GridNetOpt: Fast Full-Chip EM-Aware Power Grid Optimization Accelerated by Deep Neural Networks

Han Zhou, Yibo Liu, Wentian Jin, and Sheldon X.-D. Tan *Senior Member, IEEE*

***Abstract*—This article presents a fast full-chip electromigration (EM) aware IR drop constrained optimization framework, named *GridNetOpt*, for on-chip power grid networks accelerated by deep neural networks (DNN). Compared to the existing linear programming-based methods, the new method employs more flexible conjugate gradient-based optimization to size the wire width of the power grids. To mitigate the high cost of sensitivity calculation of the adjoint network using full-chip IR drop analysis at every iteration step, the sensitivity is computed via a trained conditional generative adversarial network (CGAN). The new method exploits the differentiable characteristics of DNNs for fast sensitivity computation. The sensitivity, which is the node voltage with respect to wire resistance, will guide the search direction during the optimization process. In order to consider more accurate EM failure effects, the training data is obtained from the power grids under different wire widths and current loads analyzed by a state-of-the-art full-chip multi-physics-based coupled EM-IR drop analysis tool. This is in contrast with the existing linear programming-based methods, in which only immortal wires or wires with non-zero resistance can be dealt with. Numerical results on a number of synthesized power grid benchmarks from ARM Cortex-M0 processor designs show that the proposed *GridNetOpt* can lead to at least an order of magnitude speedup over the conjugate gradient-based method using the traditional adjoint network method. Compared to the previous localized power grid fixing work with *GridNet*, *GridNetOpt* leads to smaller area overhead for all the benchmarks we tested. It can also reduce IR drops for power grid circuits with immortal wires, which is not possible with the localized *GridNet* method.**

I. Introduction

On-chip power distribution networks (PDNs) are a crucial backbone for feeding power to all transistors from top metals on a chip, because they directly affect chip performance and reliability. At the same time, electromigration (EM) remains the top failure mechanism for copper-based interconnects in all the subnanometer technologies. The International Roadmap for Devices and Systems (IRDS) [2] predicts that the allowable current density will continue to decrease due to EM while the required current density to drive the gates will continue to increase. As a result, the EM-related aging and reliability will become worse for current 5nm and below technologies.

EM is a physical phenomenon of the migration of metal atoms along the direction of the applied electrical field. Atoms migrate along the trajectory of conducting electrons. For practical VLSI chips, the on-chip power supply networks are most susceptible to EM failures because of large and unidirectional current densities [3], [4]. Due to EM aging effects, voids may be formed in the interconnects of the power grid networks, which can lead to resistance increase of the wire segment, or even open circuit, making the IR drop of the power grids increase. Therefore, EM-induced aging and IR drop changes at the target lifetime have to be taken into consideration to make the PDN more robust. We notice that EM effects may also lead to hillocks or extrusion at the anode nodes of the wires, which may bring about short circuits. However, the majority of the EM failures are due to void nucleation [5] and hence we focus on the void-induced EM failure in this work.

To design robust PDNs in the physical synthesis flow, the wires have to be properly sized after the topology of the PDN has been determined to minimize the area and meet the IR drop requirement at the target lifetime. Many research efforts have been investigated in the past based on nonlinear or linear optimization methods [6], [7], [8], [9], [10], [11]. Early works were mainly based on Black's EM model. This is also the requirement widely adopted in industry today – EM constraint is simply represented as the maximum allowed current density of individual wire segments to avoid nucleation. Recent studies indicate that we have to analyze all the wire segments of the entire interconnect wire simultaneously [12], [13], [14], [15], [16], [17], [18].

To alleviate the above drawback, some works use new multi-segment EM models to size the power grids to fix the EM failures and IR drop violations came up. Zhou *et al.* [19] proposed a power grid network sizing method based on a multi-segment EM immortality check criteria. It automatically considers all the wire segments and their interactions within an interconnect tree. However, the EM immortality constrained optimization is still too conservative as it requires all the interconnect trees to be immortal, i.e., void nucleations are not allowed. To further mitigate this issue, Moudallal *et al.* [11] proposed to directly consider EM-induced IR drops instead of EM constraints on the time-varying power grid networks. It can consider post-voiding resistance change of wires based on finite difference analysis of EM-induced stress in multi-segment wires. Then the resulting nonlinear problem is solved by applying successive linear programming. This method, however, may suffer high computational costs if the number of violation nodes is large as the sensitivities of those violating nodes needs to be computed by solving the circuit matrices. Furthermore, this method has the limitation in which wires can only be sized up, which restricts its application in many practical problems.

On one hand, Chang *et al.* [20] introduced a learning-based EM violation waiver system, which investigates every EM violation and takes an expert decision to either ignore the violation (waive-off) or resolve it (must-fix) in the design. However, the proposed method cannot directly perform the EM violation fixing. On the other hand, deep neural networks (DNN) have propelled an evolution in machine learning fields and redefined many existing applications with new human-level AI capabilities. DNNs such as convolution neural networks (CNN) have been applied to many cognitive

Han Zhou (hzhou012@ucr.edu) is with Synopsys Inc. The work was performed at University of California, Riverside.

Yibo Liu, Wentian Jin, and Sheldon X.-D. Tan (stan@ece.ucr.edu) are with Department of Electrical and Computer Engineering, University of California, Riverside.

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.

The preliminary results of this work have been published in [1].

applications such as visual object recognition, object detection, speech recognition, natural language understanding, etc. due to dramatic accuracy improvements in those tasks [21].

Recently, generative adversarial networks (GAN) [22] gained much traction as they can learn features (latent representation) without extensively annotated training data. GAN-based methods have been applied for solving several EDA problems such as layout lithography analysis [23], sub-resolution assist feature generation [24], and analog layout well generation [25] and high level thermal analysis [26] and electromgration analysis [27].

Inspired by the modeling power of the DNN/GAN for 2D images, in this article, we try to mitigate the limitations on the existing EM-aware power grid optimizations. We develop a new fast EM-aware optimization framework, called *GridNetOpt*, for full-chip power grid network sizing and fixing. It capitalizes on the power of fast GAN-based full-chip IR drop estimation method, which not only provides fast EM-induced IR drop estimation, but also enables fast and scalable sensitivity computation for optimization via the inherent differential function of trained GAN models. The key contributions of this paper are as follows:

- First, the new method applies a more general and flexible conjugate gradient based optimization framework instead of the existing sequence of linear programming method. To be more specific, it only requires sensitivity information to size any given power grids, with or without mortal wires. Compared to the successive linear programming method [11], the proposed method does not have the limitation of reducing the IR voltage drop by only widening the wires of the given power grid, and there is no need to solve matrices to get sensitivities.
- Instead of using the traditional adjoint network-based sensitivity computation method, which requires full-chip IR drop analysis at every iteration step, we propose to use a deep learning based model for sensitivity computation. Once the model is trained, obtaining sensitivity will be much simpler and faster. The trained GAN model not only provides the IR drop information at the target aging time but also provides the critical sensitivity information of node voltage with respect to the wire resistance or width. The sensitivity computation cost is marginal for any given power grid designs with the same topology by taking advantage of the auto-differentiation function of the DNN model.
- We leverage the previously proposed GAN-based full-chip IR drop analysis tool *GridNet* [1] for fast IR drop estimation. *GridNet* is trained using 2D EM-induced IR drop maps of power grid designs at different aging time under different wire widths and current workloads. The EM-induced IR drops of those power grids are simulated from a coupled EM-IR analysis tool, *EMspice* [28], which computes time-varying EM-induced IR drop and can handle both early failure (open circuit) and late failure (non-zero resistance) cases.
- Numerical results on a number of synthesized power grid benchmarks from ARM Cortex-M0 processor designs show that the proposed *GridNetOpt* can lead to an order of magnitude or more speedup over the conjugate gradient-based method using the traditional adjoint network method. Compared to the previously proposed localized power grid fixing method with *GridNet*, *GridNetOpt* can lead to smaller area overhead for all the benchmarks we tested due to global optimization nature. It can also reduce IR drops for power grid circuits with immortal wires, which is not possible with the previous method.

This paper is organized as follows: Section II reviews the related preliminary works. Section III presents the details of the GAN-based EM-aware IR drop prediction approach. Section IV shows the formulation of the new EM-induced voltage constrained optimization and its solution method. Section V introduces the optimization strategies, including the fast gradient calculation via deep neural networks. Experimental results and discussions are summarized in Section VI. Section VII concludes the paper.

II. Related works

In this section, we summarize some related literature on physics-based EM-induced IR analysis and machine learning-based IR drop analysis methods.

A. Full-chip EM-induced IR drop analysis

EM aging process typically leads to resistance increase or even open-wire segments. For on-chip mesh-structured power grid networks, due to its inherent design redundancy, a few wire failures may not immediately result in a significant IR drop increase. But as more wires nucleate, the IR drop will eventually lead to timing violations. As a result, the power grid networks become time-varying networks with time-varying IR drops due to the EM-induced aging process [14], [15], [29], [28]. On the other hand, the failed wire segments alter the current distributions of all the interconnect wires, which may further accelerate the failure process. Hence, one has to consider the interplay between the two physics: electrical characteristics and hydrostatic stress in the interconnect wires.

