



Worst-Case Power Integrity Prediction Using Convolutional Neural Network

XIAO DONG, Zhejiang University, China

YUFEI CHEN, Zhejiang University, China

JUN CHEN, Giga Design Automation Co., Ltd, China

YUCHENG WANG, Giga Design Automation Co., Ltd, China

JII LI, Giga Design Automation Co., Ltd, China

TIANMING NI, Anhui Polytechnic University, China

ZHIGUO SHI, Zhejiang University, China

XUNZHAO YIN, Zhejiang University, China

CHENG ZHUO, Zhejiang University, China

Power integrity analysis is an essential step in PDN sign-off to ensure the performance and reliability of chips. However, with the growing PDN size and increasing scenarios to be validated, it becomes very time- and resource-consuming to conduct full-stack PDN simulation to check the power integrity for different test vectors. Recently, various works have proposed machine learning based methods for PDN power integrity prediction, many of which still suffer from large training overhead, inefficiency, or non-scalability. Thus, this paper proposed an efficient and scalable framework for the worst-case power integrity prediction, which can handle general tasks including dynamic noise prediction and bump current prediction. The framework first reduces the spatial and temporal redundancy in the PDN and input current vector, and then employs efficient feature extraction as well as a novel convolutional neural network architecture to predict the worst-case power integrity. Experimental results show that the proposed framework consistently outperforms the commercial tool and the state-of-the-art machine learning method with only 0.63-1.02% mean relative error and 25-69 \times speedup for noise prediction and 0.22-1.06% mean relative error and 24-64 \times speedup for bump current prediction.

CCS Concepts: • Hardware → Power and thermal analysis; • Computing methodologies → Machine learning.

Additional Key Words and Phrases: power distribution network, power integrity, sign-off, dynamic noise validation, bump current prediction, convolutional neural network

1 INTRODUCTION

With the continuous voltage scaling and ever growing integration density, power integrity has become a concerning issue in modern lower power SoC designs [1, 2, 14, 24, 32, 33]. The two major issues encountered with power

This work was supported in part by National Key R&D Program of China (Grant No. 2021ZD0114703), Zhejiang Provincial Key R&D program (Grant No. 2020C01052), National Natural Science Foundation of China (Grant No. 62034007, 62141404 and 62174001), Guangdong Provincial Key R&D program (Grant No. 2021B1101270003) and Anhui Provincial Natural Science Foundation (Grant No. 2208085J02).

Authors' addresses: X. Dong, Y. Chen, Z. Shi, X. Yin and C. Zhuo, Zhejiang University, Hangzhou, China; emails: {xdong, chenyufei, shizg, xzyin1, czhuo}@zju.edu.cn; J. Chen, Y. Wang and J. Li, Giga Design Automation Co., Ltd, Shenzhen, China; emails: {jchen, ycwang, jli}@giga-da.com; T. Ni, Anhui Polytechnic University, Wuhu, China; email: timmyni126@126.com.

Corresponding Author: Cheng Zhuo.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

1084-4309/2022/10-ART \$15.00

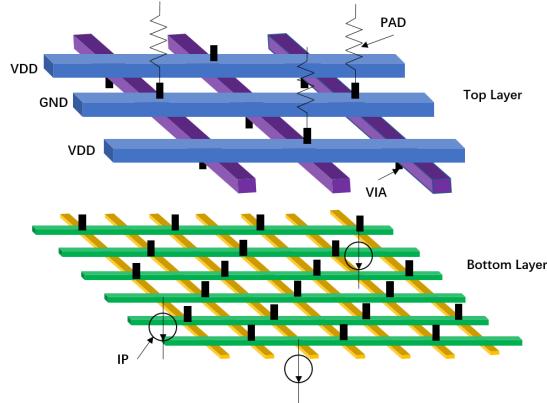


Fig. 1. An example of on-die power grid and current excitation.

integrity are power supply noise and electro-migration (EM) [11]. A too large supply noise and EM degradation in power distribution network (PDN) for an SoC not only degrades the performance of critical circuits, but also impairs its reliability [8, 11, 19]. Thus, worst-case power integrity analysis plays a significant role in VLSI design [3, 34]. In this paper, we mainly focus on worst-case dynamic PDN noise and bump current prediction, which is an essential step in PDN sign-off.

A modern VLSI PDN consists of board, package, and on-die power grid. Fig. 1 shows an illustrative plot of on-die power grid, which may contain more than ten metal layers with wire pitches up to several nano-meter for advanced technologies [8, 34]. Since the worst-case power integrity analysis needs to be accurate for sign-off, in commercial practices, it is commonly conducted with very detailed distributed RC/RLC model representing the on-die grid along with package and board modelled as macro-models [34]. Obviously, such a simulation needs to solve the voltages for millions to billions of nodes in the on-die grid, which is a non-trivial problem. It has been reported that, for 32nm commercial memory controller, even the commercial tool takes up to 24 hours to solve a few hundred nano-second trace when accounting for the entire PDN from board to die [34]. On the other hand, since there are many application scenarios for an SoC, power integrity analysis needs to be executed for tens of test vectors during sign-off, which *makes it very time- and resource-consuming* [1].

Mathematically speaking, given a test vector (e.g., in the form of switching current), worst-case noise estimation is to solve a sparse linear system (representing the entire PDN) and find the worst case noise both temporally (along the vector) and spatially (across the entire power grid). Then worst-case bump current is computed with the node voltages and branch parasitics. In the past two decades, many works have been proposed to speed up the sparse linear system solve, from multigrid, random walk, to graph spectral sparsification [22, 26, 27, 29, 31].

Recently, with the popularity of machine learning, a few works have deployed various machine learning techniques to accelerate the computation of supply noise or EM prediction without actually invoking the sparse linear system simulator [6, 9, 12, 13, 15, 17, 20, 21, 25, 28]. In order to achieve accurate prediction, the detailed PDN structural and electrical information are extracted as input features to train the model, such as instance power and path resistance [12, 13, 17, 25]. However, such instance-level information and path resistance itself demands power or static power delivery analysis, which actually involves implicit but *non-trivial training overhead*. In addition, while most existing works placed focus on static IR drop of PDN [6, 13, 21], *dynamic (or transient) PDN noise* is actually more important for sign-off, which is triggered by the resonance between package and die and hence results in more severe noise that needs to be validated. Finally, since the size of a commercial PDN can be tremendous (millions to billions of nodes), the direct employment of deep learning techniques to predict node voltage may easily result in a too huge neural network with serious *scalability issue* [6]. Thus, many works have

to either simplify the input to the neural network or check the noise region by region [12, 17, 25], inevitably compromising accuracy or efficiency. Moreover, few works pay attention to PDN power bump current prediction with machine learning method so far. While excessive bump current would cause local overheating, EM and even burnout, especially for modern designs in advanced technology node. Bump current needs to be validated in an efficient way with test vectors to ensure the reliability of PDN designs. Thus, it still *remains an open question to design an efficient and scalable framework to predict the worst-case dynamic noise and bump current for the entire PDN without incurring too much training overhead.*

To address the above issues, we propose an efficient, scalable yet accurate framework to handle general power integrity prediction tasks, including worst-case dynamic PDN noise prediction and power bump current prediction. Since the maximum noise and bump current caused by a test vector is more concerned in power integrity analysis, we do not waste time predicting how they change over time but only predict the maximum value in each tile caused by the input vector, which we called worst-case power integrity prediction, as introduced in [12]. The framework accounts for the redundancy both temporally and spatially. It then employs much simpler feature extraction strategy to reduce the training overhead. Finally, a novel convolutional neural network (CNN) architecture is designed to accurately incorporate all the information and predict the worst-case power integrity for the given test vector. The contributions of our work are summarized as below:

- We design a scalable and efficient framework with machine learning models for worst-case power integrity prediction. The framework consists of three subnets and can be trained with a small set of randomly produced test vectors. By only changing the last subnet, the trained model can support fast dynamic noise prediction or bump current prediction with the given test vector and hence speeds up the repeated validations of worst-case PDN sign-off.
- In order to accelerate the prediction, an algorithm for current feature compression is presented. By filtering out the irrelevant segments in the current sequences, the efficiency can be improved without sacrificing much accuracy.
- We propose to select the load current and distance to power bumps as the input features, which are easily accessible and provide sufficient information for accurate simulation.
- Our framework incorporates a novel CNN architecture to predict the dynamic worst-case PDN noise. The framework can predict the noise map of the entire PDN with just one-time execution, while many other frameworks demand repeated calculations.

