Generative Multi-Physics Models for System Power and Thermal Analysis Using Conditional Generative Adversarial Networks

Priyank Kashyap*
Storage Division
Hewlett Packard Enterprise
Colorado Springs, CO
priyank.kashyap@hpe.com

Chris Cheng
Storage Division
Hewlett Packard Enterprise
Milpitas, CA
chris.cheng@hpe.com

Yongjin Choi
Storage Division
Hewlett Packard Enterprise
Colorado Springs, CO
yongjin.choi@hpe.com

Paul Franzon
ECE Dept.
North Carolina State University
Raleigh, NC
paulf@ncsu.edu

***Abstract*—As system performance increases, chip density and power consumption also increase. Power integrity and thermal management have become critical to the design flow and are codependent on each other. Advanced simulation tools perform co-simulation of electrical and thermal analysis on package-board designs. This paper describes a novel way of using a class of deep learning algorithms called conditional GANs (cGANs) to efficiently model the power/thermal co-simulation task. As the name suggests, cGANs are generative models that can predict unseen simulation conditions. Using the cGAN, the root-mean-squared error on unseen test cases is 0.015 in a [-1,1] range, translating to an error under 0.3 C°. Furthermore, a trained network exhibits fast inference speeds, allowing for near real-time generation of analysis results. This is a common goal of digital twins for dynamic system performance tuning.**

***Index Terms*—Power integrity, thermal analysis, digital twins, GAN , multi-physics, co-simulation**

I. Introduction

As computer system performance follows "Moore's Law" in doubling every 18 months, power distribution and thermal analysis are no longer independent. A typical high-performance system can comprise a printed circuit board (PCB) and multiple high-power packages under a range of ambient temperatures with limited airflow to cool it. Power is dissipated inside the package on the die and in the PCB as current flows through the copper power planes with electrical resistance. The heat dissipates out of the system through conduction, convection, and radiation. Since the electrical resistance increases with temperature, physics increasingly couples the power integrity and thermal response. As the resistance increases, power dissipation increases resulting in further temperature rises. The current state-of-the-art multi-physics simulators, such as Cadence© PowerDC™, have an iterative approach. The tools perform power and electrical simulations and feed the temperature solution back to recalculate the electrical parameters. The electrical stimulation will be restarted with new parameters until the power and thermal solution reach equilibrium.

Fig. 1 shows an example where excessive power is drawn through a backplane with highly perforated power planes; a local hot spot near the power connector (shown in white) results in overheating and reliability concerns on the PCB.

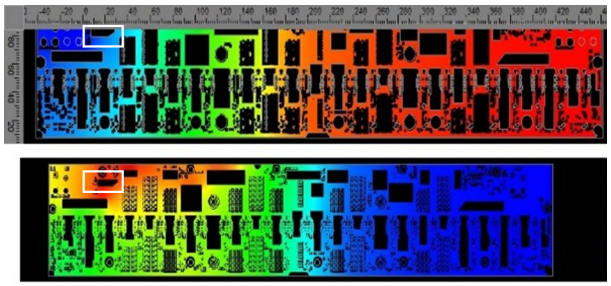


Fig. 1: The backplane thermal hotspot (bottom) due to excessive power drop (top).

With machine learning (ML) finding wide applications across the industry, there has been a push to look at thermal problems across the design flows. Chhabria et al. [1] show how a U-Net [2], which comprises an encoder-decoder network, converts a power map to a temperature map for a power delivery network (PDN). In contrast, Jin et al. [3] propose a generative adversarial network (GAN) to predict the thermal performance of a commercial chip based on the chip's performance counter and extend the approach to predict transient thermal maps. Stipsitz and Sanchis-Alepuz [4] performed a study where they randomized placements for elements on a PCB and predicted the thermal response given the power consumption using convolutional neural networks (CNNs). Lastly, Kashyap et al. [5] use cGANs to predict the heat map for a layer in a 3D integrated chip (3DIC) given a fixed 3D stackup with fixed power along one of the layers.