EMspice [28], [30] is a full-chip coupled EM-IR drop co-simulation tool that considers the dynamic interplay between the hydrostatic stress and electrical characteristics in a power grid network. The tool consists of a finite difference time domain (FDTD) solver for EM stress and a linear network DC solver for IR drop, which can be described as

$$\mathbf{C}\dot{\sigma}(t) = \mathbf{A}\sigma(t) + \mathbf{P}I(t), \tag{1}$$

$$\mathcal{V}_v(t) = \textstyle\int_{\Omega_L} \frac{\sigma(t)}{B} d\mathcal{V}, \tag{2}$$

$$\mathbf{M}(t) \times u(t) = \mathbf{P}I(t), \tag{3}$$

$$\sigma(0) = \left[\sigma_1(0), \sigma_2(0), ..., \sigma_n(0)\right], at\ t = 0 \tag{4}$$

Specifically, in the nucleation phase, hydrostatic stress is modeled by the Korhonen's equation with zero-flux boundary condition at the terminals and initial stress condition. After the FDTD process [31], the partial differential equation will be converted to the linear time invariant (LTI) system as shown in Eq. (1). Suppose we have n nodes, then $\mathbf{C}$ is an $n \times n$ identity matrix and $\mathbf{A}$ is an $n \times n$ coefficient matrix. Note that $\sigma(0)$ denotes the initial stress at $t = 0$. In the incubation phase [17], a void starts to form, the void volume and stress distribution of the remaining wire are correlated by the atom conservation equation as shown in Eq. (2), where $\mathcal{V}_v(t)$ is the void volume, Ω_L is the volume of the remaining interconnect wire and $\mathcal{V}$ is the volume of the wire.

In the growth phase, the void continues to grow and thus the wire resistance starts to increase. Modified nodal analysis (MNA) is applied to calculate IR drops as shown in Eq. (3). $\mathbf{M}(t)$ is the conductance matrix of the power grid network. It is time-varying because wire resistance changes with EM

This article has been accepted for publication in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. This is the author's version which has not been fully edi
content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2022.3206397

failure process. $\mathbf{P}$ is a $b \times p$ input matrix, where p is the number of inputs. $u(t)$ represents the node voltages of the network and $I(t)$ contains the current sources from the function blocks of the chips. The above equations are solved together, and finally, the resulted IR drops and EM failure hotspots at the target aging time are reported. In this work, we use data simulated from the open-source tool *EMspice* to train the DNN models.

B. Machine learning accelerated IR drop estimation

In general, IR drop analysis is concerned with voltage drop estimation from given current or power sources, which can be time-varying for dynamic analysis. Numerical techniques are well developed and perform IR drop analysis well on power grids, such as hierarchical methods, random walk methods, Krylov-subspace methods, multi-grid techniques, and vectorless verification methods.

Several machine learning-based IR drop analysis methods have been proposed based on various deep neural networks [32], [33], [34], [35], [36]. Those methods typically aim to replace the standard full-chip IR drop analysis tool such as ANSYS RedHawk, via data-driven learning and feature selection. For instance, Lin *et al.* [32] proposed a full-chip dynamic IR drop analysis based on some power and physical features extracted from cells and layouts. Fang *et al.* [33] tried to improve the scalability by training the models for the localized region of the layout. Xie *et al.* [35] proposed a CNN-based model transferable across different designs that is able to incorporate design-dependent features during preprocessing. Ho *et al.* [34] focused on incremental IR drop prediction and mitigation. The gradient boosting framework uses more electrical and physical features for training. Chhabria *et al.* [36] proposed a CNN-based generative network method, called *IREDGe*, to predict on-chip temperature and IR drop contours. Temperature and power grid analyses are mapped to image-to-image and sequence-to-sequence translation tasks. A good summary of recent work on machine learning-based IR drop analysis can be found at [37]. These machine learning methods indeed have achieved significant progress in IR drop estimation. But none of them takes EM aging effects into consideration.

III. DNN-BASED FAST EM-INDUCED IR DROP PREDICTION

A. The overall workflow of the GridNetOpt framework

Fig. 1 shows the overall workflow of the proposed *GridNetOpt* framework. The workflow consists of three phases: training, inference and optimization. The first two phases are also called *GridNet*. The training phase is shown in Fig. 1(a), the yellow block shows how the power grids are generated. Then in the red block, we use *EMspice* [28], the coupled EM-IR analysis tool, to simulate the EM-induced IR drop for synthesized power grid network. In the blue block, *GridNet* receives the EM-induced voltage from 0 to T_{target} aging years as well as the initial power grid. Electrical and geometrical information are extracted afterwards. The training process is shown with dashed arrows. Fig. 1(b) illustrates the inference phase and the sensitivity-based full-chip power grid optimization flow. *GridNet* has two outputs, one is default - the EM-induced voltage of all nodes at a specific aging year. The other is optional - the sensitivity information: sensitivity of node voltages with respect to the input resistances. These sensitivities can be obtained as a by-product from the differentiable DNN model as we will show later. The sensitivity information will then be utilized for power grid optimization in the chip design flow. After the power grid is incrementally updated, the *GridNet* model predicts new EM-induced voltage. If the IR drop violations remain unaddressed, *GridNetOpt* will perform the next round of fixing and prediction iteratively until all the IR drop violations are eliminated.

B. Feature selection for GridNet

Given a mesh-structured power network, we can look at the node voltages $u(t)$ and the input current sources $I(t)$ in Eq. (3). For the DNN-based modeling, the input features should include both $I(t)$ and $\mathbf{M}(t)$. $\mathbf{M}(t)$ is represented by the resistance vectors of wire segments in the power grid networks. The resistance of a wire segment depends on its length and cross-sectional area that is proportional to wire width. Since we deal with mesh-structured power grids, the topology of wire connections is implicitly presented if all the wire resistance or features are pre-ordered (as a vector) based on the counting order. As a result, the *GridNet* model is able to deal with different workloads, i.e., $I(t)$ and initial wire resistances (different $\mathbf{M}$ at $t = 0$) under the same power grid structure.

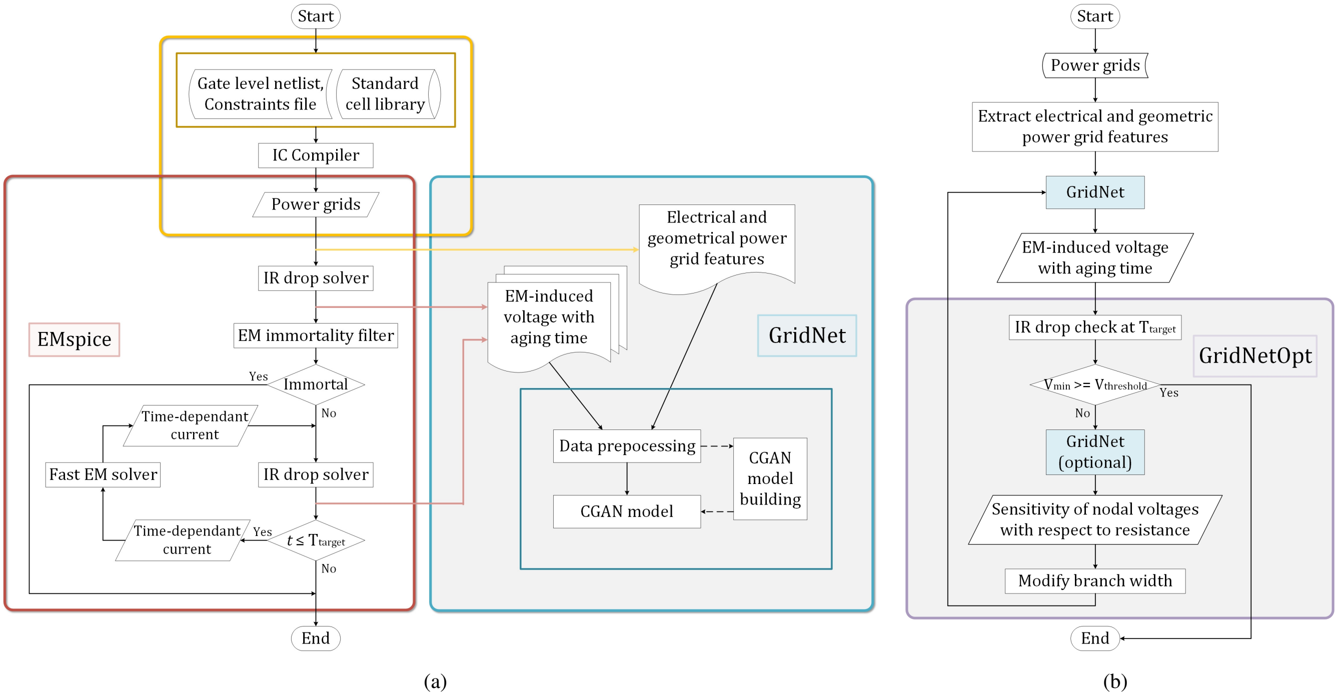
C. Training data preprocessing and representation

The preprocessing step extracts the electrical features and geometries from raw layouts. After preprocessing, the workload samples will be represented in a customized scheme.

1) Data preprocessing: Given a specific design, Synopsys IC Compiler II (ICC II) takes a synthesized gate-level netlist and a standard cell library as input, and then automatically creates the circuit layout. In the preroute (design planning) step, one important procedure is performing power network synthesis. As shown in Fig. 2(a), the power and ground network are generated based on the constraints that the user defines. It consists of VDD power nets, VSS ground nets, and external power supplies. The results later are used to examine the voltage drop, resistance, and EM effect. Fig. 2(b) shows the voltage drop from the same power grid and the unit is mV. Since our goal is to obtain EM-induced IR drops which contain aging effect, we dumped the power grid information including layout geometry, layer, via, as well as branch currents for later simulation.

