Graph Representation Learning for Parasitic Impedance Prediction of the Interconnect

Pratik Shrestha and Ioannis Savidis
Department of Electrical and Computer Engineering, Drexel University
Philadelphia, Pennslyvania 19104
ps937@drexel.edu, isavidis@coe.drexel.edu

2023 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-5109-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISCAS46773.2023.10181447

***Abstract*—An accurate early estimate of the post routing interconnect parasitics allows for pre-emptive changes to the circuit in earlier phases of the design flow, significantly reducing the design time and effort. In this work, graph based deep regression models are proposed to predict the post routing interconnect capacitance of a circuit by utilizing layout information and the post placement estimates of the interconnect parasitics. The post placement capacitance determined by a commercial physical design tool is used as a baseline for the models, with the mean absolute percentage error (MAPE), the mean absolute error (MAE), and R^2 score calculated for comparison. The proposed methodology outperforms the baseline provided by the commercial physical design tools based on results obtained across all trained models, with an average improvement of 23.39% in MAPE, 5.33% in MAE, and 1% in R^2 score. The proposed methodology also provides better prediction of the worse case errors as compared to the commercial tool, with the model providing an average improvement of 47.43% in MAPE and 14.31% in MAE for the nets with the largest 1% of errors as determined by the tool.**

***Index Terms*—parasitic prediction, physical design, machine learning, graph convolutional networks**

I. Introduction

A significant portion of the run time of modern electronic design automation (EDA) is spent on iterative execution of the design flow to meet the target circuit specifications and timing constraints. Timing verification in the early stage of physical design depends on the accurate estimate of interconnect parasitics. While performing static timing analysis (STA) is possible after standard cells are placed, no interconnect topology is defined until detailed routing. Since routing requires significant computational resources, a predictive tool that provides an accurate estimate of the post routing interconnect parasitics after placement significantly reduces the time to complete the physical design of a circuit.

Graphs are an effective way to represent a wide range of structures. Graph based neural networks (GNNs) [1] are utilized to model complex representations into a low dimensional vector embedding, which is then applied to different machine learning applications. Circuit netlists represented as graphs leverage structural information to allow for more holistic learning. Graph based learning approaches have been utilized for design automation [2]–[4] and performance modeling [5]–[7] of circuits. In [8], GNNs are used to predict layout parasitic capacitances and device parameters from the topological and sizing information of analog and mixed signal circuits provided in the schematics. An XGBoost based regression model is proposed in [9] to predict the parasitic impedance post detailed routing of a digital circuit utilizing data available after global routing.

In this paper, a novel graph based regression framework is introduced to predict the post routing interconnect capacitance after the placement of a digital circuit. The framework utilizes, as input data for the models, the post-placement estimates of the interconnect impedance within a spatial graph representation that includes node features populated from the circuit netlist and the parasitic reports. The contributions of the paper include

- Error characterization of the estimates of the parasitic impedance of the interconnect provided by a commercial physical design tool as a baseline, with the mean absolute percentage error (MAPE), mean absolute error (MAE), and R2 score calculated for comparison,
- A netlist to interconnect graph representation with embedded node feature attributes, and
- A graph-based deep neural regression model to predict the total capacitance of each interconnect net of a circuit.

The paper is organized as follows. Background on graph machine learning and spatial graph convolutional modeling is provided in Section II. The dataset generated for the modeling and error analysis of the estimated interconnect parasitics using a commercial physical design tool are described in Section III. The parasitic prediction framework is discussed in Section IV, and an analysis of the results produced by the framework is provided in Section V. Some concluding remarks are provided in Section VI.

