}+\frac{\omega_N^2}{\omega_x}A_2D\hat{N}\frac{\partial\hat{N}}{\partial\hat{r}},\\
&\hat{\mathbf{J}}_\mathbf{z}=\frac{\omega_N}{\omega_x}(-D\frac{\partial\hat{N}}{\partial\hat{z}})+\omega_N\hat{\mathbf{A}_{1\mathbf{z}}}D\hat{N}+\frac{\omega_N^2}{\omega_x}A_2D\hat{N}\frac{\partial\hat{N}}{\partial\hat{z}},\\
&\hat{\mathbf{A}_{1\mathbf{r}}}=\omega_\mathbf{j}\frac{eZ^*\rho\hat{\mathbf{j}_\mathbf{r}}}{kT},\hat{\mathbf{A}_{1\mathbf{z}}}=\omega_\mathbf{j}\frac{eZ^*\rho\hat{\mathbf{j}_\mathbf{z}}}{kT},\\
&\frac{\omega_V}{\omega_x^2}(-\frac{1}{\rho})(\frac{\partial^2\hat{V}}{\partial\hat{r}^2}+\frac{\partial^2\hat{V}}{\partial\hat{z}^2}+\frac{1}{\hat{r}}\frac{\partial\hat{V}}{\partial\hat{r}})=0.
\end{aligned}
\tag{19}
$$

The scaling scheme is employed to normalize the inputs and coefficients of loss function. It is emphasized that this scaling scheme is also applied in BCs and ICs. Using these scaling factors $\omega_{\boldsymbol{x}},\omega_t,\omega_N,\omega_V,\omega_\mathbf{j},\omega_\sigma$, we can adjust the inputs for subsequent training process, thereby enhancing the training efficiency and optimization accuracy of the proposed network. Additionally, the prediction of atomic concentration N, electric potential V, current density $\mathbf{j}$ and back flow stress σ should be restored to the original magnitude by the scaling factors ω_N, ω_V, $\omega_\mathbf{j}$, and ω_σ, respectively.

3.5 The Proposed Neural Network

In this section, we propose a physics-informed learning structure based on the modified self-adaptive loss balanced method for transient electromigration analysis of TSV. The framework of the proposed neural network is shown in Figure 5. We first construct a **Deep Neural Network (DNN)** $f(\hat{r},\hat{z},\hat{t};\theta)$ to define the solution $\hat{\mathbf{u}}(\hat{r},\hat{z},\hat{t};\theta)$ of the fundamental PDEs governing the EM process in TSV. The inputs of the DNN model consist of the scaled spatial coordinates $(\hat{r},\hat{z})$ and the scaled temporal coordinate $\hat{t}$. The outputs of this network $\hat{\mathbf{u}}(\hat{r},\hat{z},\hat{t};\theta)$ include atomic concentration $\hat{N}$ and electrical potential $\hat{V}$. It should be noted that atomic concentration $\hat{N}$ and electric potential $\hat{V}$ are directly outputted by the neural network, while current density $\hat{\mathbf{j}}$ and back flow stress $\hat{\sigma}$ are indirect predictions. This is because the prediction of current density $\hat{\mathbf{j}}$ and back flow stress $\hat{\sigma}$ can be obtained from the transformation of the network outputs based on $\hat{\mathbf{j}}=\sqrt{(\frac{\partial\hat{V}}{\partial\hat{r}})^2+(\frac{\partial\hat{V}}{\partial\hat{z}})^2}$ and

$\hat{\sigma} = B\frac{\hat{N}-\hat{N}_0}{\hat{N}_0}$, in which $\hat{N}_0 = N_0/\omega_N$. We then implement automatic differentiation and compute the BC residual of Equation (7) and PDE residuals of Equations (6) and (9). The corresponding loss terms of PDE residuals, boundary conditions and data, $\mathcal{L}_f, \mathcal{L}_e, \mathcal{L}_{bc}$, and $\mathcal{L}_{data}$, can be presented as the following form:

$$
\begin{aligned}
\mathcal{L}_f &= \frac{1}{K}\sum_{i=1}^{K}\left|\frac{\partial\hat{N}_f^i}{\partial\hat{t}_f^i} - D\left(\frac{\partial^2\hat{N}_f^i}{\partial\hat{r}_f^{i\,2}} + \frac{\partial^2\hat{N}_f^i}{\partial\hat{z}_f^{i\,2}} + \frac{1}{\hat{r}_f^i}\frac{\partial\hat{N}_f^i}{\partial\hat{r}_f^i}\right) + \mathbf{A_{1r}}D\left(\frac{\partial\hat{N}_f^i}{\partial\hat{r}_f^i} + \frac{\hat{N}_f^i}{\hat{r}_f^i}\right) + \mathbf{A_{1z}}D\frac{\partial\hat{N}_f^i}{\partial\hat{z}_f^i}\right.\\
&\quad \left. + A_2D\left(\hat{N}_f^i\frac{\partial^2\hat{N}_f^i}{\partial\hat{r}_f^{i\,2}} + \hat{N}_f^i\frac{\partial^2\hat{N}_f^i}{\partial\hat{z}_f^{i\,2}} + \frac{\hat{N}_f^i}{\hat{r}_f^i}\frac{\partial\hat{N}_f^i}{\partial\hat{r}_f^i} + \left(\frac{\partial\hat{N}_f^i}{\partial\hat{r}_f^i}\right)^2 + \left(\frac{\partial\hat{N}_f^i}{\partial\hat{z}_f^i}\right)^2\right)\right|^2,\\
\mathcal{L}_{bc} &= \frac{1}{K_{zbc}}\sum_{i=1}^{K_{zbc}}\left(\left|\mathbf{A_{1z}}D\hat{N}_{zbc}^i - D\frac{\partial\hat{N}_{zbc}^i}{\partial\hat{z}_{zbc}^i} + A_2D\hat{N}_{zbc}^i\frac{\partial\hat{N}_{zbc}^i}{\partial\hat{z}_{zbc}^i}\right|^2 + \left|\hat{V}_{zbc}^i - V_g^i\right|^2\right)\\
&\quad + \frac{1}{K_{rbc}}\sum_{i=1}^{K_{rbc}}\left(\left|\mathbf{A_{1r}}D\hat{N}_{rbc}^i - D\frac{\partial\hat{N}_{rbc}^i}{\partial\hat{r}_{rbc}^i} + A_2D\hat{N}_{rbc}^i\frac{\partial\hat{N}_{rbc}^i}{\partial\hat{r}_{rbc}^i}\right|^2 + \left|\frac{1}{\rho}\frac{\partial\hat{V}_{rbc}^i}{\partial\hat{r}_{rbc}^i}\right|^2\right),\\
\mathcal{L}_e &= \frac{1}{K}\sum_{i=1}^{K}\left|\frac{\partial^2\hat{V}_e^i}{\partial\hat{r}_e^{i\,2}} + \frac{\partial^2\hat{V}_e^i}{\partial\hat{z}_e^{i\,2}} + \frac{1}{\hat{r}_e^i}\frac{\partial\hat{V}_e^i}{\partial\hat{r}_e^i}\right|^2, \quad \mathcal{L}_{data} = \frac{1}{K}\sum_{i=1}^{K}\left(\left|\hat{N}_{data}^i - N_g^i\right|^2 + \left|\hat{V}_{data}^i - V_g^i\right|^2\right).
\end{aligned}
\tag{20}
$$