Having a sufficient amount of training data is a crucial requirement for machine learning approaches. The DNN-based EM-induced IR drop prediction requires a lot of power grid samples and their corresponding ground truth EM-induced IR drop along with the aging time. However, synthesizing a large number of designs and dumping their power grid information is not realistic. We first synthesized three power grid designs, and then for each design, we randomly generated 12k different workloads respectively. The network samples have the same topology as the synthesized designs. Although they have the same number of power strips, they differ in the branch width and length. Note that different workloads can have different EM impacts, thus the wires can be sized properly later on.

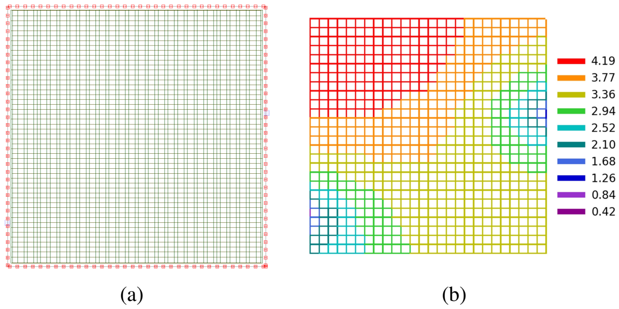
2) Data representation: Representation of data has a tremendous impact on the behavior of deep neural networks. To preserve the geometric and spatial relationship, we first encode the EM-induced voltage at each node into a matrix and then convert the matrix to a color image, as illustrated in Fig. 3. Either Python API *matplotlib.pyplot.imshow()* or MATLAB API *image()* can display the scalar data as an image. Each pixel stands for one voltage value of one node, the



(a)

(b)

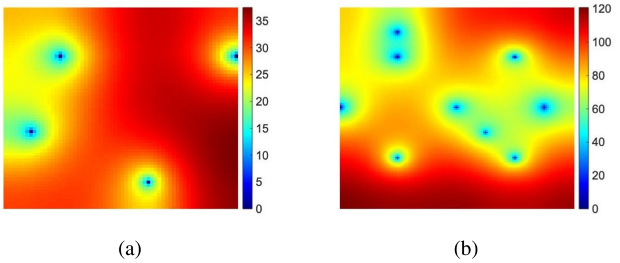
Fig. 1. Proposed *GridNetOpt* framework: (a) *GridNet* training flow; (b) *GridNet* prediction flow and *GridNetOpt* optimization flow.



(a)

(b)

Fig. 2. (a) Power and ground networks of Cortex-M0 DesignStart; (b) Voltage drop map of the power network of (a).



(a)

(b)

Fig. 3. Compact IR drop image of power grid networks (a) *Design 2*: 4k nodes; (b) *Design 3*: 16k nodes.

length and width information are discarded, while the relative position of each node and its voltage value are kept. Such compact representation will dramatically reduce the image size compared with the representation from Fig. 2(b), which will further speed up the training process.

As the pixels in our images are real voltage values, they usually do not change dramatically, e.g., the maximum voltage value is 1.05V and most values fall in the range $[0.7\ 1.05]$. The channels of input are real resistance and current, thus they have the same numerical problem. Such a small numerical range is not suitable for neural networks. As a result, we rescaled all data in the training to the range between -1 and 1.

D. The proposed GridNet architecture

GAN is a neural network model widely used in unsupervised machine learning tasks. A traditional GAN is composed of two separate deep neural networks, one is generator G and the other is discriminator D, there is no control on modes of the data being generated. In the Conditional GAN (CGAN) model, the generator learns to generate a fake sample with a specific condition rather than a generic sample from unknown noise distribution.

GAN can be seen as an enhanced version of CNN because the loss function not only consists of L2-loss between the ground truth and the predicted result, but also the score given by discriminator, which is a trainable loss function given by another CNN model. This makes the GAN perform at least the same as a CNN model if we use zero-weight to cancel out the discriminator's score in the loss function.

Back to our problem, *GridNet* does not generate voltage maps from the random noises, instead, the inputs are the selected electrical and implicit geometrical features of the power grid networks and aging time. In order to implicitly learn the distribution of the voltage and map it to the corresponding 2D voltage image, we use a CGAN as the backbone for our model.

In this work, we also tried an additional CNN model, which has exactly the same architecture as that of the generator in our proposed GAN model. The only difference is the loss function used in the training phase. The loss function of GAN consists of two parts, i.e., discriminator's output which reflects the quality of the generator's output, and the L2 difference

This article has been accepted for publication in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. This is the author's version which has not been fully edit
content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2022.3206397

between the generator's output and the ground truth, while the loss function of CNN preserves only the L2 difference part but discarded the discriminator-related part as there is no discriminator in CNN-alone architecture. The results proved that the CGAN model can produce higher accuracy and smoother node voltage images.

Fig. 4 shows the full model structure in training process. Once the model is trained, only the generator G is preserved for inference. To make the GAN model learn the temporal dynamics of EM-induced IR drops, we propose to use the time variable as the continuous condition for both generator and discriminator, which was demonstrated to be effective for financial market risk analysis [38].

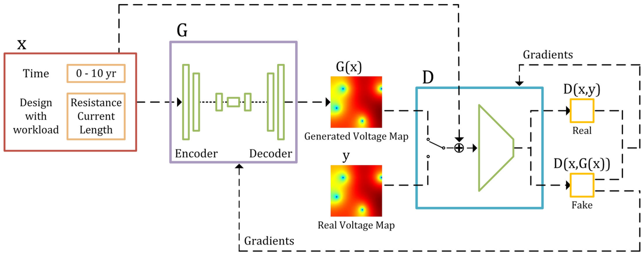


Fig. 4. The CGAN architecture for EM-induced voltage estimation.

The five channels in our input tensor are column resistance, row resistance, current source, wire length and aging time separately. We employ an encoder-decoder architecture as our generator that is widely used in image-to-image applications. The input is downsampled through a series of convolutional layers until a bottleneck layer, at which the latent features are extracted and then reversely upsampled through transposed convolutional layers. The generator is trained to extract useful latent features from the input and then reconstruct the output voltage map basing on this information.

Take a power grid design with 120 rows and 120 columns as an example, there are five channels of input for the generator: the column resistance image $img_{col} \in \mathbb{R}^{119\times120\times1}$, the row resistance image $img_{row} \in \mathbb{R}^{120\times119\times1}$, the current source image $img_{cur} \in \mathbb{R}^{120\times120\times1}$, wire length l, and aging time t. t and l are expanded into $\mathbb{R}^{128\times128\times1}$ by channel-wise duplication, respectively. In addition, the three images are all expanded to the same size, such that img_{col}, img_{row}, img_{cur}, l and t can be concatenated depth-wise. Specifically, the missing columns and rows are filled with zeros. In this example, size 120×120 is close to 128×128. We padded equally, i.e., 4 zero entries, on each side of the matrix so that the resulting image is sized to 128×128. The resulting input x given to the generator is a $128 \times 128 \times 5$ tensor with all entries normalized as described in the previous section. If the dimensions are larger than 128, e.g., it is more than 128 but smaller than 256, then an additional layer in the generator and discriminator should be introduced before the 128×128 layer. In other words, the input image should be sized to 256×256 instead of 128×128. Moreover, if the dimensions are even larger than 256, then another extra layer should be introduced before the 256×256 layer, and so on and so forth. If the dimensions are less than 128, e.g., it is less than 64, then we have to remove the 128×128 layer from the model and size the input image to 64×64 instead of 128×128. Similarly, if the dimensions are even smaller, then more layers have to be discarded.

The output of the generator is a voltage map, which is denoted as $G(x)$. Either the generated $G(x)$ or the real EM-induced voltage image y is fed into the discriminator D alternatively together with its corresponding workloads and aging time x as the condition input. The output of the discriminator is denoted as $D(G(x), x)$ or $D(y, x)$ depending on whether the generated or the real EM-induced voltage image was inputted. In the training process, we use the Wasserstein Distance [39] as the measurement of the difference between the real and the generated EM-induced voltage image distribution to take advantage of higher stability and convergence possibility.

Note that image-based DNN IR drop analysis like *GridNet* and *IREDGe* [36] are very scalable. In general, the power grid meshes on a chip are very sparse. When the chip gets bigger we can select different pixel resolution for the layout images. We can also leverage existing highly efficient GPU-based computation framework to train GAN or CNN models for large images.

E. Fast sensitivity calculation using the automatic differentiation in DNNs

One important observation for all the deep neural networks including the GAN model is that they are all differentiable with respect to the model weights. Thus, training can be performed through the automatic differentiation scheme, specifically the back-propagation algorithm, with sensitivity/gradient information.

In this work, we leverage the automatic differentiation to compute the sensitivity information between the output and all of the input resistance through *GridNet*. To be specific, we can compute the partial derivatives of one output voltage map with respect to every input resistance in one back-propagation of the generator DNN network. It is exactly the same technique employed in the training process, and the cost of usingthe Tensorflow *tf.gradients()* API is the same as one inference. The only difference is that the derivative is taken with respect to the input of the generator instead of the trainable variables in the model. In other words, one has to perform one inference using *GridNet* to compute sensitivity for b resistances for one output node. Our sensitivity calculation is similar to the adjoint network-based approach [40]. However, such method requires two simulations of *EMspice* for each output node. In our case, we do not require computing the sensitivities for all the output nodes, instead, we only focus on a few nodes that have IR drop violations, which makes the sensitivity computation much more efficient.

