EMGraph: Fast Learning-Based Electromigration Analysis for Multi-Segment Interconnect Using Graph Convolution Networks

Wentian Jin, Liang Chen, Sheriff Sadiqbatcha, Shaoyi Peng, and Sheldon X.-D. Tan
Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521
{wjin018, liangch, ssadi003, speng004, sheldont}@ucr.edu

***Abstract*—Electromigration (EM) becomes a major concern for VLSI circuits as the technology advances in the nanometer regime. With Korhonen equations, EM assessment for VLSI circuits remains challenged due to the increasing integrated density. VLSI multisegment interconnect trees can be naturally viewed as graphs. Based on this observation, we propose a new graph convolution network (GCN) model, which is called *EMGraph* considering both node and edge embedding features, to estimate the transient EM stress of interconnect trees. Compared with recently proposed generative adversarial network (GAN) based stress image-generation method, EMGraph model can learn more transferable knowledge to predict stress distributions on new graphs without retraining via inductive learning. Trained on the large dataset, the model shows less than 1.5% averaged error compared to the ground truth results and is orders of magnitude faster than both COMSOL and state-of-the-art method. It also achieves smaller model size, 4× accuracy and 14× speedup over the GAN-based method.**

***Index Terms*—Electromigration (EM), graph convolution network (GCN), multisegment interconnect, hydrostatic stress assessment**

I. Introduction

Electromigration (EM) is still the primary reliability concern for VLSI interconnect as the technology advances in the nanometer regime. As predicted by International Technology Roadmap for Semiconductors (ITRS), EM is projected to only get worse in future technology nodes [1]. EM-related aging and reliability will become worse for current 5nm and below technologies. As a result, it is crucial to ensure the reliability of the very large scale integration (VLSI) chips during their projected lifetimes.

It is well accepted that existing Black and Blech-based EM models [2], [3] are overly conservative and can only work for single wire segment [4], [5]. To mitigate those problems, recently many physics-based EM models and simulation techniques have been proposed [6]–[18]. These computational techniques primarily focus on finding a solution of Korhonen equations [19], which is the partial differential equations (PDEs) describing the hydrostatic stress evolution in the confined multi-segment interconnects subject to blocking materials boundary conditions. A number of conventional numerical and analytical methods are proposed to attempt to solve the PDEs efficiently and accurately [15], [17], [20]–[22]. Although the numerical methods, such as finite difference method [20], [21] and finite element method [15], can work for the complex interconnect structures and obtain EM stress accurately, they impose high computational cost due to discretization of space and time. Recently, semi-analytical method based on separation of variables method has been proposed [17], [22], which show promising performance in both accuracy and efficiency on general multi-segment interconnect. However, solving the Korhonen equation in particular and PDEs in general by traditional numerical methods still remains a big challenge due to the inherent limitation of those methods.

On the other hand, recent breakthrough from deep learning for cognitive tasks based on deep neural networks (DNN) bring new opportunities for solving the differential equations for many applications in electric design automation (EDA) field [23]. However, how to apply the deep learning techniques to solve nonlinear partial differential equations still remains in its infancy.

In this work, we propose to leverage graph neural networks (GNN) to solve the Korhonen equations for multi-segment interconnects for fast EM failure analysis. GNN can naturally present multi-segment interconnects, which can be viewed as graphs with node relationship presented by edges. GNN [24] is a more general data representation and learning framework for complex relationship beyond Euclidean space. Recently, numerous GNN models have been developed for learning various kinds of graph structures [25], [26] and in the EDA areas [27]–[31]. Our new contributions are as follows:

- We apply GNN to perform transient EM stress on multi-segment interconnect for the first time to the best of our knowledge. A graph dataset on EM assessment is created using COMSOL multiphysics. The input of the GNN model is edge features, such as length, width, current density, a graph structure and time. Its output is the stress on the edges. Then, we can estimate the hydrostatic stress in each segment wire at the given time.
- We design our own graph neural network (called *EMGraph*) to perform the node-edge regression task based on the popular GraphSage network. Compared with GAN based method, the proposed EMGraph model can learn more transferable knowledge to predict stress distributions on new graphs without retraining via inductive learning. We use EMGraph to predict EM stress on large and unseen designs with good accuracy and fast speed, which can not be achieved by recently proposed EM-GAN method [32] because of its size-fixed image limitation. In addition, the size of EMGraph model is much smaller than that of EM-GAN model.
- A novel graph convolution-decoder structure is employed in EMGraph model. Our model first processes the input graph using graph convolution. The resulting graph embedding features are then fed into node and edge decoders which convert latent features to stress outputs.
- Our evaluation results show that the EMGraph yields 1.5% averaged error compared to the ground truth results and is orders of magnitude faster than both COMSOL and state-of-the-art method. It also leads to better accuracy

This work is supported in part by NSF grants under No. CCF-1816361, in part by NSF grant under No. CCF-2007135 and No. OISE-1854276.