II. Graph Representation Learning

In this work, spatial graph convolution layers (SGCN) [10] are utilized to model interconnect capacitance from a distributed interconnect graph. SGCN extends a normal GCN to leverage node positions within the convolution function and uses spatial features to learn from graphs that have an inherent spatial arrangement. Given a directed graph $G = (V, E)$, where $v \in V$ is a node of the graph and $e \in E$ is an edge of the graph, a graph convolution layer is defined by an aggregation function and multi-layer perceptron. The aggregation function is defined as

$$\bar{h}_i = \sum_{(v_i, v_j) \in E_i} u_{ij} h_j, \tag{1}$$

where E_i is the set of edges connecting node v_i to all neighboring nodes represented by a set V_j such that node $v_j \in V_j$, h_i denotes node features applied as input to a convolutional layer, and u_{ij} is a trainable weight for aggregation of node v_i with respect to features of node v_j. The representation of (1) in matrix form is given by $\bar{H} = UH^T$, where H denotes the matrix of node features such that $\bar{H} = [\bar{h}_1, ..., \bar{h}_n]$, and U denotes a trainable weight matrix. SGCN extends the aggregation function as given by

$$\bar{h}_i(U, b) = \sum_{(v_i, v_j) \in E_i} ReLU(U^T(p_i - p_j) + b) \odot h_j, \tag{2}$$

where p_n is the matrix representation of the coordinate location of node v_n, ReLU is a rectified linear activation function, and $\odot$ represents element-wise multiplication. The extended aggregation function is utilized by the multi-layer perceptron layer as given by

$$MLP(\bar{H};W) = ReLU(W^T\bar{H} + b), \tag{3}$$

where W and b are the trainable weight matrix and bias vector, respectively.

III. Dataset and Baseline Analysis

In this work, six IWLS'05 sequential benchmark circuits [11], which are listed in Table I, are utilized to build a dataset of layouts. The dataset is generated on a 65 nm technology node. The initial benchmark circuit is synthesized into a technology-specific gate-level netlist using Synopsys Design Compiler and the gate-level netlist is placed and routed into a final layout using Synopsys IC Compiler. Interconnect parasitics are also extracted from Synopsys IC Compiler as Standard Parasitic Exchange Format (SPEF) files [12], with data from the files utilized to develop model features as listed in Table II. The circuits are synthesized to provide an operating frequency of 1 GHz, a circuit aspect ratio of 0.5, and an area utilization of 70%.

TABLE I: IWLS'05 benchmark circuits.

Circuit	ac97_ctrl	aes_core	wb_conmax	des3_perf	pci	usb_funct
No. of nets	5041	14044	17326	56018	7586	6898

Given the initial (placement) and final (routing) stages of the IC physical design flow, the total capacitance of the initial stage provided by the design tool is considered as the baseline prediction for the parasitic profile of the final stage. Scatter plots of the post placement and post routing interconnect capacitance of the entire dataset are shown in Fig. 1a. The metrics evaluating the error, specifically the MAPE, MAE, and R^2 score, are selected as a baseline and are calculated as, respectively,

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}|1 - \frac{\hat{y}_i}{y_i}|, \tag{4}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \text{ and} \tag{5}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\sum_{i=1}^{n}|y_i - \bar{y}_i|}, \tag{6}$$

where n is the total number of nets in the overall dataset, y is the interconnect capacitance post routing, $\hat{y}$ is the estimated interconnect capacitance post placement, and $\bar{y}$ is the average interconnect capacitance post routing. The baseline errors for each of the circuits are listed in Table III. There is a significant error between the capacitance provided from the two different design stages, especially when considering the maximum average errors and average percentage errors in certain critical interconnect.

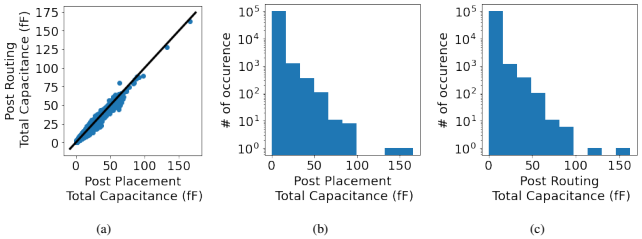

Fig. 1: Analysis of the (a) post placement and post routing interconnect capacitance, (b) post placement interconnect capacitance histogram (log scaled), and (c) post routing interconnect capacitance histogram (log scaled).