IV. New EM-induced voltage constrained optimization problem

A. Problem formulation

Let $G = \{N, B\}$ be a power grid network with n nodes $N = \{1, ..., n\}$ and b branches $B = \{1, ..., b\}$. Each branch i in B connects two nodes p and q with current flowing from p to q. l_i, w_i, and g_i are the length, width, and conductance of branch i, respectively. ρ is the sheet resistance. The width w_i of branch i is

$$w_i = \rho \frac{l_i}{r_i} = \rho l_i g_i \tag{5}$$

We remark that in the following notations, we only consider power networks for the sake of simplicity. The formulation of ground networks can be easily obtained for the same optimization framework.

1) Objective function: We can express the total routing area of the power grid network in terms of sheet resistance, branch length, width, and conductance as follows

$$a = \sum_{i \in B} l_i w_i = \sum_{i \in B} \rho l_i^2 g_i \tag{6}$$

The objective is to minimize the area of the power grid network. Assume that the topology and physical locations of the network are fixed, ρl_i^2 will become a constant and can be expressed as α_i, then the objective function is simplified as

$$a = \sum_{i \in B} \alpha_i g_i \tag{7}$$

2) Constraints: The constraints that need to be satisfied for a reliable power grid network are shown as follows.

1. *EM-induced voltage drop constraints:* When a void is nucleated and the interconnect enters into the growth phase, an increase over time in branch resistance will happen and may lead to time-varying node voltages.
   Suppose $v_{j,t}$ is the node voltage of the leaf node j at aging time t, which is a nonlinear function of conductances, the voltage drop is limited by a constant
   $$v_{dd} - v_{j,t} \le u \tag{8}$$
   where v_{dd} is the supply voltage and u is the bound of the IR drop. In real design, normally a voltage drop of less than 10% v_{dd} is acceptable.
2. *Minimum width constraints:* Usually, different layers have different requirements for the width of the metal wires
   $$w_i \ge w_{i,min} \tag{9}$$
   where $w_{i,min}$ is the minimum metal line width.
   According to Eq. (5), the above equation can be rewritten as
   $$g_i \ge \frac{w_{min}}{\rho l_i} \tag{10}$$
3. *Kirchhoff's current law (KCL):* We express Kirchhoff's current law in terms of node voltages
   $$\sum_{(j,k)} (v_k - v_j) g_{jk} = i_j \tag{11}$$
   where i_j is the current demand at node j and each k indicates a neighboring node of node j. In our approach, we view node voltages as functions of conductance, so it is implicitly satisfied.

B. Penalty method

The power grid optimization aims to minimize objective function (7) subject to constraints (8) and (9). It will be referred as problem *P*. Problem *P* is a constrained nonlinear optimization problem.

The penalty method is adopted to solve problem P. By adding a penalty term to the objective function that prescribes a high cost for the constraint violations, the original constrained problem is approximated with a sequence of unconstrained problems.

1) Penalty function formulation: We adopt a penalty function as follows

$$f = a + p_t = a + \beta \cdot \sum_j c_{j,t}^2 \tag{12}$$

where a is the network area of function (7), p_t is the penalty term and β is the penalty parameter. For the voltage drop constraint violation

$$c_{j,t} = \begin{cases} 0, & \text{if } v_{j,t} \ge v_{dd} - u \\ v_{j,t} - (v_{dd} - u), & \text{else} \end{cases} \tag{13}$$

Eq. (13) is further simplied as

$$c_{j,t} = v_{j,t} - (v_{dd} - u), \text{for all } j \in E_{vdrop} \tag{14}$$

where E_{vdrop} represents a set of indexes of the nodes that violate voltage drop constraint in the power grid network.

Minimum width constraints are not added into penalty function (12), the reason is that the proposed algorithm simply sets the branches that do not satisfy minimum width constraints with the minimum metal line width. The original constrained problem P is transformed to the problem of minimizing the penalty function (12) with minimum width constraints (9).

Moudallal *et al.* [11] observed that the IR drop $v_{dd} - v_{j,t}$ is a monotonically increasing function with respect to time, in other words, $v_{j,t_1} \ge v_{j,t_2}$ for $0 \le t_1 \le t_2$. Although branch resistance increase does not necessarily lead to an IR drop increase, this assumption holds in most cases. With this, we restrict our attention to the target aging time T, then Eq. (14) becomes

$$c_{j,T} = v_{j,T} - (v_{dd} - u), \text{for all } j \in E_{vdrop} \tag{15}$$

2) Optimization scheme: We first analyze the network for node voltages and branch currents while considering its aging time t and then identify the constraint violations. Generally, penalty method transforms the original constrained optimization problem into a sequence of unconstrained minimization problems. Back to our problem, the conjugate gradient method is adopted to update branch widths during each iteration, the process stops when all the constraints are satisfied. The solution procedure can be described as follows.

1. Obtain the initial conductance vector $G^{(0)}$, set an initial value of penalty parameter β and error bound $\varepsilon_b > 0$.
2. Solve the unconstrained minimization problem (12), obtain the current conductance vector $G^{(k)}$.
3. If $p_t < \varepsilon_b$, then stop; else, update penalty parameter β, set $k = k + 1$, and go to step 2).

Note that penalty parameter β cannot be a constant because different power grids need different β. In addition, small β may result in overconsideration of the objective function while large β may lead to an ill-conditioning problem. If we set the ratio of penalty terms to objective function as a constant r, then we will get the initial β_0 and can start minimizing the penalty function. β is updated automatically in the next minimization iteration, i.e., $\beta_{k+1} = \beta_k \cdot r \cdot a / p_t$. The process continues until all the constraints are satisfied.

V. Optimization strategies

A. Conjugate gradient method

In the penalty method, the efficiency of solving unconstrained minimization dominates the execution time. The conjugate gradient method, which is a method between the steepest descent method and the Newton method, deflects the direction of the steepest descent method by adding to it a positive multiple of the direction used in the last step. This method only requires the first-order derivatives but overcomes the steepest descent method's shortcoming of slow convergence. At the same time, the method does not need to save and compute the second-order derivatives that are needed by the Newton method.

We notice that the conjugate gradient method has been used for the IR drop and current density constrained optimization [41] and for on-chip decap optimization as well [42]. The work in [41] shows that the gradient-based optimization method is more scalable than linear programming-based methods [8]. However, this method is still based on Black's EM model, which adds current density constraints for each wire segment. It cannot optimize the power grids with nucleated wires for a target lifetime. In our approach, a more complicated physics-based EM model is applied to solve the EM-induced IR drop optimization problem over the target lifetime, such problem involves extensive computation-intensive simulations of full-chip PDNs.

In this work, we utilize the Fletcher-Reeves (F-R) conjugate gradient method. The algorithm is shown as Algorithm 1.

Algorithm 1 Unconstrained power grid area minimization algorithm

```
Input: Current conductance vector $G$.
Output: New conductance vector $G$.
$k := 0$.
Set initial descent direction to negative direction of the gradient $P^{(k)} = -\nabla f\left(G^{(k)}\right)$.
/*F-R conjugate gradient method*/
repeat
    Line search to determine a nonnegative scalar $\lambda_{opt}^{(k)}$ that minimizes $f$.
    Update conductance vector $G^{(k+1)} = G^{(k)} + \lambda_k P^{(k)}$.
    Choose new descent direction $P^{(k+1)} = -\nabla f\left(G^{(k+1)}\right) + \frac{\left\|\nabla f\left(G^{(k+1)}\right)\right\|^2}{\left\|\nabla f\left(G^{(k)}\right)\right\|^2} P^{(k)}$.
    $k := k + 1$.
until $\left\|\nabla f\left(G^{(k)}\right)\right\| < \varepsilon_{FR}$
```

B. DNN-based fast EM-induced IR drop estimation

The conjugate gradient optimization framework requires the sensitivity of penalized objective with respect to wire conductance or width. It actually requires intensive full-chip coupled EM and voltage (IR drop) analysis using *EMspice* as we will show later. Such circuit-level multi-physics-based full-chip power grid simulations are very expensive and even prohibitive for large problem sizes.

In this work, we build machine learning-based models based on the physics-based simulation to accelerate the sensitivity calculation. Since we are seeking the task as an image transforming problem and GAN has already been proved to be successful in all kinds of image applications among different DNN candidates, we select to employ conditional GAN (*GridNet*) to estimate EM-induced voltage maps via a supervised learning process based on the physics-based simulation data from *EMspice*. Details of this CGAN architecture have already been introduced in Section III-D.