978-1-6654-3274-0/21/$31.00

and 14× speedup over the EM-GAN method on several interconnect trees benchmarks.

The paper is organized as follows: Section II reviews the physics-based EM model and its assessment techniques. Section III defines the input and output features, and graph construction. Based on the popular GraphSage network, we propose EMGraph in Section IV. Experimental results are presented in Section V. Finally, section VI concludes this paper.

II. Review of relevant work

A. DNN based approaches for solving PDEs

In order to solve the PDE (1) accurately, numerical methods [15], [20], [21], such as finite difference and finite element methods, are applied for EM assessment. However, it requires huge computational cost and is not scalable for modern chips. Therefore, an analytic based method, called separation of variables method, is employed to estimate the transient hydrostatic stress with eigenvalues, which suffers from computing eigenvalues slowly and determining the number of eigenvalues [17], [22].

Deep learning has revolutionized the machine learning fields with breakthoughs in many cognitive applications such as image, text, speech and graph recognition [23], [33]. inspired by the observations, neural networks are modified to solve the PDEs. There are several strategies to solving PDE by DNN based methods. One approach is to frame the PDE solving process into a nonlinear optimization process coded by DNN with the loss functions to enforce the physics laws represented by the PDE and boundary conditions. Recently proposed physics-informed neural networks [34], [35] or physics-constrainted neural networks [36], [37] methods represent this strategy. But most of the reported works only solve very small PDE with simple boundary conditions. Furthermore, it is unclear that such methods can deliver sufficient accuracy without any labels (unsupervised learning). On the other hand, the second approach uses supervised learning to build DNN models based on the measured or simulated label data. Recently a generative adversarial networks (GAN) based method, called *EM-GAN*, is proposed to perform a fast transient hydrostatic stress analysis by solving Korhonen equations [32]. It can achieve an order of magnitude speedup over the efficient analytic based EM solver with good accuracy. However, this method only works for a fixed region because its output is an fixed image, which restricts its application in the real chips. What is more, the image is not a natural tool to represent the multisegment interconnects as the region with large areas are filled with nothing.

To efficiently represent the multisegment interconnect structure, graph is more suitable to store the node and edge information of the interconnect trees. Kipf and Welling introduce a definition of convolution operation on a graph, which aggregates information into the node from its neighborhood nodes [25]. However, this method only does the task on a fixed graph because the input needs an adjacency matrix representing a graph. Once the graph is changed, the model needs to be trained again. To mitigate this problem, GraphSAGE network is proposed for inductive learning on graphs [26]. Unlike matrix factorization method proposed by Kipf and Welling, GraphSAGE only learns the local node features by aggregating the information from its neighborhood and can predict the features at unseen nodes, which means the model can predict the embedding features on new graphs without retraining. Also several works leveraging GNN have been proposed recently for solving various problems in EDA such as analog circuit clustering [27], estimation of device parameters in [28], chip power estimation in [38], 3D circuit partitioning [29], transistor sizing [30], analog IC placement [31]. Since GNN represents more general and natural relationship among different design objectives, knowledges learned by GNN models tend to be more transferable for different designs, which is highly desirable.

B. Review of the EM and EM modeling

EM is a diffusion phenomenon of metal atoms migrating from cathode to anode of confined metal interconnect wires due to the momentum exchange between the conducting electrons and metal atoms [2]. With the EM driving force, the hydrostatic stress increases over time. When the stress reaches a critical value, voids is nucleated at the cathode and hillock is created at the anode of the interconnects. This eventually leads to an open or short circuit, which is a EM-induced reliability problem in modern VLSI circuits.

Black's equation predicts EM-induced the time-to-failure (TTF) based on an empirical or statistical data fitting, which only works for one specific single wire [2]. Blech's limit, which is an immortality check method, can not estimate transient hydrostatic stress and is subject to growing criticism due to unnecessary overdesign [3]. To mitigate this problem, the physics-based EM model, Korhonen equations [19], is employed to describe the hydrostatic stress evolution for general multi-segment interconnects.