The first loss term, denoted as $\mathcal{L}_f$, represents the residuals of PDE evaluated at a randomly selected set of training data points $\{\hat{r}_f^i, \hat{z}_f^i, \hat{t}_f^i, \hat{N}_f^i\}_{i=1}^K$ within the domain. The second loss term, $\mathcal{L}_{bc}$, comprises two components: $\mathcal{L}_{zbc}$ and $\mathcal{L}_{rbc}$, and it employs a designated set of training data to enforce boundary conditions. The third loss term, $\mathcal{L}_e$, ensures that the training points $\{\hat{r}_e^i, \hat{z}_e^i, \hat{t}_e^i, \hat{V}_e^i\}_{i=1}^K$ satisfy Equation (9). Lastly, the fourth loss term, $\mathcal{L}_{data}$, quantifies the discrepancy between the ground truth $\{N_g^i, V_g^i\}_{i=1}^K$ and the neural network's approximation at the sample data points $\{\hat{r}_{data}^i, \hat{z}_{data}^i, \hat{t}_{data}^i, \hat{N}_{data}^i, \hat{V}_{data}^i\}_{i=1}^K$. Here, $\hat{\mathbf{A}_{1\mathbf{r}}} = \omega_{\mathbf{j}}\frac{eZ^*\rho\hat{\mathbf{j}_r}}{kT}$, $\hat{\mathbf{A}_{1\mathbf{z}}} = \omega_{\mathbf{j}}\frac{eZ^*\rho\hat{\mathbf{j}_z}}{kT}$, $A_2 = \frac{\Omega DB}{kTN_0}$, and $\hat{\mathbf{j}_r} = -\frac{1}{\rho}\frac{\omega_V}{\omega_x}\frac{\partial\hat{V}}{\partial\hat{r}}$, $\hat{\mathbf{j}_z} = -\frac{1}{\rho}\frac{\omega_V}{\omega_x}\frac{\partial\hat{V}}{\partial\hat{z}}$, and $K, K_{zbc}, K_{rbc}, K_{data}$ denote the total number of points in their respective sets, respectively. It is essential to underline that the physical quantities and the related coefficients differ by a few orders of magnitude as a result of the aforementioned scaling scheme in the proposed neural network for transient EM modeling in TSV.

The uncertainty parameter $\varepsilon^* = [\varepsilon_f, \varepsilon_{data}, \varepsilon_e, \varepsilon_{bc}]$ is initialized after constructing the neural network $f(\hat{r}, \hat{z}, \hat{t}; \theta)$. To stabilize the training process, we utilize log variance s^* to substitute the adaptive variables ε^*. We then compute the initialized adaptive weight s^* based on the values of uncertainty parameter ε^* and combine the multiplication of different loss terms and corresponding adaptive weight s^*. Thus, Gaussian probabilistic model is constructed for EM modeling and simulation of TSV. The transformed loss function is given by

$$
MSE = \frac{1}{2}\left(\exp(-s_f)\mathcal{L}_f + \exp(-s_{bc})\mathcal{L}_{bc} + \exp(-s_e)\mathcal{L}_e + \exp(-s_{data})\mathcal{L}_{data} + s_f + s_{data} + s_e + s_{bc}\right), \tag{21}
$$

in which $s_f = \log\varepsilon_f^2$, $s_{data} = \log\varepsilon_{data}^2$, $s_e = \log\varepsilon_e^2$, $s_{bc} = \log\varepsilon_{bc}^2$. Finally, the Adam optimization algorithm and L-BFGS optimization are utilized to find the optimal model parameters θ^* and adaptive weight s^* by minimizing the overall loss function Equation (21). At each iteration, we first define the overall loss function with temporal self-adaptive weights based on maximum likelihood estimation and then adjust the adaptive weights utilizing Adam optimizer or L-BFGS optimizer to meet the constraints of PDE, BCs, and ICs in EM analysis of TSV. Lastly, we update the model parameters θ^* and adaptive weight s^* for subsequent optimization. The main difference

ALGORITHM 1: The Proposed Neural Network Framework for Transient Modeling and Simulation of Electromigration in TSV.