Experimental results on four different commercial PDNs show that the proposed framework consistently outperforms the commercial simulator and another state-of-the-art CNN based prediction model, PowerNet [25]. The worst-case noise prediction model can achieve 0.63-1.02% mean relative error with less than 1mV mean absolute error, while it can achieve 25-69× speed up over the commercial tool. The detailed analysis on D4 clearly shows that the predicted noise distribution is very consistent with the ground-truth, which can identify almost all the hotspots. The worst-case bump current prediction model can achieve 0.22-1.06% mean relative error and 24-64× speed up over the commercial tool.

The remainder of this paper is organized as follows. In Section 2, existing works about PDN noise and bump current analysis are reviewed. Section 3 presents the proposed worst-case power integrity prediction framework. The experimental results are shown in Section 4, followed by conclusions in Section 5.

2 BACKGROUND

The core problem of PDN sign-off is to solve a sparse linear system after discretization, where interconnect parasitics and decoupling capacitors (decaps) constitute a symmetric and positive definite sparse matrix representing the entire power delivery system [22, 26, 27, 29, 31]. The instance switching draws currents from the power delivery system and is typically modelled as current sources, causing voltage droops in PDN. The PDN noise

sign-off can be further categorized to static and dynamic analyses [22, 26, 27, 29, 31]. Static analysis employs DC excitation and hence ignores the impact of capacitance or inductance in the extracted parasitics, which is basically to solve a series of linear equations. Dynamic analysis is fed with dynamically changing current sources and models the impact of decap or even inductance. In the commercial PDN noise sign-off tools, the dynamic analysis is converted to a series of static analyses, where the system matrix is the same but with different right-hand-side items.

For either static or dynamic noise analysis, the computational cost exponentially increases with the number of unknowns in the power delivery system [29]. When dealing with tens of test vectors for dynamic analysis, it is very time-consuming to complete all the vectors by deploying the conventional simulation based methods [22, 26, 27, 29, 31]. With the continuous development of artificial intelligence (AI), many researches are devoted to circuit design and simulation using machine learning [5, 7, 10, 18, 30]. Recently, various machine learning based algorithms have been proposed for static IR drop analysis [6, 13, 21]. For example, the work in [13] proposed an incremental IR drop prediction model, which feeds the structural and electrical features of PDN into XGBoost [4] to estimate the static IR drop. Besides, the work in [21] considers that reliability of the extracted features will be degraded for nodes close to the power grid boundary, then border information is included for static IR drop estimation to achieve better performance. However, the overhead to obtain cell-level features for training is not well discussed in both papers, which is non-trivial in practical deployment. While the two methods focus on voltage drop estimation for each node in the power grid, a fully convolutional network based model is used to generate the entire static IR drop map for PDN [6], whose scalability is concerning. The power distribution, package and layout information are converted into images as input features, so the static IR drop analysis is translated into an image-to-image translation task.

On the other hand, machine learning based dynamic noise analysis is also investigated in a few prior work [12, 17, 20, 25] using different machine learning models, from artificial neural network (ANN), XGBoost, to CNN and natural language processing (NLP). The work in [20] proposes to predict system-level power noise with ANN, CNN and NLP models, while most of works concentrate on estimating dynamic noise for cells or regions. For instance, the work in [17] extracts power and physical features of each cell to predict its dynamic noise with trained ANN model. To improve the model accuracy, [12] builds regional models for IR drop violation clusters and adds more features on the basis of [17]. Both of the works train models with cell-level features and estimate the voltage drop of a single cell each time, so it takes a long time to predict the dynamic noise of large scale circuits with millions of cells. Besides, another work [25] utilizes power, activity, toggling rate, and neighboring information, to predict the dynamic noise. However, to limit the underlying neural network size, it has to partition the entire network to tiles and compute the noise from tile to tile for each time stamp so that the maximum transient noise is obtained, which is also very time-consuming. In addition, few works have concentrated on efficient power bump current prediction so far, which is necessary to accelerate the bump current distribution analysis with large amount of vectors in sign-off. Therefore, it remains an open question for PDN bump current prediction with machine learning techniques.

3 PROPOSED METHOD

3.1 Overall Flow

When deploying CNN for PDN power integrity prediction, it is natural to consider using the instance current map as the input and output the voltage or current map for the nodes in PDN. However, the huge dimensions of both input and output make such network infeasible to commercial PDNs with millions of nodes. Take the worst-case noise prediction as an example, it is to ensure the worst-case noise (both spatially and temporally) to meet the pre-defined specification, i.e.,

$$\max_{i \in N} \max_{t \in S} v_i(t) \leq v_{spec} \quad (1)$$

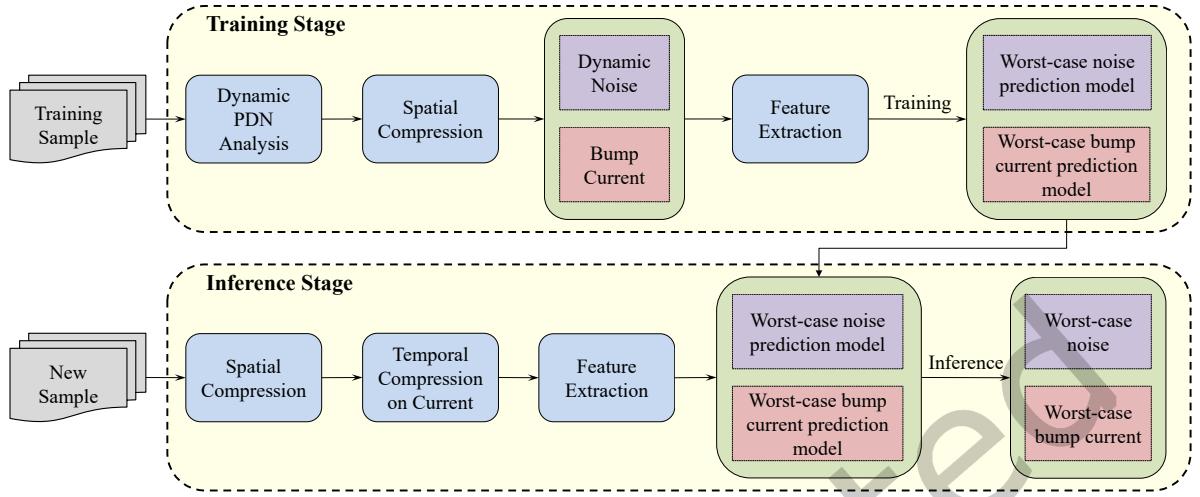


Fig. 2. Overall flow of proposed framework.

where \mathcal{N} is the set of all the nodes in a PDN and \mathcal{S} refers to the time-span of the simulated trace. Obviously, worst-case noise prediction targets at the noise of a particular node at a particular time. It is then unnecessary to explicitly compute the noises of the rest of the nodes at every time stamp, many of which are far lower than the worst-case noise, but demand significant computational time. Thus, it is desired to filter out the unnecessary computations both spatially and temporally. In the same way, we only consider the worst-case current for each power bump.

Based on such intuitions, we designed a worst-case power integrity prediction framework as shown in Fig. 2. The training procedure employs a commercial PDN sign-off tool to obtain the ground-truth and train the model. To improve the efficiency of prediction, spatial compression is employed so that the input and output dimensions are reduced while ensuring accuracy. Then, a feature extraction step is included to fuse different features. Instead of using instance-level fine-grained feature that needs additional extraction, we just use the same current vector information as the input to the commercial tool and physical information. Such feature selection is found to greatly reduce the training overhead. Then a CNN structure is designed to conduct the worst-case power integrity prediction. It is noted that the CNN structure consists of three subnets and the trained model can support worst-case noise prediction or bump current prediction by only changing the last subnet. The inference flow first employs a spatial and temporal compression step to remove redundancy in both input test vector (current in this paper) and PDN. Then the extracted features are then sent to the trained machine learning model to predict the worst-case noise or bump current. Thanks to the proposed compression algorithm and original CNN architecture, a fast and accurate power integrity prediction method is achieved.