The general multi-segment interconnect consists of n nodes, including p interior junction nodes $x_r \in \{x_{r1}, x_{r2}, \cdots, x_{rp}\}$ and q block terminals $x_b \in \{x_{b1}, x_{b2}, \cdots, x_{bq}\}$, and several branches. The physics-based Korhonen's PDE for this general structure in nucleation phase can be formulated as follows [17], [22].

$$\begin{aligned}
&\frac{\partial \sigma_{ij}(x,t)}{\partial t} = \frac{\partial}{\partial x}\left[\kappa_{ij}\left(\frac{\partial \sigma_{ij}(x,t)}{\partial x} + G_{ij}\right)\right], t > 0\\
&BC: \sigma_{ij_1}(x_i,t) = \sigma_{ij_2}(x_i,t), t > 0\\
&BC: \sum_{ij} \kappa_{ij}\left(\left.\frac{\partial \sigma_{ij}(x,t)}{\partial x}\right|_{x=x_r} + G_{ij}\right)\cdot n_r = 0, t > 0 \qquad (1)\\
&BC: \kappa_{ij}\left(\left.\frac{\partial \sigma_{ij}(x,t)}{\partial x}\right|_{x=x_b} + G_{ij}\right) = 0, t > 0\\
&IC: \sigma_{ij}(x,0) = \sigma_{ij,T}
\end{aligned}$$

where BC and IC are boundary and initial conditions respectively, ij denotes a branch connected to nodes i and j, n_r represents the unit inward normal direction of the interior junction node r on branch ij. $\sigma(x,t)$ is the hydrostatic stress, $G = \frac{Eq*}{\Omega}$ is the EM driving force, and $\kappa = D_a B\Omega/k_B T$ is the diffusivity of stress. E is the electric field, $q*$ is the effective charge. $D_a = D_0 \exp(\frac{-E_a}{k_B T})$ is the effective atomic diffusion coefficient. D_0 is the pre-exponential factor, B is the effective bulk elasticity modulus, Ω is the atomic lattice volume, k_B is the Boltzmann's constant, T is the absolute temperature, E_a is the EM activation energy. σ_T is the initial thermal-induced residual stress.

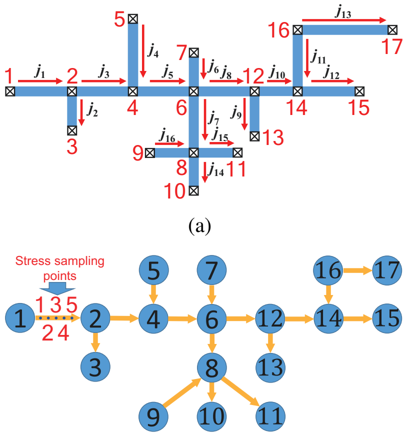
III. Problem Formulation

This work aims to predict the transient hydrostatic stress over time on the general multisegment interconnect using GCN. The current densities for each branch can be calculated by IR drop solver, such as SPICE circuit simulator. Aside from current density, the width and length of each branch also impact the EM stress. Thus, the input features include current density, width, length and time. The output feature is the hydrostatic stress. Table I lists all inputs and output of GCN model.

TABLE I
INPUT AND OUTPUT FOR GCN MODEL

	Features	Type	Definition
input	J	edge	current density (A/m^2)
	L	edge	length (μm)
	W	edge	width (μm)
	t	edge/node	time (s)
output	σ	edge/node	stress (Pa)

The general multi-segment interconnect can be naturally viewed as a graph, as shown in Fig. 1. Fig. 1(a) shows an interconnect tree extracted from a real power delivery network (PDN) where the current has direction. Each junction and branch can be represented by node and edge in a graph, respectively. To describe the direction of current, a directed graph is employed to represent the tree structure, as shown in Fig. 1(b). Then, the embedding features can be mapped into nodes and edges. Therefore, we can obtain a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ which consists of a node set $\mathbf{V}$ and a directed edge set $\mathbf{E}$. The node embedding feature of input is time ($\mathbf{x}_v = [t], v \in \mathbf{V}$). The edge embedding features of input are current density, length, width and time ($\mathbf{x}_{v,u} = [J, L, W, t]^T, (v, u) \in \mathbf{E}$), in which J is positive is current flows from v to u and vice versa. The node embedding feature of output is stress ($\mathbf{z}_v = [\sigma], v \in \mathbf{V}$) at node v. The edge embedding features of output are stress at five sampling points ($\mathbf{z}_{v,u} = [\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5]^T, (v, u) \in \mathbf{E}$), as shown in Fig. 1(b). Based on the embedding features of input and output, the graph learning task is node-edge regression.



(a)

(b)

Fig. 1. (a) Schematic and (b) directed graph of a multi-segment interconnect.

To obtain the training set, we implemented an interconnect tree generation algorithm which randomly generates branches with various width, length and current densities in a fixed area of 256×256 μm^2. The resulting dataset contains 2500 unique designs and the number of branches range from 5 to 110. To obtain the ground truth stress results, the designs are simulated in a finite element based commercial software COMSOL, and for each design, 10 results at 1st to 10th discrete aging years are produced.