Input: The scaled spatial coordinates and temporal coordinate $\hat{r}, \hat{z}, \hat{t}$;
Output: The scaled atomic concentration $\hat{N}$ and electrical potential $\hat{V}$;
Step 1: Construct a DNN $f(\hat{r}, \hat{z}, \hat{t}; \theta)$ to define the solution of PDEs.
Step 2: Compute the residuals of PDEs and BCs by automatic differentiation and define loss terms.
Step 3: Initialize uncertainty parameters $\{\varepsilon_f, \varepsilon_e, \varepsilon_{bc}, \varepsilon_{data}\}$.
Step 4: Compute the corresponding initialized adaptive weights $\{s_f, s_e, s_{bc}, s_{data}\}$ and combine loss terms with corresponding adaptive weights.
Step 5: Use $K_1 + K_2$ gradient decent iterations to adjust the model parameters and self-adaptive weights as:
for $m = 1 \rightarrow K_1$ **do**
(a) Define the overall loss function based on maximum likelihood estimation.
(b) Adjust the model parameters and self-adaptive weights via Adam optimizer to meet the constraints.
(c) Update the model parameters and weights.
end for
for $n = 1 \rightarrow K_2$ **do**
(a) Define the overall loss function based on maximum likelihood estimation.
(b) Adjust the model parameters and self-adaptive weights via L-BFGS optimizer to meet the constraints.
(c) Update the model parameters and weights.
end for

between the proposed method and the PINN-based methods [10, 23, 35, 37] is the application of the modified self-adaptive loss balanced method. In the PINN-based methods, the loss function is a simple sum of various loss terms and the coefficients of different loss terms are constant. However, in our proposed network, the loss function is derived by multiplying various loss terms by dynamic coefficients and subsequently summing these resultant products. The dynamic coefficients are defined by the constructed Gaussian probabilistic model and adjusted in the training process automatically to achieve the optimized coefficients and therefore enhance the network performance in Algorithm 1.

4 Results and Discussions

In this section, we use PyTorch to implement neural networks. To ensure a fair comparison of various models' performances, DNN, the baseline PINN, PFNN, PINN-LA, PINN-LA-2, LBPINN, and the proposed model all employ identical neural network architectures. First, we present experimental results on EM modeling analysis of TSV, including the evolution of atomic concentration, electrical potential, back flow stress and electric current density. Subsequently, we also verify the effectiveness of the proposed physics-informed learning method and compare the results with the finite element method, DNN, PINN, PFNN, PINN-LA, PINN-LA-2, LBPINN in terms of accuracy and speed. We further study the performance of the proposed neural network by varying the initial adaptive weights, numbers of hidden layers and neurons per layer.

4.1 Experiment Setup

All programs are implemented in Python 3.9.15 with PyTorch 1.13.1 and are executed on a Linux server equipped with 2.20-GHz Xeon processors and an NVIDIA RTX A6000. The ground truth data is obtained from the finite element simulations performed using COMSOL Multiphysics software. The utilized parameters for modeling are listed in Table 1. During the optimization, Adam optimization is initially performed with learning rate 0.001, followed by L-BFGS for the subsequent training stage. For simplicity, we choose tanh as the activation function.

Table 1. The Utilized Parameters for Modeling

Parameter	Value	Description
N_0	1.53e46 Atoms/m^3	Atomic concentration
E_A	2.1 eV	Activation energy
k	1.38e-23 J/K	Boltzmann constant
e	1.6e-19 C	Electric charge
Z^*	-4	Effective valence charge
ρ_0	1.8e-8 Ω·m	Electrical resistivity
D_0	1e-7 m^2/s	Self-diffusion coefficient
Q^*	1.387e-20	Heat of transport
Ω	1.182e-29 m^3	Atomic volume
B	5e7 Pa	Back flow stress modular

4.2 Accuracy Analysis

The accuracy between the proposed method and the finite element method is measured by the relative L2 error. We denote the exact value as $u(r_i, z_i, t_i)$ obtained from the finite element method and its approximation as $\hat{u}(r_i, z_i, t_i)$ calculated by the neural network. For the data points $\{r_i, z_i, t_i\}_{i=1}^M$, the relative L2 error is defined as follows:

$$\text{L2error} = \frac{\sqrt{\sum_{i=1}^M |\hat{u}(r_i, z_i, t_i) - u(r_i, z_i, t_i)|^2}}{\sqrt{\sum_{i=1}^M |u(r_i, z_i, t_i)|^2}}. \tag{22}$$

We first examine the prediction accuracy of the proposed physics-informed neural network for calculating atomic accumulation N, electrical potential V, current density $\mathbf{j}$ and back flow stress σ. As can be seen from Figure 6(a), the bottom has depleted atoms ($\Delta N/N < 0$), and the top has accumulated atoms ($\Delta N/N > 0$) because copper atoms move from the bottom to the top as a result of momentum exchange following electron collisions. The result obtained from the proposed method fits well with the solution obtained from FEM, with L2error less than 0.3%. It can be observed from Figure 6(b) that the distribution of electrical potential in TSV is uniform and the configuration is that the top electrodes of TSV serve as the electrical input and the bottom electrodes of TSV are ground. Compared with the finite element method, the maximum relative L2error of the proposed model is about 0.6%. The experiment results of the current density, as presented in Figure 6(c), indicate that the current density near the bottom changes rapidly as atom depletes at the bottom and then the resistance of the bottom changes accordingly. Figure 6(d) shows the simulation results of the back flow stress at $t = 5 \times 10^6$s and indicates that compressive stress occurs at the top and tensile stress at the bottom, featuring a considerable stress gradient throughout TSV. Additionally, we find that the relative error of electric current density is larger than the other 3 quantities from Figure 6. The neural network directly outputs predictions for the atomic concentration N and electric potential V. Meanwhile, indirect predictions for the current density $\mathbf{j}$ and back flow stress σ are derived. Specifically, $\mathbf{j}$ is determined by $\mathbf{j} = \sqrt{(\frac{\partial V}{\partial r})^2 + (\frac{\partial V}{\partial z})^2}$, and σ is calculated using $\sigma = B\frac{N-N_0}{N_0}$, where V and N represent the predicted electric potential and atomic concentration, respectively. Possibly due to this reason, the relative errors for current density and back flow stress can be greater than those for the directly predicted quantities. Notably, since current density is computed from the gradients of the electric potential, its relative error may exceed that of the back flow stress.