3.2 Spatial and Temporal Compression

Instead of computing the worst-case noise for every node, we would like to merge a few nodes into a tile T_j and then predict the worst case noise of the tile, where $T_j \in \mathcal{T}$ and \mathcal{T} is the set of all the tiles. In other words, we can always partition the PDN layout into an array of $m \times n$ tiles and have:

$$\max_{i \in \mathcal{N}} \max_{t \in \mathcal{S}} v_i(t) = \max_{T_j \in \mathcal{T}} \max_{t \in \mathcal{S}} [\max_{i \in T_j} v_i(t)] \quad (2)$$

Instead of predicting $v(t)$, we can just compute $\max_{i \in T_j} v_i(t)$, which is especially important for larger PDN. Then we reduce the input and output dimensions from millions to $m \times n$, which makes our framework scalable for

Algorithm 1 Temporal Compression on Current Vector

Input: Original current maps $\{I[k] \in \mathbb{R}^{m \times n} | k \in [1, N]\}$, compression rate $r \in (0, 1)$, rate step $\Delta r > 0$

Output: Compressed current maps C_I

```

1: Initialize  $d_{min} = \infty$ ,  $r_0 = 0$ ,  $C_I = \emptyset$ 
2: for each  $I[k]$  do
3:    $S[k] = \sum_{x=1}^m \sum_{y=1}^n I[k][x][y]$ 
4: end for
5:  $\mu_s = \frac{1}{N} \sum_{k=1}^N S[k]$ 
6:  $\sigma_s = \sqrt{\frac{\sum_{k=1}^N (S[k] - \mu_s)^2}{N}}$ 
7:  $\{A[i] | i \in [1, N]\} = \text{argsort}(\{S[k] | k \in [1, N]\})$  // Sort  $\{S[k]\}$  in ascending order and return the index  $\{A[i]\}$ 
8: while  $r_0 \leq r$  do
9:    $C = \emptyset$ 
10:  for each  $p \in [1, r_0 * N] \cup [(1 - r + r_0) * N, N]$  do
11:     $C = C \cup \{S[A[p]]\}$ 
12:  end for
13:   $\mu_c = \frac{1}{r * N} \sum_{S[k] \in C} S[k]$ 
14:   $\sigma_c = \sqrt{\frac{\sum_{S[k] \in C} (S[k] - \mu_c)^2}{r * N}}$ 
15:  if  $|(\mu_s + 3\sigma_s) - (\mu_c + 3\sigma_c)| < d_{min}$  then
16:     $d_{min} = |(\mu_s + 3\sigma_s) - (\mu_c + 3\sigma_c)|$ 
17:     $r_s = r_0$ 
18:  end if
19:   $r_0 = r_0 + \Delta r$ 
20: end while
21: for each  $q \in [1, r_s * N] \cup [(1 - r + r_s) * N, N]$  do
22:    $C_I = C_I \cup \{I[A[q]]\}$ 
23: end for

```

huge scale circuits. For worst-case bump current prediction, spatial compression on input is also employed to reduce the computational cost.

For test vector, the sampled current map across the entire PDN can vary significantly over time. It is noted that steady state (with steady current) typically does not contribute to the worst-case noise. When the instance is heavily switching, it is more possible to inject the worst-case noise. While large load current often leads to worst-case bump current. Thus, it is necessary to filter out unimportant segments to accelerate the inference. Algorithm 1 describes how to conduct temporal information compression in our framework, which basically acts as a classifier that retains the segments with large current or switching activity. Thus, the basic idea of the proposed algorithm for temporal compression is to remove the segments with moderate currents. According to the algorithm, we first calculate the total current at each time stamp t_k and obtain the sequence $\{S[k] \in \mathbb{R}^{m \times n} | k \in [1, N]\}$, where N is the total number of time stamps. Then, we sort $\{S[k]\}$ in an ascending order and compress the segments with moderate currents to have the compressed set $\{S'[k]\}$ with similar $\mu + 3\sigma$ to the original set. Then, we can have the compressed current maps $C_I = \{I'[j] \in \mathbb{R}^{m \times n} | j \in [1, r * N]\}$, where r is the compression rate. The efficiency of our prediction can be improved by such spatial and temporal compression without sacrificing much accuracy.

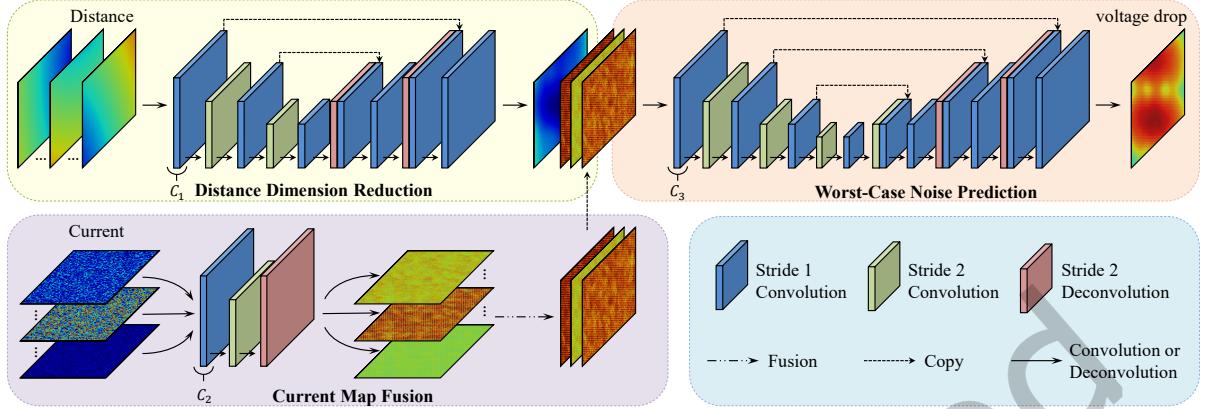


Fig. 3. The architecture of the proposed worst-case dynamic PDN noise prediction.

3.3 Feature Extraction

While detailed features of PDN may help improve model accuracy, too many features may also lead to information redundancy and model overfitting. Appropriate feature selection is not only important to the model inference efficiency but also training overhead. To ensure the machine learning model to learn effective knowledge from input features without much training overhead, we would like to use the expert knowledge of PDN dynamic analysis to facilitate the feature selection:

- **Load current:** Similar as the commercial tool that takes current activity as the input, we choose the load current as one of the features to model the excitation to the system. The load current is organized as a feature map so as to keep the correlation among neighboring instances. When spatially compressing the PDN layout, the instance currents within a tile is summed up to compute the load current.
- **Distance to power bumps:** If an instance is far from the power source, the current would go through a long path and induce a larger voltage droop. The distance from tiles to power bumps also reflects the distribution of the power bumps, which is associated with bump current. While such embedded information can be extracted by learning, we choose to explicitly take the information as input to simplify the following network size and ensure its good performance. We choose the center point of a tile as representation and then compute the Euclidean distance between the center point and all the power bumps. For a PDN with B power bumps, we are supposed to calculate the distance between each pair of tile and power bump, and then assemble a distance feature matrix $D \in \mathbb{R}^{B \times m \times n}$.

Unlike many prior works with instance-level features [12, 17], which demand additional simulations, the proposed feature extraction employs easily accessible features and is consistent with the conventional PDN sign-off flow. Thus, the proposed feature extraction can potentially help reduce training overhead as well as the following CNN architectures.

3.4 Worst-Case Dynamic PDN Noise Prediction

The architecture of the proposed worst-case dynamic PDN noise prediction model is shown in Fig. 3, which consists of three subnets to implement the distance feature processing, current map fusion, and voltage drop prediction. First, the distance feature matrix is fed into the first subnet to reduce the distance dimension. Then the current feature map is handled by a current map fusion subnet separately to obtain the fused map. Finally, the low-dimension distance map along with the fused current map is sent to the noise prediction subnet to estimate the worst-case dynamic noise. A fully convolutional neural network architecture is built as our model. The reason

why we choose CNN structure is that it has the ability to capture the spatial correlation among neighboring tiles in PDN, which is suitable for the extracted feature. On the other hand, unlike the commonly used machine learning structure of CNN followed by fully connected neural network, which restricts the input feature size, the fully convolutional network can adapt to inputs with different sizes and produce results with the same size of inputs at one time, making our model scalable and efficient.

3.4.1 Distance Dimension Reduction. In general, there are many power bumps in an SoC. For a particular tile, due to the locality [8], only a few of the power bumps have significant impact on its worst-case noise. Here we use a U-Net [23] like structure to achieve such reduction. The input distance map is first downsampled by convolutional layers and then upsampled by deconvolutional layers. Each of these two layers has the stride of 2 and is followed by a convolutional layer with the stride of 1. Moreover, skip connection is applied between the downsampling and upsampling features with the same size. Replication padding is applied in the convolutional layers and zero padding for the deconvolutional layers. Except the output layer, all the other layers adopt a ReLU activation function. The output layer has only one kernel so as to reduce the distance feature to $\tilde{D} \in \mathbb{R}^{m \times n}$.

3.4.2 Current Map Fusion. To learn the timing information in the compressed current vector, we design a current map fusion subnet. Each sampled current map is separately sent to the network, which can handle the vector with various lengths. An encoder-decoder structure is applied to the current map fusion subnet. As the input feature has only one channel, a small network with four layers is enough. For each tile, the subnet extracts three features, the maximum value of the peak current $\tilde{I}_{max} \in \mathbb{R}^{m \times n}$, the mean of maximum and minimum currents $\tilde{I}_{mean} \in \mathbb{R}^{m \times n}$, and $\mu + 3\sigma$ as the last feature $\tilde{I}_{msd} \in \mathbb{R}^{m \times n}$, where μ and σ refer to mean and standard deviation of the currents. The maximum currents feature is extracted based on the intuition that the higher current often causes larger voltage drops. However, it can be too pessimistic to only consider the peak currents. So we include the more moments of the currents to capture the other characteristics in the waveforms for more accurate prediction.