With these advancements in ML, there has been a push to use generative models across different domains. Generative models distinguish themselves from traditional discriminative models in attempting to model the joint distribution of the input and output. In contrast, a discriminative model tries to model the conditional probability of the output given an input. This work aims to model multi-physics power to thermal problems using cGANs, unlike prior work where they predict electrical-to-electrical problems.

* Work done at North Carolina State University

979-8-3503-1798-5/23/$31.00 ©2023 IEEE

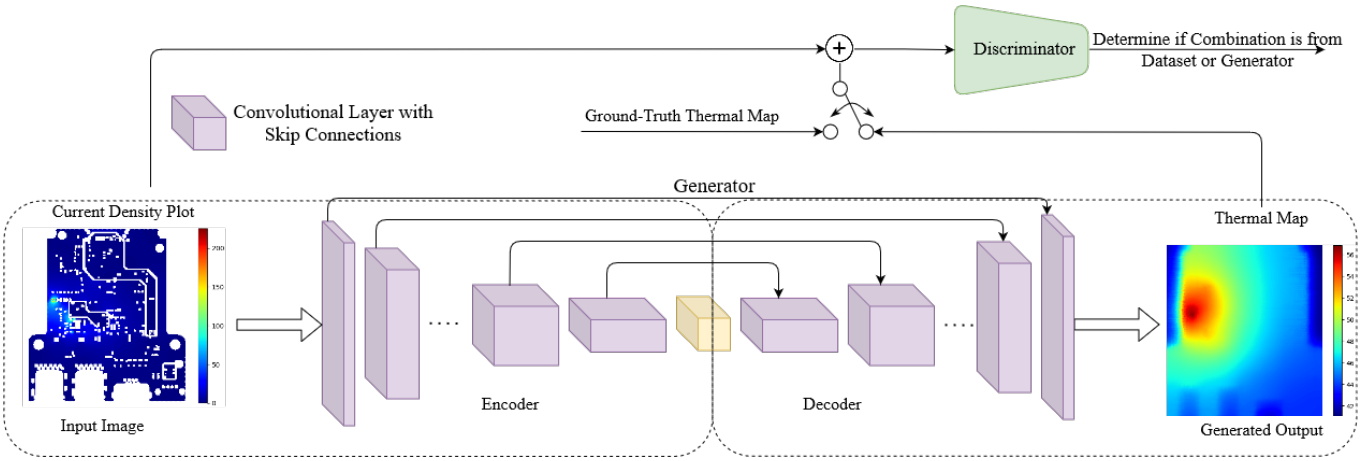


Fig. 2: GAN architecture used for training. The generator is a U-Net that forwards information from the encoder to the decoder to generate a heat map. The discriminator predicts whether the combination of the current density and heat maps are real.

The rest of the paper is organized as follows. Section II consists of background information on cGANs and details the proposed approach. Section III describes the data collection and the result evaluation criteria. Section IV shows the experimental results of a two-chip PCB design. Section V concludes the articles and some discussion of future work.

II. Conditional GANs for Thermal Analysis

GANs and their conditional variants, cGANs, have tremendous applications in numerous fields, especially in EDA [5]. GANs and cGANs refer to two models, a generator and a discriminator, that play a min-max optimization to generate samples with underlying data properties. Looking specifically at the cGAN, the two models it contains, the generator, *G*, generates a sample, $\hat{y}$, based on a random noise vector, z, and some conditional parameter, x. The discriminator, *D*, then gets either the generated sample, $\hat{y}$, or the ground-truth sample, y, and determines whether the combination of $\hat{y}$ and x is real. To that extent, Equation 1 shows the loss function for the cGAN.

$$\begin{aligned} L_{\text{cGAN}}(G,D) = &\mathbb{E}_{x,y}\left[\log D(x|y)\right] \\ &+ \mathbb{E}_{x,z}\left[\log(1 - D(G(y,z)|y))\right]. \end{aligned} \tag{1}$$

The discriminator aims to maximize the function, whereas the generator aims to minimize it. Furthermore, computing the ℓ_1 loss between the generated and real image improves the generator's ability to generate realistic samples in paired image-to-image translation tasks [6]. The addition of the ℓ_1 loss modifies Equation 1 as follows:

$$L = L_{\text{cGAN}}(G,D) + \lambda\ell_1. \tag{2}$$

where λ is a task-tuneable hyper-parameter.