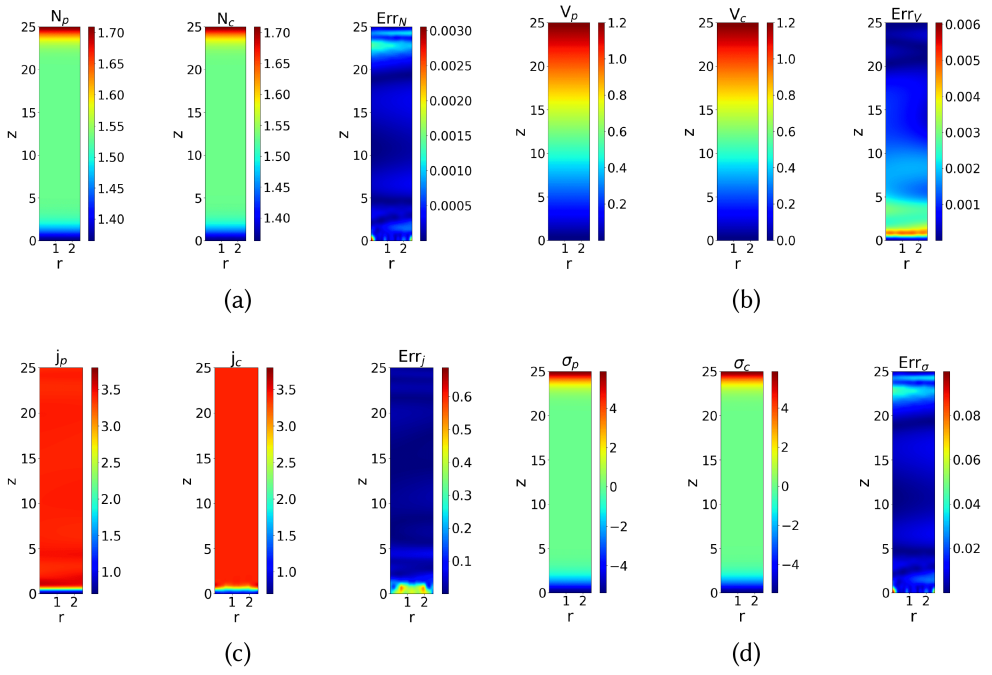


Fig. 6. The scaled atomic concentration, electric potential, electric current density and back flow stress at $t = 5 \times 10^6 s$. (a) The scaled atomic concentration at $t = 5 \times 10^6 s$: $\mathrm{N_p}$ and $\mathrm{N_c}$ are obtained by the proposed method and the finite element method, respectively, and Err_N is the absolute error between $\mathrm{N_p}$ and $\mathrm{N_c}$. (b) The scaled electric potential at $t = 5 \times 10^6 s$: $\mathrm{V_p}$ and $\mathrm{V_c}$ are obtained by the proposed method and the finite element method, respectively, and Err_V is the absolute error between $\mathrm{V_p}$ and $\mathrm{V_c}$. (c) The scaled electric current density at $t = 5 \times 10^6 s$: $\mathrm{j_p}$ and $\mathrm{j_c}$ are obtained by the proposed method and the finite element method, respectively, and Err_j is the absolute error between $\mathrm{j_p}$ and $\mathrm{j_c}$. (d) The scaled back flow stress at $t = 5 \times 10^6 s$: σ_p and σ_c are obtained by the proposed method and the finite element method, respectively, and Err_σ is the absolute error between σ_p and σ_c.

Then, we present the results of dynamic iterations of uncertainty parameters $\{\varepsilon_f, \varepsilon_e, \varepsilon_{bc}, \varepsilon_{data}\}$ and self-adaptive weights $\{s_f, s_e, s_{bc}, s_{data}\}$ in Figure 7(a) and (b). As shown in Figure 7(a), the changes of uncertainty parameters ε_f, ε_e, and ε_{bc} are relatively minor, transitioning from 2000 to 1999, from 100 to 99, and from 200 to 199, respectively. However, the change in ε_{data} is significant, decreasing from 2 to nearly 0. It can be seen from Figure 7(b) that the alterations in self-adaptive weight s_f are relatively negligible, whereas the change in s_{data} is substantial, diminishing from 0.6 to a value close to -10. The fluctuations in s_e and s_{bc} are minor, approximately from 9.21 to 9.19 and from 10.595 to 10.585, respectively.

We test the absolute error of the proposed method for predicting atomic concentration N, electric potential V, current density $\mathbf{j}$, and back flow stress σ at the boundaries $r = R_{tsv}$, $z = 0$, and $z = h$. From Figures 8–10, it is observed that the absolute error of electric potential V is less than the other physical variables. The absolute error at the boundary condition $r = R_{tsv}$ is presented in Figure 8, illustrating that the absolute errors of atomic concentration N, current density $\mathbf{j}$ and back flow stress σ are less than errors at the boundaries $z = 0$ and $z = h$. From Figure 9, it can be observed that the absolute error of electric potential V at the boundary condition $z = h$ is smallest compared to errors at the other boundaries and the maximum absolute error of V is approximately 0.00015.