3.4.3 Dynamic PDN Noise Prediction. After distance dimension reduction and current map fusion, we obtain the distance map and fused current map. The distance map represents the capability of supplying currents from the external sources to the tiles, which models the responses of the system to the external supplies. On the other hand, the fused current map is used to model the response to the excitations. With the two maps, we are able to capture the system responses to both external supplies (sources) and current excitations (sinks). The \tilde{D} is concatenated with \tilde{I}_{max} , \tilde{I}_{mean} and \tilde{I}_{msd} into a matrix with the size of $4 \times m \times n$. Then it is sent to the worst-case noise prediction subnet. The dynamic PDN noise prediction subnet also has a U-Net like structure. The final output is the predicted PDN noise map $\hat{V} \in \mathbb{R}^{m \times n}$.

3.4.4 Model Training. To reduce the redundancy of the training set and improve efficiency of training, a training set expansion strategy is applied. If the distance between each existing sample and a candidate sample is larger than a pre-defined threshold, the candidate sample will be added to the training set. By controlling the threshold, the training set approximately accounts for 60% in the dataset. The rest samples are randomly split into validation and test set in a ratio of 3:7. We use Adam optimizer [16] to train the proposed model with learning rate of 0.0001. The L1 loss function is adopted during training and formulated as:

$$\mathcal{L}_{wn} = \sum_{i=1}^{m \times n} |v_i - \hat{v}_i| \quad (3)$$

where v_i and \hat{v}_i are the simulated and predicted worst-case dynamic noise of tile i , respectively.

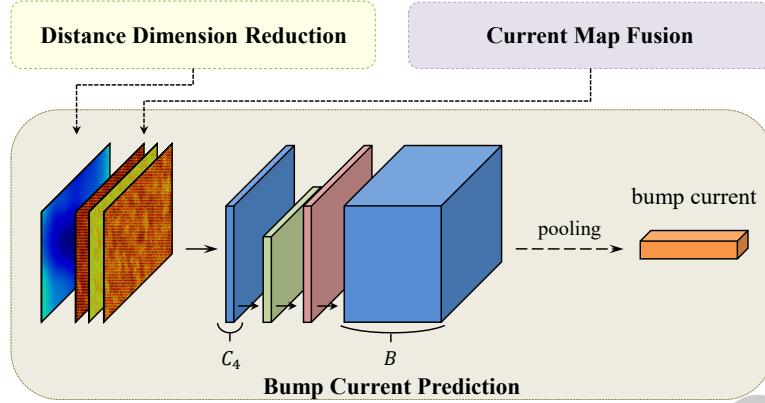


Fig. 4. The architecture of the proposed maximum bump current prediction.

3.5 Worst-case Bump Current Prediction

The architecture of the proposed worst-case bump current prediction model is shown in Fig. 4. It consists of three subnets, whose distance dimension reduction and current map fusion subnets are the same as that of the proposed worst-case dynamic noise prediction model. The low-dimension distance map and the fused current map obtained from the previous two subnets are sent to the bump current prediction subnet together to estimate the worst-case current of each power bump.

3.5.1 Bump Current Prediction. The bump current prediction subnet has a simple encoder-decoder structure, which contains three convolutional layers and one deconvolutional layer, followed by a pooling layer. The number of kernels of the last convolutional layer is the same as the number of power bumps, so the output feature map is $\tilde{F}_b \in \mathbb{R}^{B \times m \times n}$. Then, a global average pooling is applied to \tilde{F}_b to obtain the predicted worst-case bump current $\hat{I} \in \mathbb{R}^{B \times 1}$. The global average pooling calculates the mean value for each channel of feature map respectively and the result can be transposed into a vector. The reason why we choose the global average pooling rather than fully connected layers is that it is scalable for feature maps with various sizes and it has no parameters to optimize so the model is lightweight.

3.5.2 Model Training. The model is trained with the same data as the dynamic noise prediction model, except that the label is the worst-case bump current simulated by the commercial tool. We use Adam optimizer with learning rate of 0.0001 to train the model and the L1 loss function is adopted during training and formulated as:

$$\mathcal{L}_{wbc} = \sum_{j=1}^B |i_j - \hat{i}_j| \quad (4)$$

where i_j and \hat{i}_j are the simulated and predicted worst-case current of power bump j , respectively.

4 EXPERIMENTAL RESULTS

4.1 Experimental Setup

In our experiment, we employ four commercial PDN designs with different sizes, whose characteristics are summarized in Table 1 (denoted as D1 to D4), where #Node is the total number of the power grid and $\#I_{load}$ is the number of current loads (both are reported by the commercial tool). The reported Mean and Max worst-case noise (denoted as WN in the table) across all the tiles show that the four designs have dramatically different tolerance to

Table 1. Characteristics of designs in experiment.

Design	#Node (M)	# I_{load} (k)	Mean WN (mV)	Max WN (mV)	Hotspot ratio	Mean WBC (A)	Max WBC (A)
D1	0.58	2.5	100.4	131.7	56.3%	0.58	2.20
D2	0.58	16.9	91.7	128.4	30.1%	0.60	2.02
D3	2.67	122.5	127.1	290.7	57.5%	2.31	3.88
D4	4.40	810	89.0	119.9	22.5%	0.76	2.26

Table 2. Comparison of accuracy and run-time between proposed worst-case noise prediction model and the commercial tool.

Design	$m \times n$	Accuracy evaluation			Runtime comparison			Hotspot Missing rate
		Mean AE/RE	99% AE/RE	Max AE/RE	Proposed (s)	Commercial (s)	Speedup	
D1	50 × 50	0.98mV/1.02%	3.21mV/3.66%	4.45mV/5.88%	2.93	76	26×	1.09%
D2	130 × 130	0.74mV/0.83%	2.35mV/2.89%	4.12mV/5.83%	2.72	69	25×	1.95%
D3	70 × 50	0.71mV/0.63%	3.00mV/3.47%	9.38mV/10.53%	4.60	320	69×	0.28%
D4	180 × 180	0.58mV/0.71%	2.90mV/4.20%	8.23mV/16.80%	8.95	621	69×	1.93%

noise, with maximum WN varying from 12% to 29% of the nominal supply voltage (=1V). Hotspot is the number of tiles whose worst-case noise exceeds the pre-defined 10% threshold of the nominal supply voltage. The Mean and Max worst-case bump current is denoted as WBC in the table. For each design, we randomly generate 500 groups of test vectors and run dynamic PDN noise simulation with the commercial tool to obtain the worst-case dynamic noise and bump current as ground-truth. Then the groups are split into training, validation and test set. Finally, we train a special model for each design with the training set and compare the prediction result with the ground-truth on test set. The training time of worst-case dynamic noise prediction model for D1 to D4 is 130 minutes, 48 minutes, 78 minutes and 185 minutes, respectively. And the training of worst-case bump current prediction model takes 128 minutes, 44 minutes, 77minutes and 176 minutes for D1 to D4 respectively. In the following experiments, we set the time step $\Delta t = 1\text{ps}$ and the number of kernels $C_1 = C_2 = C_4 = 8$ and $C_3 = 16$, where C_1 , C_2 , C_3 and C_4 are the number of kernels of all layers (excluding the output layer) in distance dimension reduction, current map fusion, worst-case noise prediction and bump current prediction subnets, respectively. The proposed framework was implemented in Python using PyTorch.

4.2 Noise Prediction Model Evaluation

Table 2 summarizes the prediction result of the proposed worst-case dynamic noise prediction model in comparison to the commercial tool. The accuracy of the proposed model is evaluated with mean absolute error (denoted as AE), mean relative error (denoted as RE), 99% percentile AE and RE, and maximum AE and RE. It is noted that we take the absolute value of the difference between the predicted and simulated results when calculating AE and RE. As shown in the table, the proposed model can achieve a mean AE less than 1mV and 0.63-1.02% mean RE. Moreover, even for 99% percentile, AEs are only 2-3mV with REs of 2.9-4.2%. Though D4 reports larger maximum RE of 16.8%, the corresponding AE is only around 8mV, whose tile has a small worst-case noise. On the other hand, we compare the run-time of the proposed model with the commercial tool. The former includes time of feature extraction and CNN forward propagation without temporal compression, while the latter is the dynamic noise simulation time. Comparison result shows that the run-time of the proposed model is much faster than the commercial tool, with 25-69× speed up.