In this work, the generator architecture is a U-Net, which performs domain-to-domain translation for image tasks and has found applications for integrated chip (IC) thermal prediction [1], [5]. In this implementation, the generator takes a current density map of the signal layer of the PCB as an input. Then it passes it through an encoder which compresses the input to a bottleneck phase. The generator's decoder then reconstructs the desired heat map using the bottleneck phase. Unlike regular GANs, this implementation implicitly adds noise by using dropout layers in the decoder.

Fig. 2 shows the overall generator architecture with the encoder, decoder, and bottleneck shown in yellow. Each layer in the encoder contains a convolutional layer and layer normalization, which reduces the input dimensions and prevents a covariate shift. The output of each of the encoder layers has a LeakyReLU activation. The decoder contains ConvolutionalTranspose layers that upsample from the bottleneck to the desired resolution, with the first 3 decoder layers having dropout. The final layer of the decoder has $tanh$ activations to ensure it can accurately recover the image in a $[-1, 1]$ range.

The discriminator in this work is a CNN that outputs a simple binary prediction to indicate whether the input is real. The discriminator takes the conditional current density map with either the ground truth or generated heat map. The network is a series of encoder blocks that reduce the input's dimension with a flattened and fully connected layer with a single unit at the output. To enable the training of the cGAN, the loss function is binary cross entropy.

III. Dataset Creation

Fig. 3 is the example problem using two high-power chips (highlighted in red) on the PCB, drawing a high current. The voltage regulator at the upper left corner supplies the current. The high current concentrated near the regulator results in a high-temperature rise. As a part of the data collection process, we sweep randomly select the sink current between $20\,\text{A}$ to $40\,\text{A}$ for one chip while keeping the sink current on the other to $0.1\,\text{A}$, the minimum allowed in Cadence© PowerDC™. The range for the selection is between $20\,\text{A}$ to $40\,\text{A}$ as currents above that yield an unrealistic temperature range above $100\,\text{C}^\circ$. Then we flip the currents on the chips and capture the relevant heat maps. The collection process first runs a purely electrical simulation to obtain the relevant current density maps. Then it switches the workflow to an electrically aware thermal simulation which takes $30\,\text{seconds}$ per run. As a part of this data collection, we collect 100 samples, of which 80 are for model training and validation, with the remaining

20 for testing the model. Fig. 2 shows a sample current density map and the corresponding heat map where the first chip has current and the second chip is off.

Before training the cGAN, preprocessing resizes the current density and heat maps to 256×256. After scaling to the resolution, it fills with the current density maps with zeros in regions where items are present, such as vias. It then scales min-max the current density and heat maps so that the ranges are $[-1, 1]$.

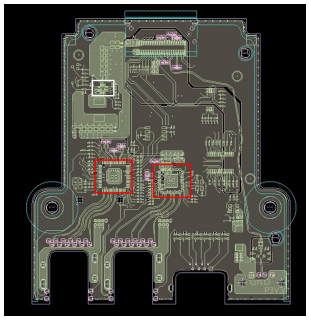


Fig. 3: Reference PCB design for train/test cases.

IV. Experimental Results

This section details the results of the cGAN on the previously unseen test set. Fig. 4 compares the ground truth heat map to the cGAN generated. As evident from the figure, the cGAN recovers the heat maps with high accuracy with a root-mean-squared error (RMSE) on the test as 0.015, which translates to 0.3 C°. Fig. 5b shows the distribution of the RMSE over the entire test for each heat map. The RMSE distribution ranges from $[0.007, 0.066]$ or $[0.02, 1.8]$ C°, with over 70% samples having an error less the 0.01. Furthermore, in regions with no electrical components, the cGAN recovers the correct ambient temperature, and the locations of the hotspots are also highly accurate.