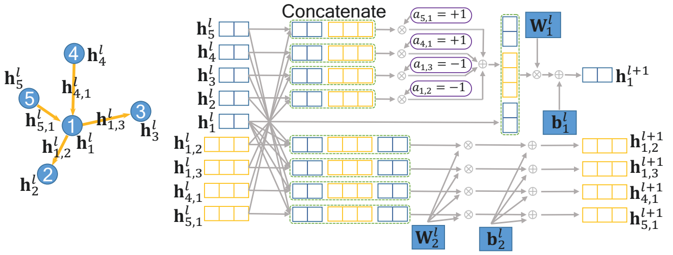
IV. EMGRAPH: EM GRAPH NEURAL NETWORK

In this section, we focus on developing a GCN model which takes node and edge features as input and output as described in Section III. However, there is no GNN model for node-edge regression task. Therefore, we proposed our own GCN model, which is called EMGraph, for EM stress assessment. The primary challenges are as follows: first, the stress ranges from -2×10^9 to 2×10^9Pa. It is difficult for a neural network to predict the entire range which spans 10 orders of magnitude; Second, edge is directed and the output has both node and edge features. The GCN model should be complicated enough to deal with this graph; Third, the accuracy of stress should be high. However, the regression using GCN has low accuracy.

A. Data rescaling

It is commonly accepted that values around zero are numerically more stable for neural networks. Thus, we rescale all input and output features to -1 to 1 using min-max normalization method. However, Considering the large range of the output stress values (4×10^9Pa), such normalization squeezes values with less orders of magnitude into a small range around zero. This may lead to more accurate predictions at large stress points but impact the accuracy at the smaller ones as they may be considered noises by the model. This concern is verified by our experiment results in Sec. V-A. However, such configuration is actually in favor of our goal since the large stress points are the ones that may lead to reliability issues and require higher accuracy, while the smaller ones are less important and can be ignored. This is further justified as hydrostatic stress typically will exceed the critical stress before void is nucleated [39].

B. Graph convolution network



(a)

(b)

Fig. 2. (a) One node and four neighborhood nodes. Both nodes and edges are embedded with features. Each edge has the direction. (b) One hidden layer node-edge embedding update of EMGraph. The convolution operation is to aggregate the information of its neighborhood nodes and connected edges into one node. In the meantime, a convolution also aggregates the information of two end nodes into one edge.

We propose an EMGraph architecture based on the GraphSAGE network [26] since the GraphSAGE only works for the node classification task. The input layer of EMGraph is represented by

$$\mathbf{h}_v^0 = \mathbf{x}_v \text{ and } \mathbf{h}_{v,u}^0 = \mathbf{x}_{v,u} \tag{2}$$

where $\mathbf{h}_v^0$ and $\mathbf{h}_{v,u}^0$ are node and edge hidden embedding features of the 0th layer, respectively. The lth hidden layer of EMGraph is given by

$$\mathbf{h}_v^{l+1} = \text{ReLU}(\mathbf{b}_1^l + \mathbf{W}_1^l(\mathbf{h}_v^l || \sum_{u \in N(v)} a_{v,u}(\mathbf{h}_u^l || \mathbf{h}_{v,u}^l))) \tag{3}$$

$$\mathbf{h}_{v,u}^{l+1} = \text{ReLU}(\mathbf{b}_2^l + \mathbf{W}_2^l(\mathbf{h}_v^l || \mathbf{h}_u^l || \mathbf{h}_{v,u}^l)) \tag{4}$$

where ReLU$(\cdot)$ is an activation function, $a_{v,u}$ is a known parameter representing the direction of edge, $N(v)$ is the set of neighborhood nodes of the node v, $||$ denotes concatenation, $\mathbf{h}_v^l$ and $\mathbf{h}_{v,u}^l$ are node and edge hidden embedding features of the lth layer, $\mathbf{W}^l$ and $\mathbf{b}^l$ are learnable weights and biases, respectively. Fig. 2 gives an example of one hidden layer node-edge embedding update for EMGraph. The edge features can impact the node features. In turn, the node features can also influence the edge features. The convolution of EMGraph consists of two parts: one is to aggregate the information of its neighborhood nodes and connected edges into one node and another is to aggregate the information of two end nodes into one edge. Concatenation is similar to the “skip connections” in different layers and is also employed to consider both node and edge features. Therefore, EMGraph can do the node-edge regression task. To represent directed edge, we introduce a parameter $a_{v,u}$, which is 1 for the inward direction and -1 for the outward direction at the node, as shown in Fig. 2. The output layer of EMGraph is expressed as

$$\mathbf{z}_v = \mathbf{b}_1^L + \mathbf{W}_1^L(\mathbf{h}_v^L || \sum_{u \in N(v)} a_{v,u}(\mathbf{h}_u^L || \mathbf{h}_{v,u}^L)) \tag{5}$$

$$\mathbf{z}_{v,u} = \mathbf{b}_2^L + \mathbf{W}_2^L(\mathbf{h}_v^L || \mathbf{h}_u^L || \mathbf{h}_{v,u}^L)) \tag{6}$$