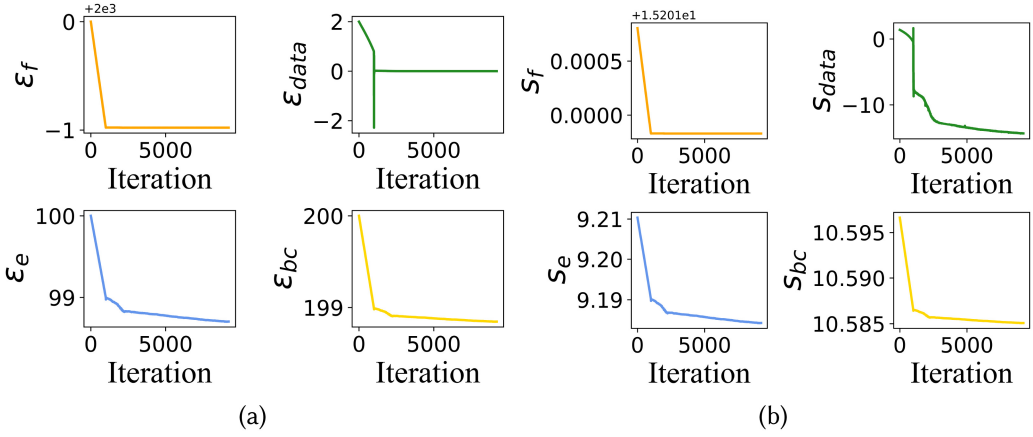


Fig. 7. The uncertainty parameters and self-adaptive weights over iteration in the training process. (a) The uncertainty parameters $\{\varepsilon_f, \varepsilon_s, \varepsilon_{data}, \varepsilon_{bc}\}$ over iteration in the training process of the proposed physics-informed learning model. (b) The self-adaptive weights $\{s_1, s_2, s_3, s_4\}$ over iteration in the training process of the proposed physics-informed learning model, in which $s_f = \log \varepsilon_f^2$, $s_{data} = \log \varepsilon_{data}^2$, $s_e = \log \varepsilon_e^2$, $s_{bc} = \log \varepsilon_{bc}^2$.

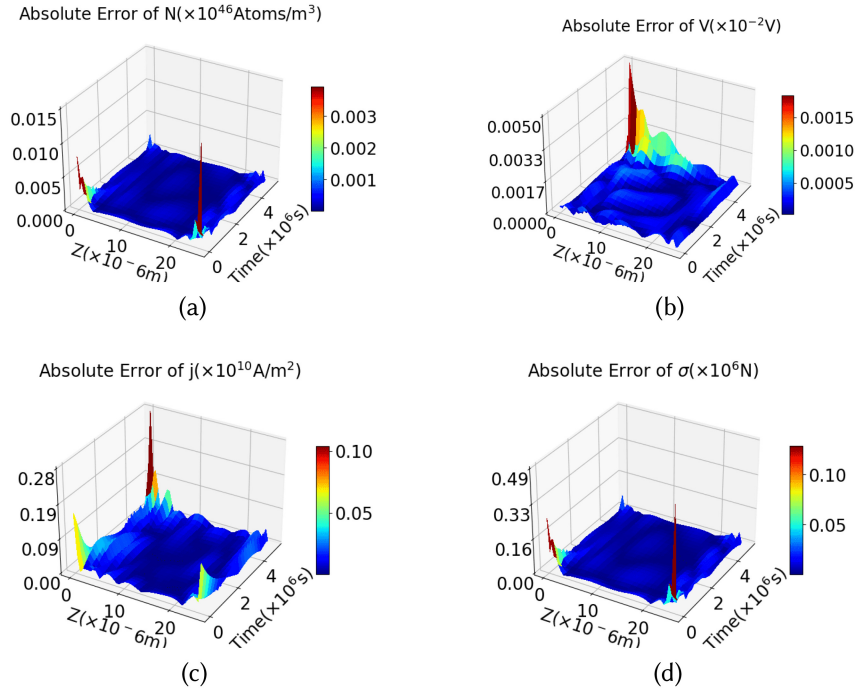


Fig. 8. Absolute error between the proposed model and FEM at the boundary $r = R_{tsv}$ in the time range $0-5\times10^6 s$. (a) Absolute error of atomic concentration N. (b) Absolute error of electric potential V. (c) Absolute error of current density $\mathbf{j}$. (d) Absolute error of back flow stress σ.

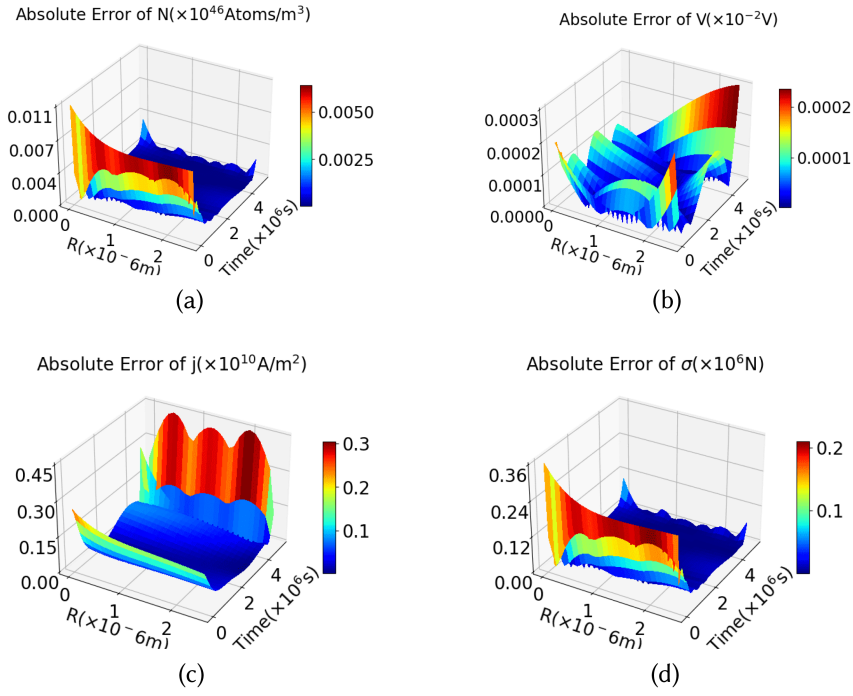


Fig. 9. Absolute error between the proposed model and FEM at the boundary $z = 0$ in the time range $0 - 5 \times 10^6 s$. (a) Absolute error of atomic concentration N. (b) Absolute error of electric potential V. (c) Absolute error of current density $\mathbf{j}$. (d) Absolute error of back flow stress σ.