C. Gradient calculation for the objective function

In the first step of the F-R conjugate gradient method, we analyze the network and derive the node voltage and current flow. From Eq. (12), the partial differential of penalty function with respect to conductance can be expressed as

$$\frac{\partial f}{\partial g_i} = \frac{\partial a}{\partial g_i} + \frac{\partial p_t}{\partial g_i} \tag{16}$$

The first term of Eq. (16) is equal to the constant α_i and the second term can be expanded easily

$$\frac{\partial f}{\partial g_i} = \alpha_i + \beta \cdot \sum_j \frac{\partial v_{j,t}}{\partial g_i} \cdot 2 \cdot c_{j,t}, \text{for all } j \in E_{vdrop} \tag{17}$$

Since our main focus is to ensure that the EM-induced voltages at target time T do not have violations, it is enough to search for a solution that decreases voltage drops at time T.

$$\frac{\partial f}{\partial g_i} = \alpha_i + \beta \cdot \sum_j \frac{\partial v_{j,T}}{\partial g_i} \cdot 2 \cdot c_{j,T}, \text{for all } j \in E_{vdrop} \tag{18}$$

Thus, the gradient of penalty function f with respect to conductance vector G is

$$\nabla f\left(G\right) = \left[\frac{\partial f}{\partial g_1}, \frac{\partial f}{\partial g_2}, \ldots, \frac{\partial f}{\partial g_i}, \ldots, \frac{\partial f}{\partial g_b}\right]^T \tag{19}$$

D. Gradient calculation via merged adjoint network

Traditionally, the adjoint network method has been proposed to calculate the partial differential of the node voltages with respect to branch conductance [40]. The adjoint network method can compute the sensitivity of one node voltage with respect to all resistance or conductance, but the cost of computing the sensitivities for all the node voltages can be very high. Instead of solving all adjoint networks separately, the merged adjoint network method only needs to solve circuit equations twice to calculate the final gradient of the objective function [42]: one is for the original network and the other is for the merged adjoint network. In this work, we implement merged adjoint networks for performance comparison.

Let N and $N'\left(j\right)$ be the original network and the adjoint network, respectively. The two networks have the same topology and conductance values. By running *EMspice* simulator, we can easily obtain conductance matrices of N and $N'\left(j\right)$ at time T. The only difference between the two networks is that all the absorbing current of $N'\left(j\right)$ is set to zero except node j. Since *EMspice* also tells the node voltages for N at time T, we only have to build $B\left(j\right)$ to solve the branch voltages for $N'\left(j\right)$.

$$B\left(j\right) = \left[0, 0, \ldots, -1, 0, \ldots, 0\right]^T \tag{20}$$

Let $v_{i,T}$ and $v'_{i,T}$ denote branch i's voltage of N and $N'\left(j\right)$, the partial differential of node voltage v_j with respect to the conductance of branch i is computed by

$$\frac{\partial v_{j,T}}{\partial g_i} = v_{i,T} \times v'_{i,T} = \left(v_{p,T} - v_{q,T}\right) \times \left(v'_{p,T} - v'_{q,T}\right) \tag{21}$$

Then Eq. (18) becomes

$$\begin{aligned} \frac{\partial f}{\partial g_i} =& \alpha_i + 2 \cdot \beta \cdot \left(v_{p,T} - v_{q,T}\right) \\ & \times \left(\sum_j v'_{p,T}\left(j\right) c_{j,T} - \sum_j v'_{q,T}\left(j\right) c_{j,T}\right) \end{aligned} \tag{22}$$

Suppose $V'\left(j\right)$ is a vector formed by the node voltages of $N'\left(j\right)$, we have

$$v'_p\left(j\right) = C_p V'\left(j\right), v'_q\left(j\right) = C_q V'\left(j\right) \tag{23}$$

where $C_p = [0, 0, \ldots, 0, 1, 0, \ldots, 0]$ with 1 appears at index p, and $C_q = [0, 0, \ldots, 0, 1, 0, \ldots, 0]$ with 1 appears at index q.

Therefore, Eq. (22) can be rewritten as

$$\begin{aligned} \frac{\partial f}{\partial g_i} =& \alpha_i + 2 \cdot \beta \cdot (v_{p,T} - v_{q,T}) (C_p - C_q) \\ & \times \left(\sum_j c_{j,T} V'(j) \right) \end{aligned} \tag{24}$$

E. Fast gradient calculation via deep neural networks

As mentioned earlier, sensitivity computation by adjoint network methods based on the detailed multi-physics *EMspice* simulation is very computationally expensive. To mitigate this issue, we propose to use the DNN-based model for sensitivity computation.

The objective of problem *P* is to minimize the power grid area while ensuring that the functional modules work properly at the target EM aging time T. Note that Eq. (5) holds only before the interconnect enters into the growth phase. Once the growth phase starts, the resistance starts increasing as the current starts to flow through the more resistive barriers of the copper wire. In other words, the decrease in conductance g_i does not have an impact on the wire width w_i.

Let us add subscript time t to illustrate. Back to our EM-induced voltage drop constrained problem, the sensitivity value s we expect is $\partial v_{j,T}/\partial w_i$, which means the partial differential of the node voltages at aging time T with respect to the branch width. According to Eq. (18), what we need to calculate is $\partial v_{j,T}/\partial g_i$. Since the width does not change during the EM process, i.e., $w_{i,T} = w_{i,0}$, it indicates that g_i here should be $g_{i,0}$. The rationale behind this is that we have to update conductance matrix $G^{(k)}$ for the next iteration, and updating $G^{(k)}$ implies updating width $W^{(k)}$, however, only the initial $W^{(k)}$ can be modified.

In *EMspice*, the coupled EM and IR simulation undergoes complex stress evolution and the change of EM-induced voltage drop with respect to time is nonlinear. From initial time 0 to target time T, the resistance of branch i may increase or remain unchanged, while the width of branch i always unchanged. It is impossible to express those partial derivatives with equations. Therefore, by applying the above merged adjoint network method, we can easily get $\partial v_{j,t}/\partial g_{i,t}$, but cannot obtain $\partial v_{j,T}/\partial g_{i,0}$.

As presented in Section III-E, we leverage the automatic differentiation scheme in *GridNet* to compute the sensitivity information for *GridNetOpt*. Specifically, we assume that we have m violation nodes at time T whose node voltages are represented by v_j, $j \in \{1, .., m\}$. The CGAN model is able to give the estimated sensitivity values in milliseconds. Then we can compute the following partial sensitivity matrix $\mathbf{S}_{m \times b}$ easily

$$\mathbf{S}_{m \times b} = \begin{bmatrix} \frac{\partial v_{1,T}}{\partial g_{1,0}} & \frac{\partial v_{1,T}}{\partial g_{2,0}} & \cdots & \frac{\partial v_{1,T}}{\partial g_{b,0}} \\ \frac{\partial v_{2,T}}{\partial g_{1,0}} & \frac{\partial v_{2,T}}{\partial g_{2,0}} & \cdots & \frac{\partial v_{2,T}}{\partial g_{b,0}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial v_{m,T}}{\partial g_{1,0}} & \frac{\partial v_{m,T}}{\partial g_{2,0}} & \cdots & \frac{\partial v_{m,T}}{\partial g_{b,0}} \end{bmatrix} \tag{25}$$

More importantly, this automatic differentiation scheme is able to tell $\partial v_{j,T}/\partial r_{i,0}$ $(\partial v_{j,T}/\partial g_{i,0})$ directly, which is more reasonable to employ in our problem. With this, Eq. (18) via deep neural networks becomes

$$\frac{\partial f}{\partial g_i} = \alpha_i + \beta \cdot \sum_j \frac{\partial v_{j,T}}{\partial g_{i,0}} \cdot 2 \cdot c_{j,T}, \text{for all } j \in E_{vdrop} \tag{26}$$

VI. Experimental results and discussion

A. Experiment setup

The proposed EM-aware IR drop constrained power grid optimization is implemented in Python with the *TensorFlow* library. The experiments are carried out on a Linux server with 2 Xeon E5-2698v2 2.3GHz processors and Nvidia Titan X RTX GPU with 24 GB memory.

In order to validate our work, we start from the power grid of the Cortex-M0 DesignStart processor, which is a 32-bit processor that implements the ARMv6-M architecture and is placed and routed using ICC II with Synopsys 32/28nm Generic Library. The power grid of Cortex has two layers, and there are 1k nodes in total.

Power grid information obtained from ICC II is then fed into the power grid parser. The information includes but is not limited to structure, node location, wire layer, wire length, current source, voltage source, and resistance values. The netlist format extracted from the grids is consistent with IBM power grid benchmarks [43]. In order to obtain enough power grids with different EM conditions, we generate lots of IBM-format power grid networks so that different workloads with different EM conditions can be tested and verified.

We train our CGAN model using three different designs/topologies and the size of the trained model varies with the grid size. Each design has a dataset containing 12k samples (workloads and aging time, EM-induced IR drop). *Design 1* comes from Cortex-M0, *Design 2* and *Design 3* are self-synthesized power grids with a format similar to *Design 1.* As shown in Fig. 3(a), *Design 2* has 4k nodes, 128 interconnect trees and 4 external power supplies. *Design 3* is demonstrated in Fig. 3(b), and it has 16k nodes, 256 interconnect trees and 9 external power supplies. The maximum allowable IR drop is set to 10% V_{dd} and the target EM lifetime T is 10 years. For each workload, we collect the EM-induced IR drop results obtained by *EMspice* at 11 discrete aging time instants (0 to 10 years).

We randomly select 15% workloads for testing and the remaining 85% are assigned for the training set. Our training and test data are separated on design basis, which means that the designs in the test dataset were never seen by the model during the training process. This ensures that the results of testing reflect the generalizability of the model. We have to emphasize that the designs in test dataset are to some extent similar to the ones in training dataset, otherwise it is impossible for the model to generalize to these unseen designs. During the training phase, all samples are randomly permuted at the beginning of every epoch.

B. EM-induced IR drop prediction results

1) Accuracy: Once the *GridNet* model is trained, the generator is preserved and serves as the model for inference. The model can take any power grid workload for a certain topology as input and give the predicted EM-induced voltage at a specified aging year. The predicted results from *GridNet* are compared with the baseline, which are the simulation results from *EMspice*. To evaluate the estimation error, we employ the root-mean-square error (RMSE) as the metric. We

This article has been accepted for publication in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. This is the author's version which has not been fully edit
content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2022.3206397

evaluate our trained *GridNet* model on the testing set which was set aside during the training phase. The random generation process guarantees that there is no overlap between these two datasets. The details and results are shown in Table. I.