C. Node and edge decoder

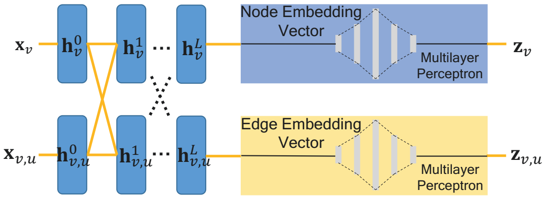


Fig. 3. Framework of EMGraph with Multilayer perceptron network.

To improve the modeling capacity of the EMGraph, we feed node and edge features of the output in Section IV-B to node and edge decoders which are two separate multilayer perceptron (MLP) networks, as shown in Fig. 3. The GCN model, which is the first part of EMGraph, is only responsible for graph embedding which converts the input graph into latent edge and node features. These features are trained to extract and contain important neighboring information for stress prediction which are then utilized by node and edge decoders to infer the stress values on each branch.

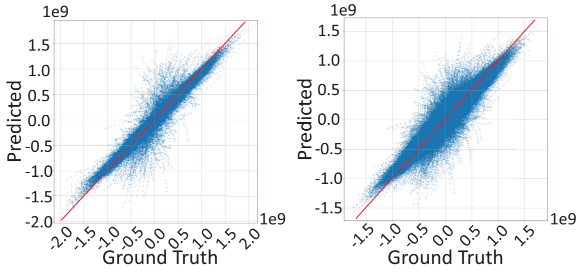
We propose such architecture basing on the observation that after certain point, increasing the number of hidden units and hidden layers in GCN model does not help much on improving the output stress accuracy. Due to the small size of GCN, modeling capacity needs to be increased to further improve the accuracy. Therefore, we employ another way to increase number of learnable parameters by combining GCN and MLP network. MLP network can further process the information as it learns node and edge features separately.

V. Experimental Results

In this section, we demonstrate the accuracy and speed of EMGraph models on our randomly generated dataset consisting of 2500 circuit designs. The dataset is randomly split into training set with 2125 samples and test set with 375 samples. The EMGraph model is implemented in Deep Graph Library (DGL) [40], which is developed for deep learning on graph and built on PyTorch. For the GCN part, we employed 5 layers with number of hidden features set to 8, 16, 32, 64 and 128 respectively. For the node and edge decoders, the fully connected layers are set to [128, 256, 1024, 256, 64, 1] and [128, 256, 1024, 256, 64, 5] separately. The model is trained and tested on a Linux server with 2 Xeon E5-2699v4 2.2 GHz processors and Nvidia Titan RTX GPU. The training batch size is set to 32 and the learning rate of the Adam optimizer is set to 10^{-4}. The cross validation technique is employed and the model was trained for 80 epochs in 2 hours.

A. Accuracy of EM Stress Prediction

Fig. 4(a) and Fig. 4(b) shows the predicted stresses vs ground truth of all 223380 nodes and 1114350 edges in the test set. The results are obtained using trained EMGraph to predict EM-induced stresses on all 375 test cases which were never seen by the model during the training process. For each case, 10 predictions for the 1st to 10th aging year are conducted.



(a) (b)

Fig. 4. EMGraph prediction vs ground truth on all testing cases for (a) nodes and (b) edges.