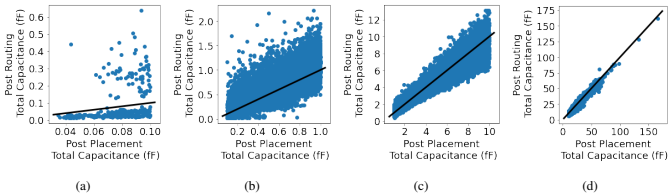

Fig. 2: Post placement and post routing interconnect capacitance distributed by post placement capacitance ranges of (a) 0 to 0.1 fF, (b) 0.1 to 1 fF, (c) 1 to 10 fF, and (d) 10 fF and above.

Histograms of both the post placement and post routing interconnect capacitance are shown in Fig. 1b and 1c, respectively. A significantly left-skewed logarithmic distribution of the interconnect capacitance is observed. Upon grouping the net capacitances from all circuits into the ranges of 0 to 0.1 fF, 0.1 to 1 fF, 1 to 10 fF, and 10 fF and above, the larger interconnects, although resulting in lower proportional error, are observed to contribute more to the total error of the model. Scatter plots of the post placement and post routing interconnect capacitance grouped by total net capacitance are provided in Fig. 2. The mean and max absolute and percentage errors by benchmark circuit and by total interconnect capacitance are listed in Tables III and IV, respectively, and are further discussed in Section V.

IV. Graph Based Parasitic Prediction

To predict the total capacitance of an interconnect, the routed nets from the physical implementation of the circuit (DEF file) are mapped into directed graphs and are populated with node features extracted from both the netlist and reports on the parasitic impedance. Additional details on the generation of the graphs are provided in Section IV-A. The generated interconnect graph representation is then used as input to a graph neural network, which is described in Section IV-B.

A. Graph Structure and Feature Engineering

The overall execution of the framework and training process is shown in Fig. 3. The SPEF file generated post placement provides an estimated parasitic impedance of each interconnect of the circuit. As shown in Fig. 4a, an estimated routing based on the placed circuit is used to calculate the distributed parasitic impedance of each net. The physical characteristics of the circuit, described in the Verilog, DEF, and SPEF files, are parsed and represented as an interconnect graph $G = (V, E)$, where node $v \in V$ represents estimated interconnect segments and edge $e \in E$ represents a connection between two or more interconnect segments. A spatial graph is depicted in Fig. 4b that is based on the estimated post placement interconnect shown in Fig. 4a. The interconnect graphs represent primary inputs to the model.

Each node of an interconnect graph is populated with a carefully selected feature set, which is listed in Table II. Parasitic features are net capacitance values extracted from the SPEF files generated by IC Compiler. Both the total parasitic impedance of each net and the distributed parasitic impedance of each net segment are extracted from the SPEF file. Interconnect graphs are traversed to compute structural features, whereas spatial features are calculated as the midpoint of the estimated location of the interconnect segments provided in the SPEF file. The interconnect segments of the SPEF file are also utilized to calculate the congestion features of each node of the interconnect graph. The circuit layout is partitioned into a grid of cells each with dimension of 10×10 μm^2 and the pin