TABLE I
PREDICTION RESULTS OF DIFFERENT DESIGNS

circuit	# nodes	# voltage sources	V_{DD} (V)	RMSE (mV)
Design 1	1024	2	1.05	5.697
Design 2	4096	4	1.05	6.100
Design 3	16384	9	1.05	3.922

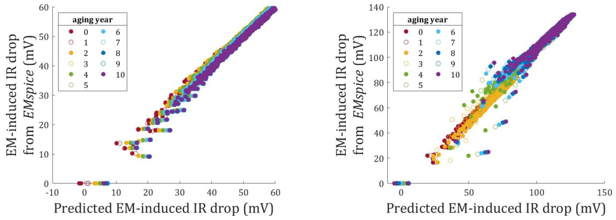
A total number of 1800 different workloads out of 12k workloads are tested for each design. For each workload, 11 voltage images at 0 to 10 discrete aging years are generated. As can be seen from Table I, comparing all 19800 generated EM-induced voltage images with the baseline on *Design 1*. *GridNet* achieves an average RMSE of 5.697mV, which represents about 0.57% error for a 1.05V power supply. The maximum RMSE is 16.48mV, which is 1.57% of the power supply. We observed that prediction from *GridNet* seems to be more accurate on larger IR drop values, the reason will be investigated in our future research.

PG-a and *PG-b* are different workloads, they are picked from *Design 1* to demonstrate different patterns. *PG-a* is a power grid with one mortal interconnect and the initial maximum IR drop is 58.75mV. After one year, the resistance of the mortal interconnect begins to increase due to EM aging and the value is changing over time. After 10 years, the maximum IR drop becomes 59.54mV, indicating the EM lifetime meets the 10-year target. In contrast, the predicted IR drop in the initial state and after 10 years are 57.93mV and 59.95mV, respectively. Fig. 5(a) presents the correlation between the predicted EM-induced IR drop and baseline from 0 year to 10 years, with a one-year interval, e.g., the purple dots indicate IR drop at 10 years. The errors of all predicted values are less than 11.42mV. The average error is 0.4035mV, with a standard deviation of 0.7525mV.

PG-b is a power grid with 6 mortal interconnects and its EM-lifetime is just 3 years. Initially, the real maximum IR drop is 84.47mV whereas the predicted maximum IR drop is 82.34mV. After 3 years, the EM-induced values become 84.58mV and 85.56mV for the baseline and predicted values, respectively. From the 4th year, wire resistance starts to increase, which has a large impact on the whole grid. As a result, both the baseline and predicted one have the maximum IR drop larger than 110.83mV, resulting in a power grid failure. Finally, in the 10th year, the baseline and the predicted IR drop value are 133.99mV and 127.39mV, respectively. From Fig. 5(a) and Fig. 5(b), the correlations for different years in the first figure have similar patterns. In contrast, the second figure looks different, the data for the first few years are concentrated in the lower part and the data for the last few years are distributed throughout the whole figure. The reason is that the EM effect is more clearly reflected in *PG-b*, which has a larger resistance increase.

The accuracy of the model on a new design is determined by the similarity between this new design and the ones that the model was trained on. If the accuracy is not acceptable, then it probably means that the model requires further re-training or fine-tuning.

2) Speed: To compare the EM-induced voltage analysis speed between *GridNet* and the baseline *EMspice*, we randomly pulled the designs from the training and testing set. The total computing time on the 500 different workloads from *Design 1* is 31.26h and 10.0s for *EMspice* and *GridNet*, respectively, indicating that about 11232 or $10^4\times$ speedup over *EMspice*. For *EMspice*, the time cost on the estimation of a single design varies from 0.57s to 427s depending on the EM immortality condition. For *GridNet*, however, the inference speed is steadily around 5ms for all the designs. The computing cost of *GridNet* is invariant to immortality conditions, which makes it much more suitable for larger-scale designs and leads to better scalability.



(a)

(b)

Fig. 5. Predicted IR drop versus the baseline of (a) PG-a; (b) PG-b.

As for larger designs, the speedup becomes more significant because the simulation time for *EMspice* grows considerably. For instance, obtaining the EM-induced IR drop results of some cases for *Design 3* at the 10th aging year takes more than 1.5h. If applying the proposed *GridNet*, the inference time will be around 10ms, indicating that the speedup will be more than 5×10^5. When training from scratch, the training loss of *Design 3* took around 68 hours to converge. If the model has to be extended to a new design variant, it just requires fine-tuning which should take much less time and the specific time cost depends on, again, the similarity of this new design and the ones in training dataset.

C. EM-aware IR drop constrained power grid optimization results

We further compare the proposed method to two methods, the conjugate gradient method based on adjoint network approach – *CG with merged adjoint network* and the sequence of linear programming based method proposed in [19] – *SLP*. In the SLP method, the optimization subjects to the multi-segment EM immortality constraint considering saturation volume of voids [44].

The power grid optimization results are shown in Table II. The power grids used in our experiments are selected randomly. In Table II, we try to list the cases that can cover different situations. *circuit* lists the power grid network benchmarks. *D1-PG1 - D1-PG4* have the same structure as the synthesized Cortex-M0 DesignStart processor (*Design 1*), each of them has 64 interconnect wires and thousands of nodes but they differ in wire resistance, length, width, and current sources. In contrast, *D3-PG7 - D3-PG9* come from the aforementioned *Design 3*, which have approximately 16k nodes and 256 interconnect wires. Therefore, the initial EM conditions of these benchmarks are different, such as the number of immortal wires. *Can't opt* means the tool cannot optimize the designs due to the presence of mortal wires.

In Table II, column 6 to 8 and 3 to 5 report the number of iterations (*# iter*), the reduced area ratio (*area reduced*) with respect to the original area and the total computation time (*time*) of *GridNetOpt* and the adjoint network method with *EMspice*, respectively. From the results shown in the table, the area from both methods can be reduced after optimization

TABLE II
GLOBAL OPTIMIZATION COMPARISON: COMPARISON BETWEEN PLAIN CG METHOD USING MERGED ADJOINT NETWORK [41] AND *GridNetOpt*

circuit	SLP [19]	CG with merged adjoint network [41]			*GridNetOpt*			
	area reduced (%)	# iter	area reduced (%)	time (s)	# iter	area reduced (%)	time (s)	speedup
D1-PG1	16.96	4	34.79	982.18	6	33.95	57.93	17.34
D1-PG2	20.07	5	35.65	1631.45	4	37.32	44.63	36.55
D1-PG3	can't opt	5	31.69	1632.23	5	29.52	47.61	34.29
D1-PG4	can't opt	6	14.25	2826.75	7	17.07	66.06	42.79
D2-PG5	can't opt	4	19.10	2078.06	6	19.19	57.48	36.15
D2-PG6	can't opt	4	15.15	1826.90	5	15.33	39.00	46.83
D3-PG7	can't opt	2	6.51	7806.35	2	9.96	21.72	359.37
D3-PG8	can't opt	2	5.91	12635.89	3	7.22	157.43	80.26
D3-PG9	can't opt	2	2.92	9621.64	4	4.29	52.74	182.45

TABLE III
THE COMPARISON OF PLAIN CG METHOD AND *GridNetOpt* ON *D1-PG1*

iteration	CG with merged adjoint network			*GridNetOpt*		
	time (s)	area (μm^2)	# failed node	time (s)	area (μm^2)	# failed node
1	473.94	0.5592	1003	13.78	0.5592	1014
2	179.28	0.6131	781	13.30	0.6130	967
3	172.94	0.6367	207	12.30	0.6309	901
4	156.03	0.6783	0	11.35	0.6494	773
5	finish	finish	finish	7.07	0.6681	343
6	N/A	N/A	N/A	0.13	0.6871	0

TABLE IV
THE COMPARISON OF PLAIN CG METHOD AND *GridNetOpt* ON *D3-PG8*

iteration	CG with merged adjoint network			*GridNetOpt*		
	time (s)	area (μm^2)	# failed node	time (s)	area (μm^2)	# failed node
1	7235.24	0.1672	743	145.29	0.1672	751
2	5400.65	0.1748	0	10.19	0.1722	43
3	finish	finish	finish	1.90	0.1724	0

and the area reduced ratio is similar. This demonstrates that our work achieves comparable optimization results compared to other conjugate gradient-based optimization works. Column 1 shows the optimization results from the saturation volume-based EM immortality constrained SLP method. Among the 9 test examples, only *D1-PG1* and *D1-PG2* are initially EM immortal, thus can be optimized through the SLP optimization within 2 iterations. In contrast, the other 7 examples contain mortal wires and cannot be performed successfully with this method. We notice that the reduced area ratio of the power grids from *Design 3* is not big, the reason is that the test cases we used are already well-designed, the optimization space left is not large enough.

With *GridNetOpt*, we are able to meet the power grid lifetime target much faster than using the adjoint network method with *EMspice*, which would be a great advantage especially when the optimization space is not that large because designers do not want to wait for a long time to only seek for a reduction potential. For example, in the *D1-PG4* case, the lifetime of the whole power grids is predicted to be greater than 10 years and the maximum voltage at T does not exceed $10\% V_{dd}$. There is 1 mortal wire and 44 violation nodes in total. Note that the lifetime definitions of individual interconnect wire and power grid network are different. The power grid lifetime refers to the earliest time t that EM-induced voltage violations of a power grid occur, here, we do not care about the earliest time t but cares about if there exist IR drop violations at target time T. By utilizing the by-product sensitivity information, we are able to get the optimization direction much easier as no complex numerical calculation is required. By iteratively solving the unconstrained minimization problem and updating the conductance vector and penalty parameter, the power grid meets the lifetime target after 7 iterations. Even though *GridNetOpt* achieves better area reduction than using the adjoint network approach for this case, the optimization time of the former is less. There is no obvious relationship between reduced area and the number of iterations of the two methods, e.g., *GridNetOpt* went through more iterations for the *D1-PG1* case.