4.3 Performance Analysis

To examine the effectiveness of our method, we compare the proposed method with DNN, PINN [10], PFNN [37], PINN-LA[23], PINN-LA-2[23], and LBPINN[35]. It can be seen from Table 2 that the proposed model achieves higher accuracy in the prediction of atomic concentration, electric potential, current density and back flow stress compared with other competing methods. The corresponding MTTF predictions are compared in Table 2. The MTTF value obtained using COMSOL is 4.31×10^6s. Upon examining Table 2, it becomes evident that the MTTF prediction of the proposed method aligns precisely with this value at 4.30×10^6s, indicating a superior level of accuracy compared to other methods. Figure 11(a) presents various data loss terms of the proposed method and other competing methods over iteration, illustrating the effectiveness of the proposed method as the data loss of the proposed method is minimal. The loss term of PDE residual $\mathcal{L}_f$ and $\mathcal{L}_e$, boundary condition $\mathcal{L}_{bc}$ and data $\mathcal{L}_{data}$ over iteration during training process are plotted in Figure 11(b), in which $\mathcal{L}_f$ and $\mathcal{L}_e$ in Equation (20) represent the PDE losses described in Equation (19).

To demonstrate the computational saving and satisfactory accuracy of our proposed model, we carry out experiments to calculate the inference time and training time of the proposed method compared with the runtime of FEM. It can be seen from Table 3 that the running speed of the proposed PINN model is much faster than FEM and the acceleration ratio can reach to about 15.84. It is noted that acceleration ratio improves when the mesh gets denser and the minimum acceleration ratio can still reach to about 4.68 when the mesh density gets smaller. It is also worthy to mention that the mean squared error of atomic concentration N decreases when the mesh density gets larger, which shows the potential of the propose PINN model for EM analysis of TSV. We also

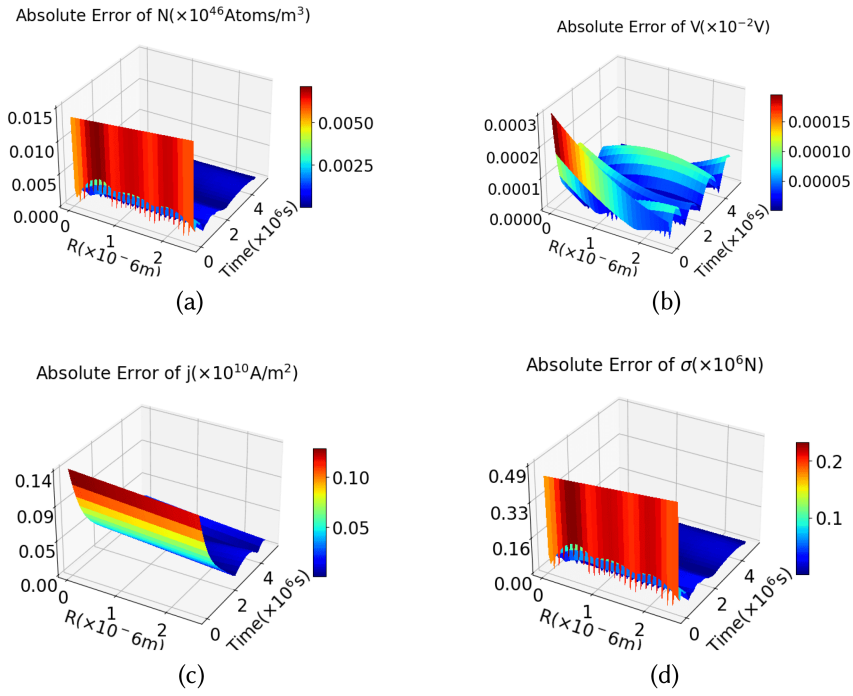


Fig. 10. Absolute error between the proposed model and FEM at the boundary $z = h$ in the time range $0-5\times10^6 s$. (a) Absolute error of atomic concentration N. (b) Absolute error of electric potential V. (c) Absolute error of current density $\mathbf{j}$. (d) Absolute error of back flow stress σ.

Table 2. Relative L2 Errors (%) and MTTF ($\times10^6$s) with Various Methods during the Time Range $0 - 5 \times 10^6$s

Terms	DNN	PINN [10]	PFNN [37]	PINN-LA [23]	PINN-LA-2 [23]	LBPINN [35]	Proposed
N	0.152	0.163	0.126	0.136	0.119	0.050	**0.0325**
V	0.299	0.275	0.119	0.101	0.126	0.068	**0.0559**
j	9.10	5.59	2.17	2.13	2.91	1.02	**0.811**
σ	7.84	8.40	6.50	7.02	6.12	2.59	**1.673**
$MTTF$	4.00	4.04	4.01	4.15	4.18	4.27	**4.30**

conduct experiments to obtain the training time between DNN without physical information and the proposed method for the same number of iterations. We find that the proposed method can obtain higher accuracy compared with DNN although the DNN demonstrates a shorter training time for the same number of iterations.

For verifying the robustness of the proposed model, we set different initial settings of uncertainty parameters $\{\varepsilon_f, \varepsilon_e, \varepsilon_{bc}, \varepsilon_{data}\}$ and all other settings stay same. Experimental results shown in Table 4 illustrate that the relative L2 errors of atomic concentration N, electric potential V, current density $\mathbf{j}$ and back flow stress σ are similar while the final settings of adaptive weights are different. This indicates that our model achieves high robustness under different values of initial weights. To test the performance of the proposed model, we implement experiments under different numbers of layers and neurons per layer. It can be seen from Table 5 that the network with more hidden layers achieve higher accuracy approximately. The relative L2 error of the proposed

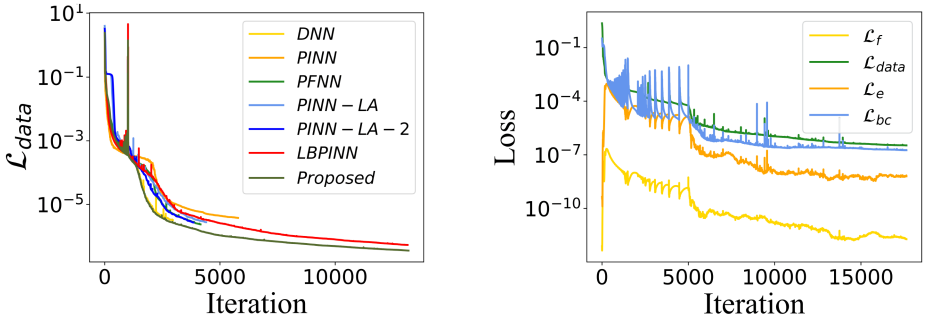


Fig. 11. The loss values over iteration in the training process. (a) The data loss values $\mathcal{L}_{data}$ of the proposed method and other competing methods over iteration. (b) The loss values $\{\mathcal{L}_f, \mathcal{L}_{data}, \mathcal{L}_e, \mathcal{L}_{bc}\}$ of the proposed physics-informed learning model over iteration.