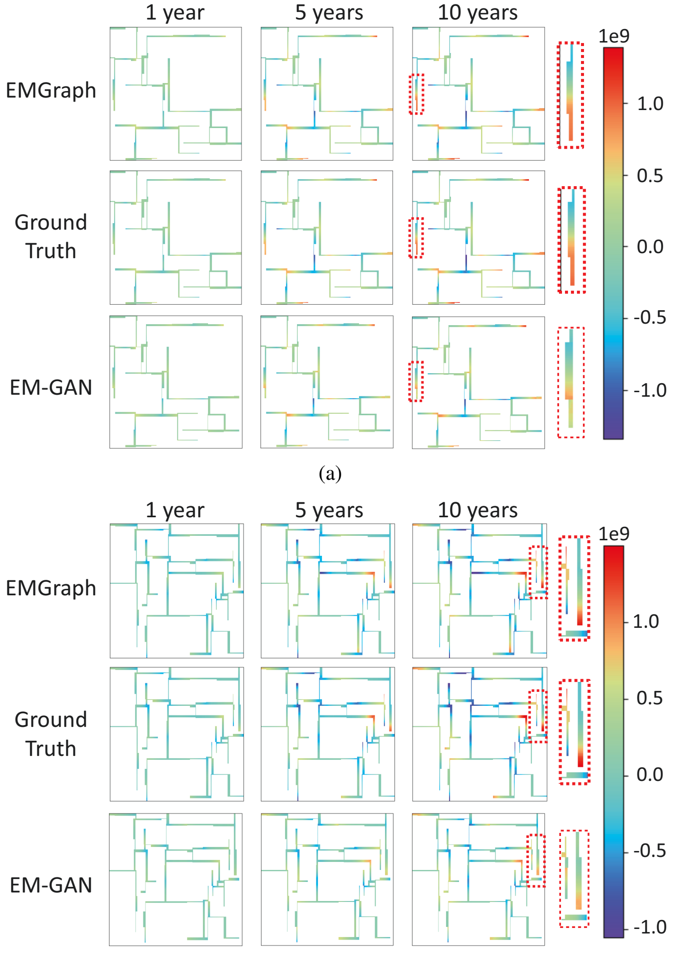
As is shown in Fig. 4, the stress values vary in a large numerical range from -2×10^9 to 2×10^9 Pa. For both nodes and edges, there are more outliers in the range around zero while the results tend to be more accurate at both ends of the full range. This is acceptable and indeed what we desired since the large stresses are the ones that may lead to reliability issues and require higher accuracy.

For better illustration and comparison, we convert all prediction results into stress maps in which stress values are filled into the interconnects topology and shown in colors. The root-mean-square error (RMSE) between the predicted stress map and its ground truth is employed to evaluate the result accuracy.

EMGraph yields RMSE from 1.6×10^7 to 1.8×10^8 Pa on the test set and achieves an averaged RMSE of 6×10^7 Pa which translates to 1.5% error rate considering the full stress value range of 4×10^9 Pa.

Fig. 5 shows the stress maps of both the best and the worst cases (in terms of averaged RMSE) predicted by EMGraph. The results of EM-GAN [32] and ground truth obtained from COMSOL are also shown in parallel for comparison. As the EM-induced stress is a time varying process, for each case, we show the results at the 1st, 5th and the 10th aging year for a better illustration of the stress evolution.

As shown in Fig. 5, EMGraph yields much better accuracy in both cases (7× better in the best case and 2.5× better in the worst case). Although the RMSE of the worst case is 11× larger than that of the best case, the predicted stress map for the worst case is still quite accurate and closer to the ground



(a)

(b)

Fig. 5. Comparison of the predicted stress maps obtained by EMGraph and EM-GAN on (a) best case and (b) worst case.

truth than the result of EM-GAN. Compared with EM-GAN, EMGraph yields much better accuracy in both cases (7× better in the best case and 2.5× better in the worst case). More statistics on the comparison between EM-GAN and EMGraph are listed in Table II.

B. Speed of Inference

The training process of the EMGraph costs 2 hours and the trained model consists of a 441KB GNN, a 2252KB edge decoder and a 2251KB node decoder. The lightweight model together with the highly parallelizable nature of the graph input makes EMGraph have the potential to yield fast inference speed. In what follows, we compare the speed performance of EMGraph, EM-GAN and also the state-of-the-art work [17], which is a separation-of-variables based analytical method.

As is shown in Table II, the average inference speed of EMGraph for each case is only 0.27ms which is 14× and 265× faster than the 3.8ms and 71.7ms inference speeds yielded by EM-GAN and work [17]. These statistics are obtained by running three methods on all test cases and taking the average of the time cost for each case. Although work [17] yields higher accuracy, EMGraph achieves 2 orders of magnitude speedup while still rendering comparable results accuracy. Moreover, as EMGraph is treating each graph input as individual nodes and edges which can be processed in parallel, it has the potential to achieve even more significant speedups on large designs.

TABLE II
ACCURACY AND SPEED COMPARISON

Metrics	EMGraph	EM-GAN	State-of-the-art
Max RMSE	1.8×10^8 Pa	5.2×10^8 Pa	Close to ground truth
Min RMSE	1.6×10^7 Pa	1.2×10^8 Pa	
Mean RMSE	6×10^7 Pa	2.6×10^8 Pa	
Mean Error Rate	1.5%	6.6%	
Inference Speed	0.27ms	3.8ms	71.7ms

C. Scalability on large unseen designs

In this section we further test the scalability of the trained EMGraph model on 13 large designs which are randomly generated without any limitations on their dimensions. Although we trained EMGraph on the dataset with fixed size of 256×256 μm^2, we note that EMGraph is not limited to a certain size, contrasting with EM-GAN which is only applicable to the size it was trained on. We fix the size of the dataset in this work just to make a fair apple-to-apple comparison between two models. The scalability of EM-GAN is limited due to its image processing-based nature and the cost of its forward propagation becomes exponentially large as the input size increases, which is not the case for EMGraph. The inference cost of EMGraph is linearly dependent on the number of branches in its input graph and such calculations are highly suitable for parallelization which further boosts its scalability to large designs.