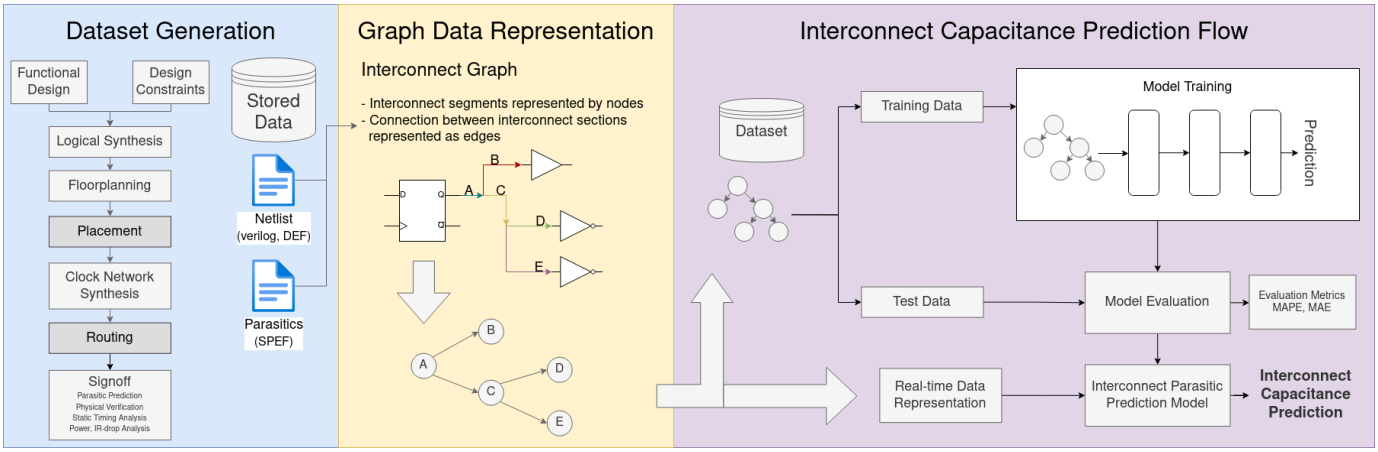

Fig. 3: Dataset generation and deep learning framework for post placement to post routing interconnect parasitic impedance prediction.

and node density is calculated for each cell of the grid. The mean and sum of all the cells through which the interconnect passes are considered as the congestion feature of the graph node representing the given interconnect segment.

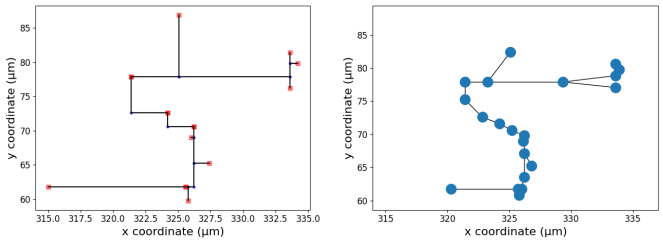

Fig. 4: Geometrical representation of the (a) routing estimated based on the post placement SPEF file and (b) interconnect spatial graph based on the estimated routing.

TABLE II: Node features of the interconnect graph.

Feature Type	Feature	Source
Post placement parasitics features	Interconnect segment capacitance	SPEF file
Structural features	Interconnect segment level	Calculated from netlist graph representation
	Interconnect segment length	
Spatial features	Interconnect segment midpoint (x, y)	LEF/DEF file
Congestion features	Pin density	Calculated from placed netlist (DEF file) and SPEF file
	Net density	

B. Model Architecture and Training

After completing feature extraction and graph representation of the circuit, a regression model is developed to predict the interconnect parasitics. The regression model, as shown in Fig. 5, is a deep neural network consisting of an SGCN layer connected to six linear layers of size 16. Each hidden layer utilizes a rectified linear unit (ReLU) [13] as an activation function. A gradient clipping for weights greater than 1 is applied after each activation to avoid exploding gradients. The primary objective function is to minimize the mean squared error (MSE) loss between the actual post routing interconnect capacitance y and the predicted estimate of the post routing interconnect capacitance $\hat{y}$. The MSE loss is given by

$$MSELoss = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{7}$$

where n is the total number of nets in the dataset.

V. Results

In this section, an analysis of the results produced by the parasitic prediction framework described in Section IV is

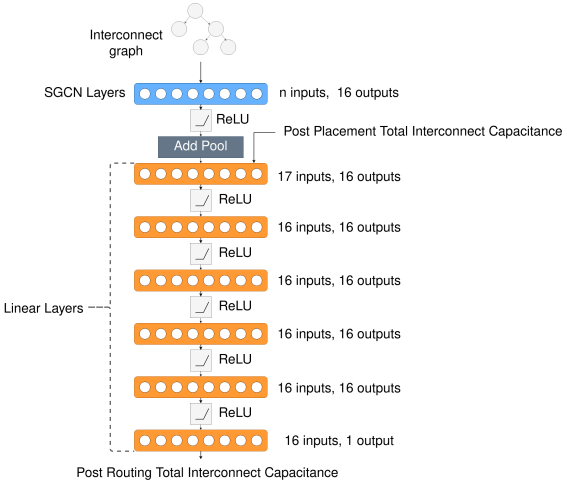

Fig. 5: Graph regression network architecture.