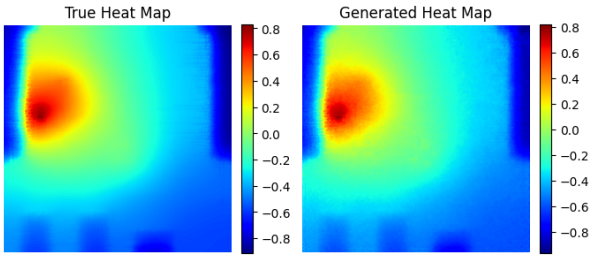
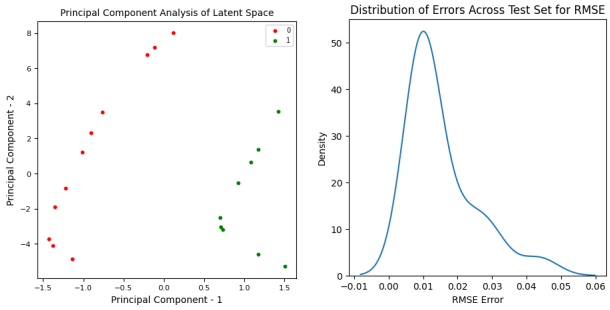


Fig. 4: Result from the test set

On further investigation of the bottleneck phase of the generator, we see that it can separate the different conditions. Fig. 5a uses principal componenet analysis to represent the high-dimensional latent space in 2 dimensions. The red and green correspond to chip 1 and chip 2 being on, respectively. It is clear from the figure that the cGAN can determine the different sink currents applied to each chip and use that to generate a heat map.

One of the more essential aspects of using ML models is their fast inference. In this work, the cGAN takes an average of $100\,\mathrm{ms}$ to predict the test cases. Moreover, the cGAN does not require significant training time, with the model taking under 20 minutes to train. The reported training/inference times are on an Nvidia© 2080-TI GPU.



(a) Latent space visualization (b) Test set error distribution

Fig. 5: Results on the latent space (left) and RMSE over all test case (right).

V. Conclusion

This paper demonstrates the flexibility of using cGANs to model complex multi-physics problems. The trained cGAN can provide near real-time prediction of the hot spot temperature given the ambient temperature and chip power load. Fan speed can be dynamically adjusted to maintain the necessary temperature to ensure component reliability at the hot spot. Even though the sample problem is a relatively simple case, the cGAN can be extended to a larger number of chips and multiple hotspots or 3D packaging. There is substantial computation headroom to train and perform inference on much more complex problems. We will continue to investigate these applications, especially in 3D packaging for chiplets.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. CNS #2137283 - Center for Advanced Electronics through Machine Learning (CAEML) and its industry members. We would like to thank Dr. Chau-Wai Wong, Dr. Tianfu Wu and Dr. Dror Baron for their technical inputs.

References

[1] V. Chhabria, V. Ahuja, A. Prabhu, et al., "Thermal and IR drop analysis using convolutional encoder-decoder networks," in *26th Asia and South Pacific Design Automation Conference*, 2021, pp. 690–696.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[3] W. Jin, S. Sadiqbatcha, J. Zhang, et al., "Full-chip thermal map estimation for commercial multi-core cpus with generative adversarial learning," in *39th International Conference on Computer-Aided Design*, 2020, pp. 1–9.

[4] M. Stipsitz and H. Sanchis-Alepuz, "Approximating the steady-state temperature of 3D electronic systems with convolutional neural networks," *Mathematical and Computational Applications*, vol. 27, no. 1, p. 7, 2022.

[5] P. Kashyap et al., "Thermal estimation for 3D-ICs through generative networks," in *2023 IEEE International 3D Systems Integration Conference (3DIC)*, 2023, pp. 1–4.

[6] P. Isola et al., "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967–5976.