Table 3. The Training and Inference Time of the Proposed PINN Model and the Computation Time of FEM

Mesh Density	FEM (s)	Proposed Model				Acceleration Ratio
		Inference (s)	Training (s)	Total (s)	Error	
40000	25	0.0073	5.3338	5.3411	1.74×10^{-4}	4.68
80000	51	0.0118	6.6198	6.6316	1.28×10^{-4}	7.69
160000	108	0.0243	10.6565	10.6808	9.41×10^{-5}	10.13
320000	229	0.0418	17.1358	17.1776	6.91×10^{-5}	13.33
640000	493	0.0829	31.0378	31.1207	4.63×10^{-5}	15.84

Table 4. Relative L2 Errors (%) of Physical Field Solutions and Final Settings $[\varepsilon_f, \varepsilon_{data}, \varepsilon_e, \varepsilon_{bc}]$ During the Time Range $0 - 5 \times 10^6$s Obtained by the Proposed Method under Different Initial Settings $[\varepsilon_f, \varepsilon_{data}, \varepsilon_e, \varepsilon_{bc}]$

Initial settings	Final settings	Error_N	Error_V	Error_j	Error_σ
[20, 2, 20, 20]	$[16.22421, -6.51 \times 10^{-4}, 16.22417, 16.22421]$	0.033	0.059	1.190	1.683
[200, 2, 20, 20]	$[198.825, 6.68 \times 10^{-4}, 17.20539, 17.20544]$	0.035	0.055	0.736	1.828
[200, 2, 100, 199]	$[198.836, 6.37 \times 10^{-4}, 98.681, 197.83]$	0.036	0.055	1.128	1.871
[2000, 2, 20, 20]	$[1999.0208, -6.125 \times 10^{-4}, 16.9230, 16.9231]$	0.031	0.056	0.858	1.597
[2000, 2, 100, 200]	$[1999.0224, 6.69 \times 10^{-4}, 98.726, 198.86]$	0.035	0.052	0.914	1.822

Table 5. Relative L2 Errors (%) of Atomic Concentration N for Different Numbers of Hidden Layers and Neurons Per Layer

layers \ neurons	10	20	30	40	50
4	0.152	0.046	0.045	0.035	0.037
6	0.113	0.042	0.043	0.037	0.033
8	0.080	0.038	0.037	0.037	0.031
10	0.059	0.033	0.035	0.038	0.033

model with 10 neurons per layer is more than other neurons per layer. It should be noted that the relative L2 errors of 20, 30, 40, and 50 neurons per layer are similar. Therefore, we choose 4 hidden layers with 40 neurons per layer for training to accelerate the speed of optimization.

5 Conclusion

In this article, we propose a physics-informed learning method for transient EM modeling in TSV. The proposed method iteratively updates the resistivity dependent on the temporal atomic concentration, models the electric current and stress gradient, and computes the atomic concentration for subsequent training. To further enhance the performance of the network, we develop a modified self-adaptive loss balanced method to automatically change the weights of different loss terms. This method constructs Gaussian probabilistic model to define the overall loss function based on maximum likelihood inference, which can improve the approximation capability of the proposed physics-informed learning network. The approximate weights are obtained after iterations during train process utilizing the positive relationship of adaptive weights and corresponding loss terms. Unlike traditional numerical methods, the proposed model no longer relies on grid meshing and temporal discretization, which can improve efficiency significantly. To the best of our knowledge, this is the $first\ study$ that utilize machine learning-based methods for multi-physical electromigration analysis of TSV. Compared with the finite element method, the proposed physics-informed learning method can lead to an acceleration with less than 0.1% relative error. Experimental results also demonstrate that the proposed model shows promising accuracy over competing learning-based methods.

References

[1] Seungwon Kim, Seokhyeong Kang, Ki Jin Han, and Youngmin Kim. 2016. Novel adaptive power-gating strategy and tapered TSV structure in multilayer 3D IC. *ACM Transactions on Design Automation of Electronic Systems* 21, 3 (2016), 1–19.

[2] Tak-Yung Kim and Taewhan Kim. 2011. Clock tree synthesis for TSV-based 3D IC designs. *ACM Transactions on Design Automation of Electronic Systems* 16, 4 (2011), 1–21.

[3] Yuanqing Cheng, Aida Todri-Sanial, Jianlei Yang, and Weisheng Zhao. 2016. Alleviating through-silicon-via electromigration for 3-D integrated circuits taking advantage of self-healing effect. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 24, 11 (2016), 3310–3322.

[4] Zhaohui Chen, Zhicheng Lv, XueFang Wang, Yong Liu, and Sheng Liu. 2010. Modeling of electromigration of the through silicon via interconnects. In *Proceedings of the International Conference on Electronic Packaging Technology and High Density Packaging*. 1221–1225.

[5] Yung Chuen Tan, Cher Ming Tan, XW Zhang, Tai Chong Chai, and DQ Yu. 2010. Electromigration performance of Through Silicon Via (TSV)–A modeling approach. *Microelectronics Reliability* 50, 9-11 (2010), 1336–1340.

[6] Jiwoo Pak, Mohit Pathak, Sung Kyu Lim, and David Z Pan. 2011. Modeling of electromigration in through-silicon-via based 3D IC. In *Proceedings of the IEEE 61st Electronic Components and Technology Conference*. 1420–1427.

[7] Xin Zhao, Yang Wan, Michael Scheuermann, and Sung Kyu Lim. 2013. Transient modeling of TSV-wire electromigration and lifetime analysis of power distribution network for 3D ICs. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 363–370.