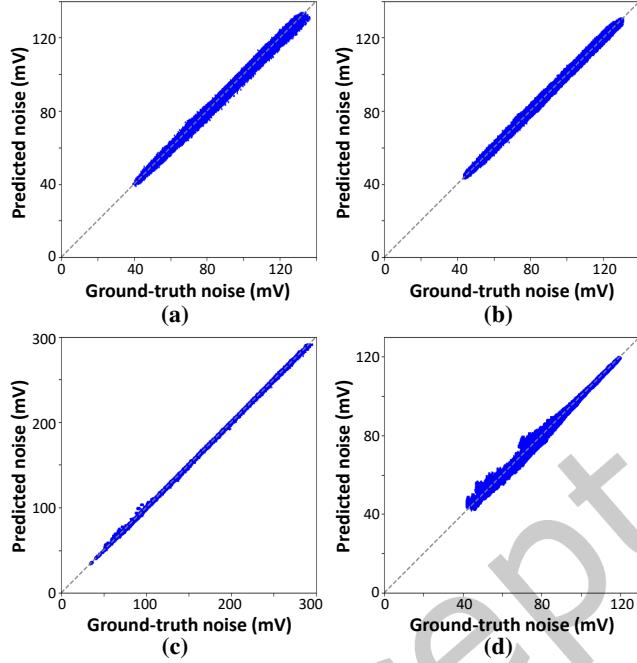


Fig. 5. Predicted noise versus ground-truth on: (a) D1; (b) D2; (c) D3; and (d) D4.

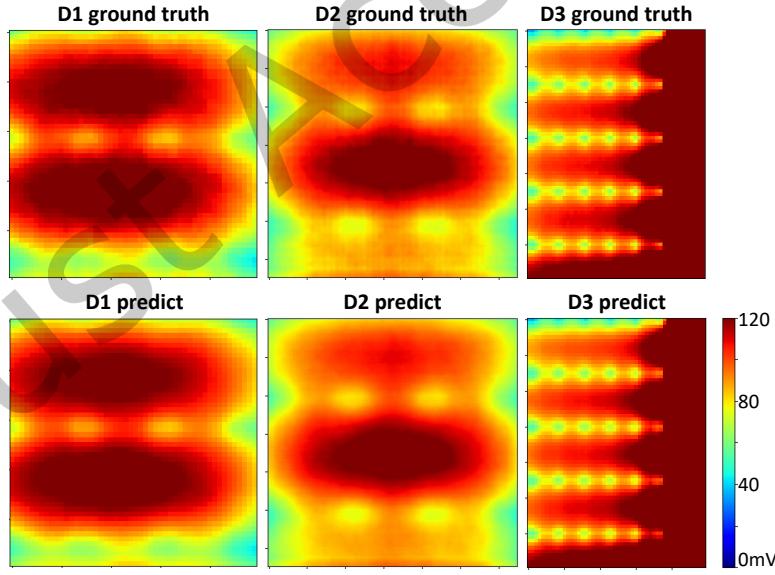


Fig. 6. Comparison between the ground-truth and predicted worst-case dynamic PDN noise map for D1-D3.

Fig. 5 shows the correlation between the predicted worst-case noise and ground-truth for D1-D4. The dashed line in diagonal indicates a perfect correlation. As shown in Fig. 5, most of the dots are distributed near the

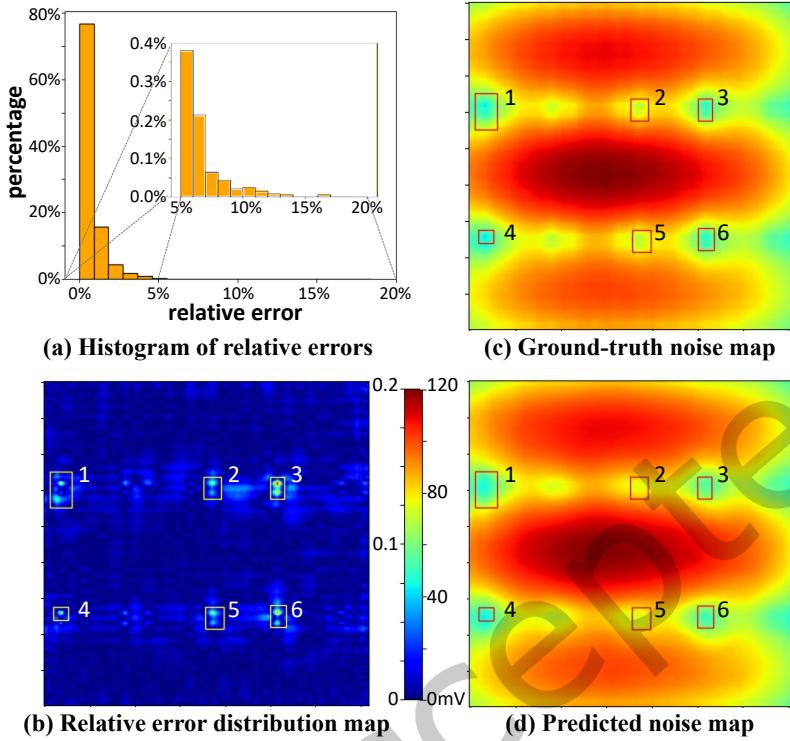


Fig. 7. Prediction results of D4: (a) Histogram of relative errors; (b) Relative error distribution map; (c) Ground-truth noise map; and (d) Predicted noise map.

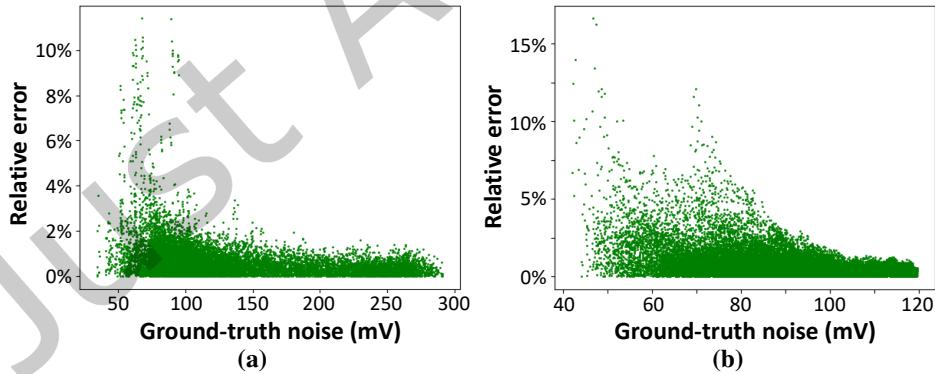


Fig. 8. The correlation between the ground-truth and prediction relative error for: (a) D3; and (b) D4.

dashed lines, which means the prediction result of the proposed model has a fairly high correlation with the simulation result of the commercial tool.

The predicted worst-case PDN noise distribution maps and their ground-truths for D1, D2 and D3 are shown in Fig. 6. The predicted results are almost identical to the ground-truths, which indicates that the proposed framework can effectively replace the commercial tool to support worst-case noise validation for different vectors.

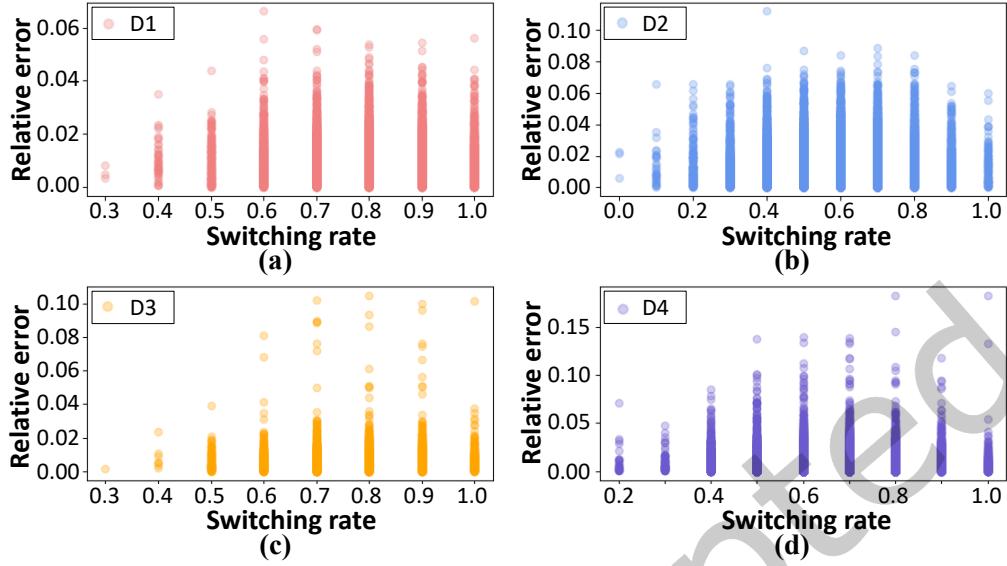


Fig. 9. The prediction relative errors under test vectors with various switching rates for: (a) D1; (b) D2; (c) D3; and (d) D4.