Fig. 6 shows the stress map predicted by EMGraph for the largest design with 401 branches at the 10th aging year. EMGraph maintains its high accuracy on the large designs and achieves an averaged RMSE of 1.1×10^8 Pa across all 13 large designs with a 8×10^7 Pa minimum and 1.6×10^8 Pa maximum. The number of branches in these large designs ranges from 113 to 401 which are much larger than the cases in previous test set. We only compare the results with ground truth here as EM-GAN is not applicable to such large designs.

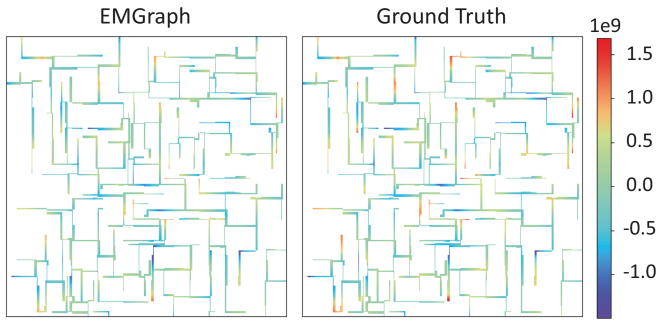


Fig. 6. EMGraph prediction vs ground truth on large design with 401 branches.

Although the accuracy on large cases slightly degenerates compared to the previous test set, considering these are much more complicated designs and were never seen by the model before, such accuracy is still acceptable. We also recorded the time cost of EMGraph and the average inference speed for each case is only 0.32ms which is still at the same level as it yields on the smaller cases thanks to the parallel nature of the nodes and edges in the input graph.

VI. Conclusion

In this paper, we propose a new graph neural network model, which is called *EMGraph*, fast prediction of the transient EM stress of the general multisegment interconnect in VLSI systems. EMGraph performs the node-edge regression task to predict the stress at the wire segment (edge). Compared with GAN based image method, the proposed EMGraph model can learn more transferable knowledge to predict stress distributions on new graphs without retraining via inductive learning. Experimental results show the model has smaller size, better accuracy and faster speed over the recently proposed learning-based method, EM-GAN method, on several interconnect trees benchmarks. Therefore, EMGraph is very powerful and suitable for the transient EM stress assessment.

References

[1] "International technology roadmap for semiconductors (ITRS)," 2015, http://www.itrs2.net/itrs-reports.html.

[2] J. R. Black, "Electromigration-A Brief Survey and Some Recent Results," *IEEE Trans. on Electron Devices*, vol. 16, no. 4, pp. 338–347, Apr. 1969.

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

[4] M. Hauschildt, C. Hennesthal, G. Talut, O. Aubel, M. Gall, K. B. Yeap, and E. Zschech, "Electromigration Early Failure Void Nucleation and Growth Phenomena in Cu And Cu(Mn) Interconnects," in *IEEE Int. Reliability Physics Symposium (IRPS)*, 2013, pp. 2C.1.1–2C.1.6.

[5] V. Sukharev, "Beyond Black's Equation: Full-Chip EM/SM Assessment in 3D IC Stack," *Microelectronic Engineering*, vol. 120, pp. 99–105, 2014.

[6] R. De Orio, H. Ceric, and S. Selberherr, "Physically based models of electromigration: From black's equation to modern tcad models," *Microelectronics Reliability*, vol. 50, no. 6, pp. 775–789, 2010.

[7] X. Huang, A. Kteyan, S. X.-D. Tan, and V. Sukharev, "Physics-Based Electromigration Models and Full-Chip Assessment for Power Grid Networks," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 11, pp. 1848–1861, Nov. 2016.

[8] V. Sukharev, A. Kteyan, and X. Huang, "Postvoiding stress evolution in confined metal lines," *IEEE Transactions on Device and Materials Reliability*, vol. 16, no. 1, pp. 50–60, 2016.

[9] H. Chen, S. X.-D. Tan, X. Huang, T. Kim, and V. Sukharev, "Analytical modeling and characterization of electromigration effects for multibranch interconnect trees," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 11, pp. 1811–1824, 2016.

[10] V. Mishra and S. S. Sapatnekar, "Predicting Electromigration Mortality Under Temperature and Product Lifetime Specifications," in *Proc. Design Automation Conf. (DAC)*, Jun. 2016, pp. 1–6.

[11] H.-B. Chen, S. X.-D. Tan, J. Peng, T. Kim, and J. Chen, "Analytical modeling of electromigration failure for vlsi interconnect tree considering temperature and segment length effects," *IEEE Transaction on Device and Materials Reliability (T-DMR)*, vol. 17, no. 4, pp. 653–666, 2017.