[8] Hao Zhou, Hengliang Zhu, Tao Cui, David Z. Pan, Dian Zhou, and Xuan Zeng. 2018. Thermal stress and reliability analysis of TSV-based 3-D ICs with a novel adaptive strategy finite element method. *IEEE Transactions on Very Large Scale Integration* 26, 7 (2018), 1312–1325.

[9] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. 2021. Physics-informed machine learning. *Nature Reviews Physics* 3, 6 (2021), 422–440.

[10] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 378, 2 (2019), 686–707.

[11] Tianshu Hou, Peining Zhen, Zhigang Ji, and Hai-Bao Chen. 2023. A deep learning framework for solving stress-based partial differential equations in electromigration analysis. *ACM Transactions on Design Automation of Electronic Systems* 28, 4 (2023), 1–20.

[12] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. 2018. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research* 18, 1 (2018), 1–43.
[13] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. 2021. DeepXDE: A deep learning library for solving differential equations. *SIAM Rev.iews* 63, 1 (2021), 208–228.
[14] Xinquan Huang, Tariq Alkhalifah, and Chao Song. 2021. A modified physics-informed neural network with positional encoding. In *Proceedings of the International Exposition and Annual Meeting*. SEG, D011S106R002.
[15] Xiaowei Jin, Shengze Cai, Hui Li, and George Em Karniadakis. 2021. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Computational Physics* 426, 2 (2021), 109951.
[16] Navid Zobeiry and Keith D. Humfeld. 2021. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Engineering Applications of Artificial Intelligence* 101, 5 (2021), 104232.
[17] Tianshu Hou, Ngai Wong, Quan Chen, Zhigang Ji, and Hai-Bao Chen. 2022. 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* 41, 12 (2022), 5501–5514.
[18] Wentian Jin, Liang Chen, Subed Lamichhane, Mohammadamir Kavousi, and Sheldon X.-D. Tan. 2022. HierPINN-EM: Fast learning-based electromigration analysis for multi-segment interconnects using hierarchical physics-informed neural network. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*. 1–9.
[19] Tianshu Hou, Peining Zhen, Ngai Wong, Quan Chen, Guoyong Shi, Shuqi Wang, and Hai-Bao Chen. 2022. Multilayer perceptron-based stress evolution analysis under dc current stressing for multisegment wires. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 42, 2 (2022), 544–557.
[20] Subed Lamichhane, Wentian Jin, Liang Chen, Mohammadamir Kavousi, and Sheldon X.-D. Tan. 2023. PostPINN-EM: Fast post-voiding electromigration analysis using two-stage physics-informed neural networks. In *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*. IEEE, 1–9.
[21] Levi McClenny and Ulisses Braga-Neto. 2023. Self-adaptive physics-informed neural networks. *Journal of Computational Physics* 474, 2 (2023), 111722.
[22] Jungeun Kim, Kookjin Lee, Dongeun Lee, Sheo Yon Jhin, and Noseong Park. 2021. DPM: A novel training method for physics-informed neural networks in extrapolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 8146–8154.
[23] Sifan Wang, Yujun Teng, and Paris Perdikaris. 2021. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing* 43, 5 (2021), A3055–A3081.
[24] Sifan Wang, Hanwen Wang, and Paris Perdikaris. 2021. 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* 384, 10 (2021), 113938.
[25] Sifan Wang, Xinling Yu, and Paris Perdikaris. 2022. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics* 449, 1 (2022), 110768.
[26] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the International Conference on Machine Learning*. PMLR, 794–803.
[27] A. Ali Heydari, Craig A. Thompson, and Asif Mehmood. 2019. Softadapt: Techniques for adaptive loss weighting of neural networks with multi-part loss functions. arXiv:1912.12355. Retrieved from https://doi.org/10.48550/arXiv.1912.12355
[28] Moongon Jung, Joydeep Mitra, David Z. Pan, and Sung Kyu Lim. 2014. TSV stress-aware full-chip mechanical reliability analysis and optimization for 3D IC. *Communications of the ACM* 57, 1 (2014), 107–115.
[29] Valeriy Sukharev, Armen Kteyan, and Ehrenfried Zschech. 2012. Physics-based models for EM and SM simulation in three-dimensional IC structures. *IEEE Transactions on Device and Materials Reliability* 12, 2 (2012), 272–284.
[30] Cheryl S. Selvanayagam, John H. Lau, Xiaowu Zhang, S. K. W. Seah, Kripesh Vaidyanathan, and T. C. Chai. 2009. Nonlinear thermal stress/strain analyses of copper filled TSV (through silicon via) and their flip-chip microbumps. *IEEE Transactions on Advanced Packaging* 32, 4 (2009), 720–728.
[31] Qiaosha Zou, Eren Kursun, and Yuan Xie. 2017. Thermomechanical stress-aware management for 3-D IC designs. *IEEE Transactions on Very Large Scale Integration Systems* 25, 9 (2017), 2678–2682.
[32] Sukanta Dey, Sukumar Nandi, and Gaurav Trivedi. 2020. Machine learning approach for fast electromigration aware aging prediction in incremental design of large scale on-chip power grid network. *ACM Transactions on Design Automation of Electronic Systems* 25, 5 (2020), 1–29.
[33] Shengcheng Wang and Mehdi B Tahoori. 2017. Electromigration-aware local-via allocation in power/ground TSVs of 3-D ICs. *IEEE Transactions on Very Large Scale Integration Systems* 25, 10 (2017), 2881–2892.

[34] Xin Zhao, Michael Scheuermann, and Sung Kyu Lim. 2012. Analysis of DC current crowding in through-silicon-vias and its impact on power integrity in 3D ICs. In *Proceedings of the 49th Annual Design Automation Conference*. 157–162.
[35] Zixue Xiang, Wei Peng, Xu Liu, and Wen Yao. 2022. Self-adaptive loss balanced Physics-informed neural networks. *Neurocomputing* 496, C (2022), 11–34.
[36] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7482–7491.
[37] Hailong Sheng and Chao Yang. 2021. PFNN: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries. *Journal of Computational Physics* 428, 3 (2021), 110085.

Received 2 July 2024; revised 10 October 2024; accepted 14 November 2024