The detailed prediction results of D4 are demonstrated in Fig. 7. Fig. 7(a) shows the histogram of the relative errors for all the tiles in D4. It is clear that most of the tiles have relative errors of less than 5%. Only very few tiles have small worst-case noise and hence result in larger relative errors. Fig. 7(b) shows the distribution map of the relative error, where the marked spots are the tiles with larger relative errors. Fig. 7(c) and 7(d) present the comparison between the predicted and simulated voltage maps. Similar as Fig. 6, the two subfigures are almost identical. It is also observed that the corresponding marked spots have low noise, which are then not the concerned spots in PDN designs. Fig. 8 presents the relationship between ground-truths and prediction relative errors of tiles for D3 and D4. It shows more clearly that high relative error tiles often have low noise. Therefore, the proposed model is accurate enough for dynamic PDN noise analysis.

In general, the worst-case dynamic noise needs to be validated with tens of test vectors during sign-off. Thus, the proposed model should be able to handle the sign-off when test vectors with various switching rates are given. Fig. 9 shows the prediction relative errors of the proposed model under test vectors with different switching rates. The switching rate is defined as the number of switches from 0 to 1 and then to 0 over the number of clock cycles. As shown in the figure, there is no significant difference among the prediction relative errors of instances with different switching rates. Besides, for the instances with same switching rate, the prediction relative errors are quite low in most cases. So the proposed model could keep accurate.

To further verify the accuracy of the proposed worst-case noise prediction model, we evaluate its performance on hotspot detection. Thanks to the accuracy of the proposed model, almost all the hotspots for the four designs are correctly identified with merely 0.28-1.95% missing rate, as shown in the last column of the table 2. The ROC curves for the four designs are shown in Fig. 10. The area below the ROC curve is called AUC ($AUC \in [0, 1]$), a higher value of which means a better classification accuracy. Fig. 10 shows that the AUC for each cases is close to 1, which indicates that the proposed model is also good for hotspot detection.

We also compare the proposed model with PowerNet [25] on D4, which is a CNN based tool to predict dynamic PDN noise. The internal and leakage power, signal arrival time, and toggling rate are extracted from data and fed to PowerNet [25]. The model architecture we used to build PowerNet is exactly the same as that in [25]. We set

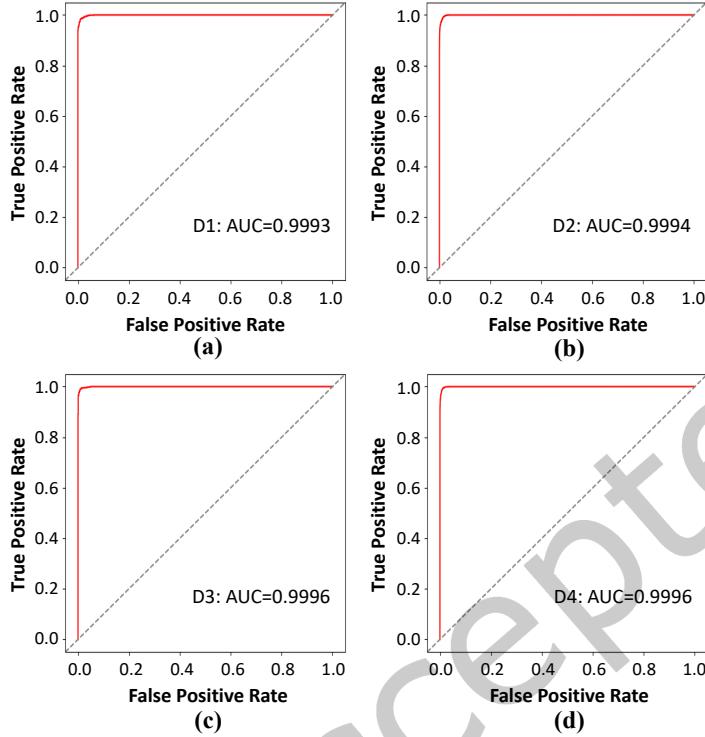


Fig. 10. The ROC curves for hotspot detection results of the proposed worst-case noise prediction model for: (a) D1; (b) D2; (c) D3; and (d) D4.

Table 3. Comparison between the proposed noise prediction model and PowerNet [25].

Model	MAE (mV)	Mean RE	Max RE	AUC	runtime (s)
PowerNet [25]	11.69	13.71%	42.08%	0.602	23.25
Ours	0.58	0.71%	16.80%	0.999	8.95

the number of time-decomposed power maps as 40, input window size as 15, and then partition the PDN design similarly (180×180 tiles) as the proposed model with tile size of 10um. L1 loss is adopted as loss function to train the model. PowerNet is trained with the same data as the proposed model for fair comparison. Table 3 shows that the proposed model outperforms PowerNet in both accuracy and run-time, with almost 20x improvement in mean AE and 3x speed up. By comparing AUC between the two, the proposed model can achieve almost 40% AUC improvement.

4.3 Effectiveness of Distance Feature

Unlike the PowerNet [25] that only focuses on instance activity, we also select distance to power bumps as the feature, which models the distribution of the power supplies in system. Although for a certain design, such information can be extracted by training the machine learning model, we take the advantage of choosing it as one

Table 4. The comparison between model trained with only current feature and original model.

Design	Model-oc		Original model	
	Mean AE/RE	Max AE/RE	Mean AE/RE	Max AE/RE
D1	1.55mV/1.66%	12.80mV/14.58%	0.98mV/1.02%	4.45mV/5.88%
D2	6.36mV/6.74%	27.98mV/42.14%	0.74mV/0.83%	4.12mV/5.83%
D3	1.04mV/0.95%	21.92mV/24.30%	0.71mV/0.63%	9.38mV/10.53%
D4	8.13mV/9.66%	48.47mV/104.19%	0.58mV/0.71%	8.23mV/16.80%

Table 5. Comparison of accuracy and run-time between proposed worst-case bump current prediction model and the commercial tool.

Design	Accuracy evaluation		Runtime comparison		
	Mean AE/RE	Max AE/RE	Proposed (s)	Commercial (s)	Speedup
D1	6.16mA/1.06%	35.00mA/3.01%	2.73	73	27×
D2	4.16mA/0.78%	22.92mA/2.10%	2.69	65	24×
D3	7.19mA/0.33%	21.58mA/1.10%	4.50	286	64×
D4	0.99mA/0.22%	4.57mA/1.44%	8.88	537	60×

of the features. To validate the effectiveness of the extracted distance feature, we remove the distance dimension reduction subnet in the worst-case noise prediction model and train the model (denoted as model-oc) with only current feature. The model-oc is trained with the same data and hyperparameters as the original model for fair comparison. The comparison of accuracy between model-oc and original model on D1-D4 is shown in Table 4. The prediction errors of model-oc are larger than the original model for all cases, which indicates that adding distance to power bumps as feature can help improve the model accuracy. The reason is that model-oc has to learn power supplies information by itself, then a more complicated architecture and more training iterations are needed to obtain an accurate enough model. Therefore, the distance feature can effectively reduce the training overhead and simplify the model design.

4.4 Bump Current Prediction Model Evaluation

Table 5 summarizes the performance of the proposed worst-case bump current prediction model compared with the commercial tool. The proposed model achieves a mean AE less than 8mA and 0.22-1.06% mean RE. The maximum RE is only 1.10-3.01% among four designs, which indicates high accuracy of the proposed model. Besides, the run-time of the proposed model without temporal compression is compared with the commercial tool. As shown in the table, the proposed model is much more efficient than the commercial tool, which achieves 24-64× speed up.

The predicted worst-case bump current and their ground-truths are compared for D1 to D4, as shown in Fig. 11. Each pair of the dot and cross is almost overlapping, which shows that the predicted results are quite close to the ground-truths simulated by commercial tool. We further observe the histogram of absolute error for D1 and D2, which have relatively higher AE and RE than the others. As shown in Fig. 12, the absolute error is less than 5mV in most cases for both D1 and D2. Thus, the proposed framework is accurate enough to predict the worst-case currents of power bumps for different vectors.