[12] S. Chatterjee, V. Sukharev, and F. N. Najm, "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, Jul. 2018.

[13] C. Cook, Z. Sun, E. Demircan, M. D. Shroff, and S. X.-D. Tan, "Fast electromigration stress evolution analysis for interconnect trees using krylov subspace method," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 5, pp. 969–980, May 2018.

[14] S. Wang, Z. Sun, Y. Cheng, S. X.-D. Tan, and M. Tahoori, "Leveraging recovery effect to reduce electromigration degradation in power/ground TSV," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*. IEEE, Nov. 2017, pp. 811–818.

[15] H. Zhao and S. X.-D. Tan, "Postvoiding fem analysis for electromigration failure characterization," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 11, pp. 2483–2493, Nov. 2018.

[16] A. Abbasinasab and M. Marek-Sadowska, "RAIN: A tool for reliability assessment of interconnect networks—physics to software," in *Proc. Design Automation Conf. (DAC)*. New York, NY, USA: ACM, 2018, pp. 133:1–133:6.

[17] L. Chen, S. X.-D. Tan, Z. Sun, S. Peng, M. Tang, and J. Mao, "Fast analytic electromigration analysis for general multisegment interconnect wires," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 1–12, 2019.

[18] S. X.-D. Tan, M. Tahoori, T. Kim, S. Wang, Z. Sun, and S. Kiamehr, *VLSI Systems Long-Term Reliability – Modeling, Simulation and Optimization*. Springer Publishing, 2019.

[19] M. A. Korhonen, P. Bo/rgesen, K. N. Tu, and C.-Y. Li, "Stress evolution due to electromigration in confined metal lines," *Journal of Applied Physics*, vol. 73, no. 8, pp. 3790–3799, 1993.

[20] C. Cook, Z. Sun, E. Demircan, M. D. Shroff, and S. X.-D. Tan, "Fast Electromigration Stress Evolution Analysis for Interconnect Trees Using Krylov Subspace Method," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 5, pp. 969–980, May 2018.

[21] V. Sukharev and F. N. Najm, "Electromigration Check: Where the Design and Reliability Methodologies Meet," *IEEE Transactions on Device and Materials Reliability*, vol. 18, no. 4, pp. 498–507, Dec. 2018.

[22] X. Wang, Y. Yan, J. He, S. X.-D. Tan, C. Cook, and S. Yang, "Fast physics-based electromigration analysis for multi-branch interconnect trees," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*. IEEE, Nov. 2017, pp. 169–176.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[24] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020.

[25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representation*, 2017.

[26] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017, pp. 1024–1034.

[27] K. Settaluri and E. Fallon, "Fully automated analog sub-circuit clustering with graph convolutional neural networks," in *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2020, pp. 1714–1715.

[28] H. Ren, G. F. Kokai, W. J. Turner, and T. S. Ku, "Paragraph: Layout parasitics and device parameter prediction using graph neural networks," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[29] Y. C. Lu, S. S. Kiran Pentapati, L. Zhu, K. Samadi, and S. K. Lim, "Tp-gnn: A graph neural network framework for tier partitioning in monolithic 3d ics," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[30] H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H. S. Lee, and S. Han, "Gcn-rl circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[31] Y. Li, Y. Lin, M. Madhusudan, A. Sharma, and W. Xu, "A customized graph neural network model for guiding analog ic placement," in *2020 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2020, pp. 1–9.

[32] W. Jin, S. Sadiqbatcha, Z. Sun, H. Zhou, and S. X.-D. Tan, "Em-gan: Data-driven fast stress analysis for multi-segment interconnects," in *Proc. IEEE Int. Conf. on Computer Design (ICCD)*, Oct. 2020, pp. 296–303.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016, http://www.deeplearningbook.org.

[34] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations," *arXiv e-prints*, p. arXiv:1711.10561, Nov. 2017.

[35] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "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.

[36] J. Sirignano and K. Spiliopoulos, "DGM: A deep learning algorithm for solving partial differential equations," *Journal of Computational Physics*, vol. 375, pp. 1339 – 1364, 2018.

[37] X. Meng and G. E. Karniadakis, "A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems," *Journal of Computational Physics*, vol. 401, p. 109020, 2020.

[38] Y. Zhang, H. Ren, and B. Khailany, "Grannite: Graph neural network inference for transferable power estimation," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[39] C. M. Tan, *Electomigration in ULSI Interconnects*, ser. International Series on Advances in Solid State Electronics and Technology. Word Scientific, 2010.

[40] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, "Deep graph library: A graph-centric, highly-performant package for graph neural networks," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.