We remark that comparison with *SLP* is not an apple to apple comparison as the two methods actually have different constraints as we explained earlier. Here we just show that SLP can't optimize many of the PDN circuits, which however can be optimized by the proposed method. Furthermore, note that we did not directly compare with [11]. The reason is that this method depends on properties in which wire width can only be sized up (increased). For the proposed method, we can start with any power grid network to optimize the wire widths to their best possible values (size up or size down). Further, this method essentially is an SLP-based method, existing

TABLE V
COMPARISON OF LOCALIZED FIXING WITH *GridNet* [1] AND GLOBAL OPTIMIZATION *GridNetOpt*

circuit	# mortal wires	# failed node	localized fixing with *GridNet* [1]		*GridNetOpt*	
			area increased (%)	time (s)	area increased (%)	time (s)
D1-PGL1	3	2	0.446	1.62	0.379	0.15
D1-PGL2	6	13	0.765	3.91	0.506	0.31
D2-PGL3	4	3	0.151	1.07	0.125	1.64
D2-PGL4	7	9	0.352	1.86	0.020	1.81
D3-PGL5	0	45	can't opt	can't opt	0.080	8.04
D3-PGL6	2	39	0.204	3.63	0.0996	8.66

work has shown that the conjugate gradient optimization method is much more scalable than linear programming-based methods [41]. In addition, we can extend our approach to statistical based optimization using Monte Carlo or other fast variational methods [45], which will be our future works.

Table III presents the detailed comparison on the *D1-PG1* case. The number of violation nodes comes from the *GridNet* CGAN model. In this circuit, the original area is $1.040\mu m^2$ and it is an EM immortal case. At the beginning of conjugate gradient-based optimization, the wire width is all set to its minimum, thus the area for optimization is $0.5592\mu m^2$. *GridNet* predicts that this circuit will have 1014 voltage violation nodes at the 10th aging year while *EMspice* simulates that it has 1003 voltage violation nodes. The conjugate gradient optimization with *GridNet* and merged adjoint network method undergo 6 and 4 iterations respectively to eliminate IR drop violations. Finally, *GridNetOpt* achieves 33.95% area reduction while *CG with merged adjoint network* makes 34.79% area reduction. However, the overall time of the latter is more than 17 times longer than the former. In contrast, EM immortality constrained SLP-based optimization only goes through 2 iterations, since the immortality constraint is more strict than the 10-year target lifetime and this method requires that all the branches within an interconnect tree have the same wire width, it only achieves 17.38% area reduction.

Table IV shows the comparison on the *D3-PG8* case, which is an EM mortal power grid with an original area of $0.1858\mu m^2$. With minimum width, the area becomes $0.1672\mu m^2$ and *GridNet* predicts that the number of nodes that violate the threshold voltage is 751. The first optimization iteration takes a relatively long time, after 3 iterations, all the voltage violations are eliminated. In contrast, due to the long simulation time of *EMspice*, the *CG with merged adjoint network* method takes 3.5 hours to finish the optimization process. As a result, we achieve about 80x speedup over existing CG-based approach.

Consider both area reduction and computation time, *GridNetOpt* gets similar area reduction but much better speedup (about **10x or more**) for all the cases, we can conclude that it outperforms the plain CG method using adjoint networks for EM-induced IR drop constrained power grid optimization problem.

D. Comparison of localized fixing with GridNet and global optimization GridNetOpt

Last but not least, we compare the proposed global optimization *GridNetOpt* with our previous work: localized fixing with *GridNet* [1].

For fair comparison, we set all branches to their minimum width and only allow a few failed nodes at target aging time T. The comparison results are shown in Table V. There are 6 test cases in total, and each design topology has 2 cases. Design *D1-PGL2* is a power grid with 6 mortal wires and its predicted lifetime is 7 years. At target aging time T, there are 13 voltage drop violations. *GridNetOpt* completes the optimization process in 1 iteration whereas the localized fixing method undergoes 2 iterations.

As we can see that *GridNetOpt* achieves better results in terms of area overhead for all the benchmarks than the localized fixing method, because the former can perform global optimization versus the localized fixing in [1]. As for computation time, the two methods are similar. The *D3-PGL6* case only has 2 mortal wires, the localized method is very efficient while the global method becomes more expensive when the chip size gets larger.

We note that design *D3-PGL5* has zero mortal wires but it has voltage violations at design time. After 10 years, the violation number is still 45. *GridNetOpt* is able to optimize the power grid in 1 iteration. However, the localized fixing method cannot perform the fixing as it needs to know vulnerable branches (mortal branches) to start from. Of course, one can find some local branches of the violating nodes to size, but it is not relevant to the EM-induced IR drop optimization.

VII. CONCLUSION

In this paper, we proposed a novel optimization framework, called *GridNetOpt*, for on-chip power distribution networks considering EM-induced IR drop constraints at the target aging time. *GridNetOpt* employs a conjugate gradient-based approach to size the wire segments, which is capable to consider all the EM failure situations, including immortal wires and mortal wires, for EM-aware power grid area optimization with a target lifetime. The optimization framework is further empowered by the data-driven learning-based time-varying IR drop modeling using deep neural networks. The new method can naturally leverage the differentiable feature of deep neural networks for fast sensitivity computation of node voltage with respect to wire resistance or width. Numerical results on a number of synthesized power grid benchmarks from ARM core CPU designs show that the proposed *GridNetOpt* can lead to an order of magnitude or more speedup over the conjugate gradient-based method using the traditional adjoint network approach. Compared to the localized power grid fixing with *GridNet*, *GridNetOpt* can lead to smaller area overhead for all the benchmarks we tested. It can also reduce IR drops for power grid circuits with immortal wires, which is not possible with the localized *GridNet* method.

References

[1] H. Zhou, W. Jin, and S. X.-D. Tan, "GridNet: Fast Data-Driven EM-Induced IR Drop Prediction and Localized Fixing for On-Chip Power Grid Networks," in *Proceedings of the 39th International Conference on Computer-Aided Design*, ser. ICCAD '20, Nov. 2020, pp. 1–9.

[2] "IEEE International Roadmap for Devices and Systems (IRDS)," 2020, http://irds.ieee.org/.

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

[4] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Boston, MA: Addison-Wesley, 1990.

[5] C.-K. Hu, M. B. Small, and P. S. Ho, "Electromigration in Al(Cu) two-level structures: Effect of Cu and kinetics of damage formation," *Journal of Applied Physics*, vol. 74, no. 2, pp. 969–978, Jul. 1993.

[6] S. Chowdhury and M. A. Breuer, "Optimum Design of IC Power/Ground Nets Subject to Reliability Constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, no. 7, pp. 787–796, Jul. 1988.

[7] R. Dutta and M. Marek-Sadowska, "Automatic Sizing of Power/Ground (P/G) Networks in VLSI," in *Proceedings of the 26th Design Automation Conference*, ser. DAC '89, Jun. 1989, pp. 783–786.

[8] X.-D. Tan, C.-J. R. Shi, D. Lungeanu, J.-C. Lee, and L.-P. Yuan, "Reliability-Constrained Area Optimization of VLSI Power/Ground Networks Via Sequence of Linear Programmings," in *Proceedings of the 36th Design Automation Conference*, ser. DAC '99, Jun. 1999, pp. 78–83.

[9] K. Wang and M. Marek-Sadowska, "On-Chip Power-Supply Network Optimization Using Multigrid-Based Technique," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, pp. 407–417, Mar. 2005.

[10] H. Zhou, Z. Sun, S. Sadiqbatcha, N. Chang, and S. X.-D. Tan, "EM-Aware and Lifetime-Constrained Optimization for Multisegment Power Grid Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 4, pp. 940–953, Apr. 2019.

[11] Z. Moudallal, V. Sukharev, and F. N. Najm, "Power Grid Fixing for Electromigration-induced Voltage Failures," in *Proceedings of the 38th International Conference on Computer-Aided Design*, ser. ICCAD '19, Nov. 2019, pp. 1–8.

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

[13] V. Sukharev, A. Kteyan, E. Zschech, and W. D. Nix, "Microstructure Effect on EM-Induced Degradations in Dual Inlaid Copper Interconnects," *IEEE Transactions on Device and Materials Reliability*, vol. 9, no. 1, pp. 87–97, Mar. 2009.

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

[15] 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.

[16] V. Mishra and S. S. Sapatnekar, "Predicting Electromigration Mortality Under Temperature and Product Lifetime Specifications," in *Proceedings of the 53rd Design Automation Conference*, ser. DAC '16, Jun. 2016, pp. 1–6.

[17] S. X.-D. Tan, H. Amrouch, T. Kim, Z. Sun, C. Cook, and J. Henkel, "Recent advances in EM and BTI induced reliability modeling, analysis and optimization," *Integration, the VLSI Journal*, vol. 60, pp. 132—-152, Jan. 2018.