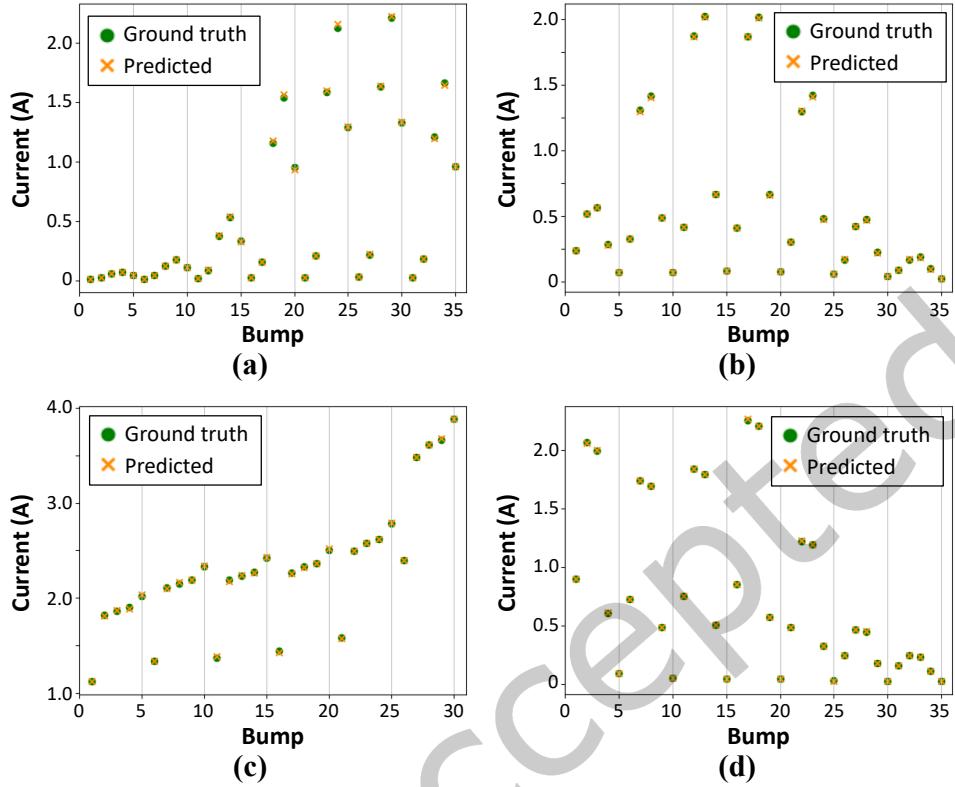


Fig. 11. Comparison between the ground-truth and predicted worst-case bump current for: (a) D1; (b) D2; (c) D3; and (d) D4.

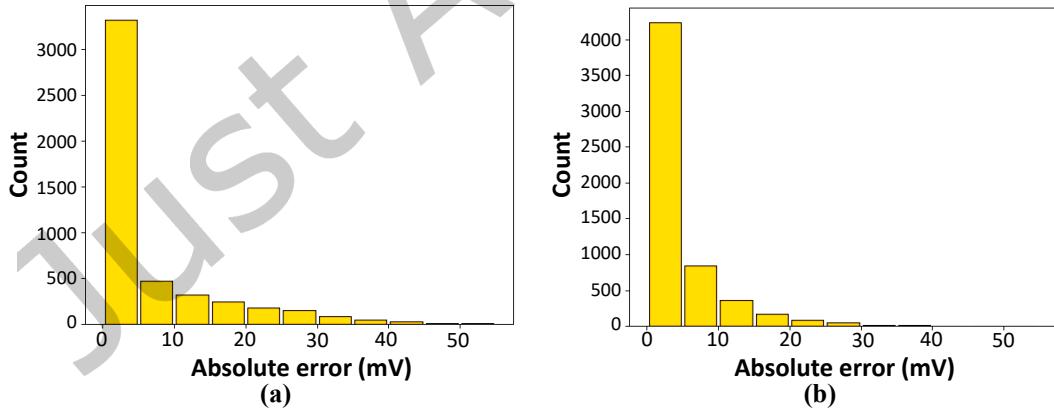


Fig. 12. Histogram of absolute error for: (a) D1; and (b) D2.

4.5 Spatial Compression

We compare the performance of the proposed worst-case noise prediction model on D3 with different number of tiles. Fig. 13 shows the prediction mean and maximum relative error, run-time and floating point of operations

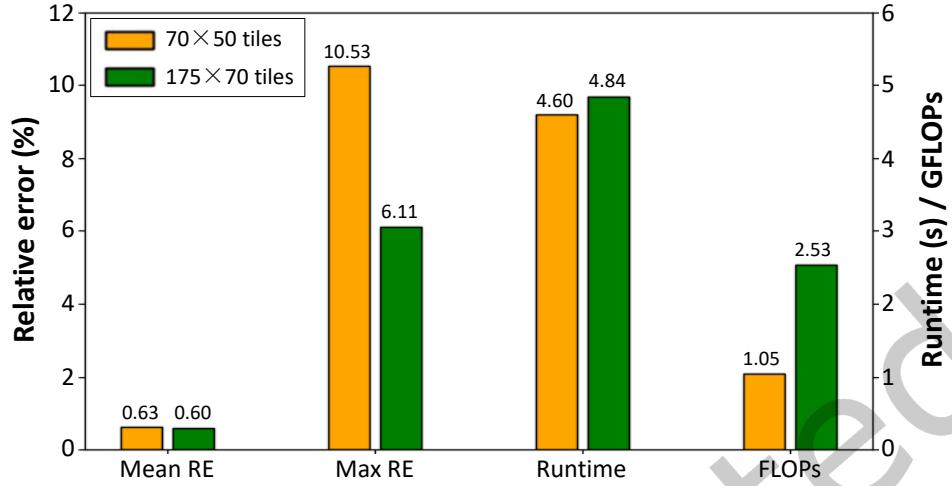


Fig. 13. Performance on D3 with different number of tiles.

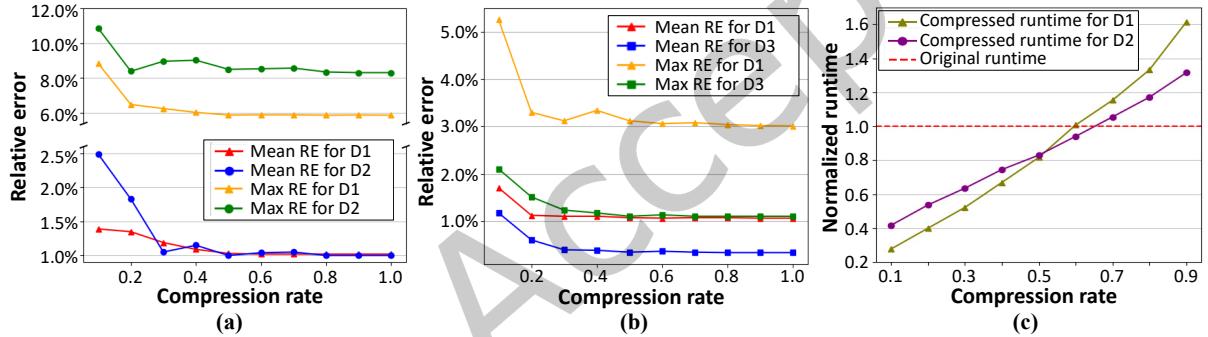


Fig. 14. Impact of the proposed temporal compression algorithm: (a) Noise prediction relative error change w.r.t. compression rate; (b) Current prediction relative error change w.r.t. compression rate; and (c) Noise prediction run-time v.s. compression rate.

(FLOPs) of two cases with 70×50 and 175×70 tiles respectively. The mean relative error is almost the same in both cases, while the maximum relative error is reduced from 10.53% to 6.11% and the run-time is increased slightly with larger number of tiles. Besides, the computational cost for the case with larger number of tiles is increased a lot. Therefore, partitioning the layout into more tiles can help reduce the prediction error but induce more time and computational cost. There is a trade-off between the accuracy and efficiency of the model, which guides us to select the appropriate number of tiles. It is noted that although maximum error is higher with small number of tiles, most of tiles have very small errors and the hotspot missing rate is low as shown in Table 2. As a result, choosing a small number of tiles is sufficient to provide good enough predictions in most scenarios.

4.6 Temporal Compression

Fig. 14 summarizes the performance of the proposed temporal compression algorithm. Fig. 14(a) and 14(b) show how the relative error changes with different compression rate for the proposed worst-case noise prediction model and bump current prediction model, respectively. Though the errors generally drops with a larger compression

rate, i.e., more data are retained, there is a knee point after which the degradation is much faster. For D1 and D2 in noise prediction, such a compression rate threshold is around 0.3, with mean relative errors of 1.19% and 1.05% for D1 and D2, respectively. For D1 and D3 in current prediction, the threshold is also around 0.3, with mean relative errors of 1.10% and 0.39% for D1 and D3, respectively. The noise prediction run-time comparison after using the proposed temporal compression is shown in Fig. 14(c). With a higher compression rate, the run-time is almost linearly increased. Due to the multiple iterations in the compression algorithm, the proposed model with temporal compression takes more time than that without compression when compression rate is higher than 0.7 for D1 and D2. However, the run-time is markedly reduced with lower compression rate and the prediction accuracy is not sacrificed much, which indicates that the proposed temporal compression algorithm can effectively improve the prediction efficiency.

5 CONCLUSIONS

This paper presents a worst-case power integrity prediction framework with CNN. The framework employs spatial and temporal compression to reduce the input and network complexity. An expert knowledge guided feature extraction is then used to further simplify the network structure. Finally, a novel network architecture is designed with three subnets and can predict the worst-case PDN noise or bump current by only changing the last subnet. Experimental results show that the proposed framework beats both the commercial tool and a recently reported solver with 1-2 orders of magnitude improvements, which has good scalability and efficiency while maintaining high accuracy.

REFERENCES

- [1] Karim Arabi, Resve Saleh, and Xiongfei Meng. 2007. Power Supply Noise in SoCs: Metrics, Management, and Measurement. *IEEE Design & Test of Computers* 24, 3 (2007), 236–244.
- [2] Chuangtao Chen, Weikang Qian, Mohsen Imani, Xunzhao Yin, and Cheng Zhuo. 2021. PAM: A Piecewise-Linearly-Approximated Floating-Point Multiplier with Unbiasedness and Configurability. *IEEE Transactions on Computers (Early Access)* (2021), 1–15.
- [3] Hai-Bao Chen, Sheldon X.-D. Tan, Xin Huang, Taeyoung Kim, and Valeriy Sukharev. 2016. Analytical Modeling and Characterization of Electromigration Effects for Multibranch Interconnect Trees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35, 11 (2016), 1811–1824.
- [4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 785–794.
- [5] Yufei Chen, Haojie Pei, Xiao Dong, Zhou Jin, and Cheng Zhuo. 2022. Application of Deep Learning in Back-End Simulation: Challenges and Opportunities. In *IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)*. 641–646.
- [6] Vidya A. Chhabria, Vipul Ahuja, Ashwath Prabhu, Nikhil Patil, Palkesh Jain, and Sachin S. Sapatnekar. 2021. Thermal and IR Drop Analysis Using Convolutional Encoder-Decoder Networks. In *IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)*. 690–696.
- [7] Vidya A. Chhabria, Andrew B. Kahng, Minsoo Kim, Uday Mallappa, Sachin S. Sapatnekar, and Bangqi Xu. 2020. Template-Based PDN Synthesis in Floorplan and Placement Using Classifier and CNN Techniques. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASP-DAC)*. 44–49.
- [8] Eli Chiprout. 2010. On-die power grids: The missing link. In *ACM/IEEE Design Automation Conference (DAC)*. 940–945.
- [9] Sukanta Dey, Sukumar Nandi, and Gaurav Trivedi. 2020. Machine Learning Approach for Fast Electromigration Aware Aging Prediction in Incremental Design of Large Scale On-Chip Power Grid Network. *ACM Transactions on Design Automation of Electronic Systems* 25, 5 (2020), 1–29.
- [10] Sukanta Dey, Sukumar Nandi, and Gaurav Trivedi. 2020. PowerPlanningDL: Reliability-Aware Framework for On-Chip Power Grid Design using Deep Learning. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1520–1525.
- [11] Sukanta Dey, Sukumar Nandi, and Gaurav Trivedi. 2021. Machine Learning for VLSI CAD: A Case Study in On-Chip Power Grid Design. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. 378–383.
- [12] Yen-Chun Fang, Heng-Yi Lin, Min-Yan Su, Chien-Mo Li, and Eric Jia-Wei Fang. 2018. Machine-Learning-Based Dynamic IR Drop Prediction for ECO. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–7.
- [13] Chia-Tung Ho and Andrew B. Kahng. 2019. IncPIRD: Fast Learning-Based Prediction of Incremental IR Drop. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–8.

- [14] Xin Huang, Armen Kteyan, Sheldon X.-D. Tan, and Valeriy Sukharev. 2016. Physics-Based Electromigration Models and Full-Chip Assessment for Power Grid Networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35, 11 (2016), 1848–1861.
- [15] Wentian Jin, Liang Chen, Sheriff Sadiqbatcha, Shaoyi Peng, and Sheldon X.-D. Tan. 2021. EMGraph: Fast Learning-Based Electromigration Analysis for Multi-Segment Interconnect Using Graph Convolution Networks. In *ACM/IEEE Design Automation Conference (DAC)*. 919–924.
- [16] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980* (2014), 1–15.
- [17] Shih-Yao Lin, Yen-Chun Fang, Yu-Ching Li, Yu-Cheng Liu, Tsung-Shan Yang, Shang-Chien Lin, Chien-Mo Li, and Eric Jia-Wei Fang. 2018. IR drop prediction of ECO-revised circuits using machine learning. In *IEEE VLSI Test Symposium (VTS)*. 1–6.
- [18] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azadeh Nazi, Jiwoo Pak, Andy Tong, Kavya Srinivasa, William Hang, Emre Tuncer, Quoc V. Le, James Laudon, Richard Ho, Roger Carpenter, and Jeff Dean. 2021. A graph placement methodology for fast chip design. *Nature* 594, 7862 (2021), 207–212.
- [19] Vivek Mishra and Sachin S. Sapatnekar. 2013. The impact of electromigration in copper interconnects on power grid integrity. In *ACM/IEEE Design Automation Conference (DAC)*. 1–6.
- [20] Seyed Nima Mozaffari, Bonita Bhaskaran, Kaushik Narayanun, Ayub Abdollahian, Vinod Pagalone, Shantanu Sarangi, and Jonathon E Colburn. 2019. An efficient supervised learning method to predict power supply noise during at-speed test. In *IEEE International Test Conference (ITC)*. 1–10.
- [21] Chi-Hsien Pao, An-Yu Su, and Yu-Min Lee. 2020. XGBIR: An XGBoost-based IR Drop Predictor for Power Delivery Network. In *IEEE/ACM Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1307–1310.
- [22] Haifeng Qian, S. R. Nassif, and S. S. Sapatnekar. 2006. Power Grid Analysis Using Random Walks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 24, 8 (2006), 1204–1224.
- [23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 234–241.
- [24] Sheriff Sadiqbatcha, Zeyu Sun, and Sheldon X.-D. Tan. 2020. Accelerating Electromigration Aging: Fast Failure Detection for Nanometer ICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 4 (2020), 885–894.
- [25] Zhiyao Xie, Haoxing Ren, Brucek Khailany, Ye Sheng, Santosh Santosh, Jiang Hu, and Yiran Chen. 2020. PowerNet: Transferable Dynamic IR Drop Estimation via Maximum Convolutional Neural Network. In *IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)*. 13–18.
- [26] Min Zhao, Rajendran V. Panda, Sachin S. Sapatnekar, Tim Edwards, Rajat Chaudhry, and David Blaauw. 2000. Hierarchical Analysis of Power Distribution Networks. In *ACM/IEEE Design Automation Conference (DAC)*. 150–155.
- [27] Xueqian Zhao, Zhuo Feng, and Cheng Zhuo. 2014. An efficient spectral graph sparsification approach to scalable reduction of large flip-chip power grids. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 218–223.
- [28] Han Zhou, Wentian Jin, and Sheldon X.-D. Tan. 2020. GridNet: Fast Data-Driven EM-Induced IR Drop Prediction and Localized Fixing for On-Chip Power Grid Networks. In *IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. 1–9.
- [29] Zhengyong Zhu, Bo Yao, and Chung-Kuan Cheng. 2003. Power Network Analysis Using an Adaptive Algebraic Multigrid Approach. In *ACM/IEEE Design Automation Conference (DAC)*. 105–108.
- [30] Cheng Zhuo, Di Gao, Yuan Cao, Tianhao Shen, Li Zhang, Jinfang Zhou, and Xunzhao Yin. 2021. A DVFS Design and Simulation Framework Using Machine Learning Models. *IEEE Design & Test (Early Access)* (2021), 1–7.
- [31] Cheng Zhuo, Jiang Hu, Min Zhao, and Kangsheng Chen. 2008. Power Grid Analysis and Optimization Using Algebraic Multigrid. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27, 4 (2008), 738–751.
- [32] Cheng Zhuo, Shaoheng Luo, Houle Gan, Jiang Hu, and Zhiguo Shi. 2020. Noise-Aware DVFS for Efficient Transitions on Battery-Powered IoT Devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 7 (2020), 1498–1510.
- [33] Cheng Zhuo, Kassan Unda, Yiyu Shi, and Wei-Kai Shih. 2019. From Layout to System: Early Stage Power Delivery and Architecture Co-Exploration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 38, 7 (2019), 1291–1304.
- [34] Cheng Zhuo, Gustavo Wilke, Ritochit Chakraborty, Alaeddin A Aydiner, Sourav Chakravarty, and Wei-Kai Shih. 2015. Silicon-Validated Power Delivery Modeling and Analysis on a 32-nm DDR I/O Interface. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 9, 23 (2015), 1760–1771.