[18] S. Tan, M. Tahoori, T. Kim, S. Wang, and Z. Sun, *Long-Term Reliability of Nanometer VLSI Systems: Modeling, Analysis and Optimization*. New York, NY: Springer, 2019.

[19] H. Zhou, S. Yu, Z. Sun, and S. X.-D. Tan, "Reliable Power Grid Network Design Framework Considering EM Immortalities for Multi-Segment Wires," in *Proceedings of the 25th Asia and South Pacific Design Automation Conference*, ser. ASP-DAC '20, Jan. 2020, pp. 74–79.

[20] N. Chang, A. Baranwal, H. Zhuang, M.-C. Shih, R. Rajan, Y. Jia, H.-L. Liao, Y.-S. Li, T. Ku, and R. Lin, "Machine Learning based Generic Violation Waiver System with Application on Electromigration Sign-off," in *Proceedings of the 23rd Asia and South Pacific Design Automation Conference*, ser. ASP-DAC '18, Jan. 2018, pp. 416–421.

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

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

[23] W. Ye, M. B. Alawieh, Y. Lin, and D. Z. Pan, "LithoGAN: End-to-End Lithography Modeling with Generative Adversarial Networks," in *Proceedings of the 56th Design Automation Conference*, ser. DAC '19, Jun. 2019, pp. 1–6.

[24] M. B. Alawieh, Y. Lin, Z. Zhang, M. Li, Q. Huang, and D. Z. Pan, "GAN-SRAF: Sub-Resolution Assist Feature Generation Using Conditional Generative Adversarial Networks," in *Proceedings of the 56th Design Automation Conference*, ser. DAC '19, Jun. 2019, pp. 1–6.

[25] B. Xu, Y. Lin, X. Tang, S. Li, L. Shen, N. Sun, and D. Z. Pan, "WellGAN: Generative-Adversarial-Network-Guided Well Generation for Analog/Mixed-Signal Circuit Layout," in *Proceedings of the 56th Design Automation Conference*, ser. DAC '19, Jun. 2019, pp. 1–6.

[26] W. Jin, S. Sadiqbatcha, J. Zhang, and S. X.-D. Tan, "Full-Chip Thermal Map Estimation for Commercial Multi-Core CPUs with Generative Adversarial Learning," in *Proceedings of the 39th International Conference on Computer-Aided Design*, ser. ICCAD '20, Nov. 2020, pp. 1–9.

[27] 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 *Proceedings of the 38th International Conference on Computer Design*, ser. ICCD '20, Oct. 2020, pp. 296–303.

[28] Z. Sun, S. Yu, H. Zhou, Y. Liu, and S. X.-D. Tan, "EMSpice: Physics-Based Electromigration Check Using Coupled Electronic and Stress Simulation," *IEEE Transactions on Device and Materials Reliability*, vol. 20, no. 2, pp. 376–389, Jun. 2020.

[29] 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.

[30] "EMspice – Coupled EM-IR Analysis Tool for Full-Chip Power Grid EM Check and Sign-off," 2020, https://github.com/sheldonucr/EMspice.

[31] 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 Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 5, pp. 969–980, May 2018.

[32] S.-Y. Lin, Y.-C. Fang, Y.-C. Li, Y.-C. Liu, T.-S. Yang, S.-C. Lin, C.-M. Li, and E. J.-W. Fang, "IR Drop Prediction of ECO-Revised Circuits Using Machine Learning," in *Proceedings of the 36th VLSI Test Symposium*, ser. VTS '18, Apr. 2018, pp. 1–6.

[33] Y.-C. Fang, H.-Y. Lin, M.-Y. Sui, C.-M. Li, and E. J.-W. Fang, "Machine-learning-based Dynamic IR Drop Prediction for ECO," in *Proceedings of the 37th International Conference on Computer-Aided Design*, ser. ICCAD '18, Nov. 2018, pp. 1–7.

[34] C.-T. Ho and A. B. Kahng, "IncPIRD: Fast Learning-Based Prediction of Incremental IR Drop," in *Proceedings of the 38th International Conference on Computer-Aided Design*, ser. ICCAD '19, Nov. 2019, pp. 1–8.

[35] Z. Xie, H. Ren, B. Khailany, Y. Sheng, S. Santosh, J. Hu, and Y. Chen, "PowerNet: Transferable Dynamic IR Drop Estimation via Maximum Convolutional Neural Network," in *Proceedings of the 25th Asia and South Pacific Design Automation Conference*, ser. ASP-DAC '20, Jan. 2020, pp. 13–18.

[36] V. A. Chhabria, V. Ahuja, A. Prabhu, N. Patil, P. Jain, and S. S. Sapatnekar, "Thermal and IR Drop Analysis Using Convolutional Encoder-Decoder Networks," in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, ser. ASP-DAC '21, Jan. 2021, pp. 690–696.

[37] Z. Xie, H. Li, X. Xu, J. Hu, and Y. Chen, "Fast IR Drop Estimation with Machine Learning," in *Proceedings of the 39th International Conference on Computer-Aided Design*, ser. ICCAD '20, Nov. 2020, pp. 1–8.

[38] R. Fu, J. Chen, S. Zeng, Y. Zhuang, and A. Sudjianto, "Time Series Simulation by Conditional Generative Adversarial Net," *arXiv e-prints arXiv:1904.11419*, Apr. 2019.

[39] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv e-prints arXiv:1701.07875*, Dec. 2017.

[40] S. W. Director and R. A. Rohrer, "The Generalized Adjoint Network and Network Sensitivities," *IEEE Transactions on Circuit Theory*, vol. 16, no. 3, pp. 318–323, Aug. 1969.

[41] X. Wu, X. Hong, Y. Cai, Z. Luo, C.-K. Cheng, J. Gu, and W. Dai, "Area Minimization of Power Distribution Network Using Efficient Nonlinear Programming Techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 7, pp. 1086–1094, Jul. 2004.

[42] J. Fu, Z. Luo, X. Hong, Y. Cai, Z. Pan, and S. X.-D. Tan, "VLSI On-Chip Power/Ground Network Optimization Considering Decap Leakage Currents," in *Proceedings of the 10th Asia and South Pacific Design Automation Conference*, ser. ASP-DAC '05, Jan. 2005, pp. 735–738.

[43] S. R. Nassif, "Power Grid Analysis Benchmarks," in *Proceedings of the 13th Asia and South Pacific Design Automation Conference*, ser. ASP-DAC '08, Mar. 2008, pp. 376–381.

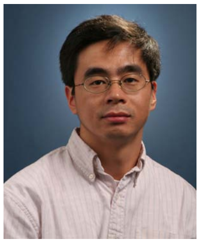
[44] Z. Sun, S. Sadiqbatcha, H. Zhao, and S. X.-D. Tan, "Saturation-Volume Estimation for Multisegment Copper Interconnect Wires," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 7, pp. 1666–1674, Jul. 2019.

[45] R. Shen, S. X.-D. Tan, and H. Yu, *Statistical Performance Analysis and Modeling Techniques for Nanometer VLSI Designs*. New York, NY: Springer, 2012.

Han Zhou received her B.Eng. and M.S. degrees in Electronic Science and Technology from Beijing Jiaotong University in 2013 and Beijing Institute of Technology in 2016, respectively and the Ph.D. degree in Electrical Engineering from the University of California at Riverside in 2021. Her research interests include VLSI reliability effects modeling, simulation and optimization, power and rail analysis, and applied machine learning in in-design place-and-route.

Yibo Liu received the B.S. degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2017, and the M.S. degree in computer engineering from the University of California at Riverside, Riverside, CA, USA, in 2019, where he is currently pursuing the Ph.D. degree. His current research interests include machine-learning accelerated power grid optimization.

Wentian Jin received his B.S. and M.S. degrees in Instrument Science and Technology in 2014 and 2017 respectively from Shanghai Jiao Tong University, Shanghai, China. He is currently a student researcher at the VLSI Systems and Computation Lab (VSCLab) and is pursuing the Ph.D. with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA, USA. His current research interests are electronic design automation (EDA) and applied machine learning in VLSI reliability analysis.



Sheldon X.-D. Tan (S'96-M'99-SM'06) received his B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China in 1992 and 1995, respectively and the Ph.D. degree in electrical and computer engineering from the University of Iowa, Iowa City, in 1999. He is a Professor in the Department of Electrical Engineering, University of California, Riverside, CA. He also is a cooperative faculty member in the Department of Computer Science and Engineering at UCR. His research interests include machine and deep learning for VLSI reliability modeling and optimization at circuit and system levels, machine learning for circuit and thermal simulation, thermal modeling, optimization and dynamic thermal management for many-core processors, efficient hardware for machine learning and AI, parallel computing and simulation based on GPU and multicore systems. He has published more than 320 technical papers and has co-authored 6 books on those areas.

Dr. Tan received NSF CAREER Award in 2004. He received Best Paper Awards from ICSICT'18, ASICON'17, ICCD'07, DAC'09. He also received the Honorable Mention Best Paper Award from SMACD'18. He was a Visiting Professor of Kyoto University as a JSPS Fellow from Dec. 2017 to Jan. 2018. He is serving as the TPC Chair for ASPDAC 2021, and the TPC Vice Chair for ASPDAC 2020. He is serving or served as Editor in Chief for Elsevier's Integration, the VLSI Journal, the Associate Editor for four journals: IEEE Transaction on VLSI Systems (TVLSI), ACM Transaction on Design Automation of Electronic Systems (TODAES), Elsevier's Microelectronics Reliability and MDPI Electronics, Microelectronics and Optoelectronics Section.