provided. The dataset from the six circuits listed in Table I is distributed for six instances of the proposed neural regression model, where for each model, one of each circuit is considered as a test set, and the remaining five circuits are considered as the training set. The models are trained with a stochastic gradient descent (SGD) optimizer [14] and divided into batches of 1024 interconnect graphs for a single iteration of training and validation. An initial learning rate of 0.01 is applied that decays at a rate of 95% every 10 training steps. Early stopping is adopted such that the training process is terminated when there is no improvement in the training loss for 10 epochs, which prevents the model from overfitting by learning the statistical noise in the training dataset. The framework is implemented in Python 3 using the PyTorch-geometric deep learning library [15], and the training is performed on a system with 96 GB memory, twenty four 4 GHz CPU cores, and an NVIDIA GeForce GTX 1080 graphics card.

The trained regression models are validated against the test datasets. The R^2 score, total MAPE, total MAE, and MAPE and MAE of the top 1% worse case errors are calculated for comparison of the models, with the results as listed in Table III. Results obtained across the six models indicate that the proposed methodology outperforms the baseline errors produced by the commercial tool with an average improvement of 23.39% in the MAPE score and 5.33% in the MAE score. The improvement is even greater when considering worse

TABLE III: Analysis of the mean absolute percentage error (MAPE), mean absolute error (MAE), and R^2 score of the interconnect impedance of the six IWLS'05 benchmark circuits.

Circuit	Error Metric	Baseline	Proposed Model	Improvement
ac97_ctrl	**MAE**	0.3171	0.2757	13.05%
	worse 1% MAE	5.8559	5.2602	10.17%
	MAPE	22.43%	15.66%	30.17%
	worse 1% MAPE	198.27%	94.95%	52.11%
	R^2	0.9822	0.9878	0.57%
aes_core	**MAE**	0.4236	0.3421	19.24%
	worse 1% MAE	5.6783	3.6251	36.16%
	MAPE	22.66%	16.21%	28.49%
	worse 1% MAPE	234.63%	116.05%	50.54%
	R^2	0.9499	0.9756	2.70%
wb_conmax	**MAE**	0.3269	0.3827	-17.07%
	worse 1% MAE	4.9284	4.6358	5.94%
	MAPE	19.12%	19.45%	-1.73%
	worse 1% MAPE	120.32%	92.61%	23.03%
	R^2	0.9900	0.9897	-0.03%
des3_perf	**MAE**	0.3080	0.2579	16.26%
	worse 1% MAE	3.8316	3.0077	21.50%
	MAPE	22.92%	15.32%	33.14%
	worse 1% MAPE	178.52%	91.85%	48.55%
	R^2	0.9309	0.9606	3.19%
pci	**MAE**	0.3782	0.3682	2.64%
	worse 1% MAE	6.4478	6.2749	2.68%
	MAPE	20.25%	14.64%	27.70%
	worse 1% MAPE	184.44%	81.24%	55.95%
	R^2	0.9809	0.9833	0.25%
usb_funct	**MAE**	0.3786	0.3865	-2.10%
	worse 1% MAE	5.9766	5.4126	9.44%
	MAPE	20.12%	15.57%	22.60%
	worse 1% MAPE	179.08%	81.64%	54.41%
	R^2	0.9699	0.9631	-0.70%
Average across all models	**MAE**	0.3554	0.3355	5.33%
	worse 1% MAE	5.4531	4.7027	14.31%
	MAPE	21.25%	16.14%	23.39%
	worse 1% MAPE	182.54%	93.06%	47.43%
	R^2	0.9673	0.9767	1.00%

TABLE IV: Analysis of the mean absolute percentage error (MAPE), mean absolute error (MAE), and R^2 score of the interconnect impedance grouped by post placement total capacitance.

Circuit	Error Metric	Baseline	Proposed Model	Improvement
0 to 0.1	**MAE**	0.0705	0.3239	-359.09%
	worse 1% MAE	0.4256	0.9710	-128.11%
	MAPE	90.40%	81.78%	9.53%
	worse 1% MAPE	584.45%	97.10%	83.39%
	R^2 **score**	0.0801	-35.6163	N/A
0.1 to 1	**MAE**	0.1403	0.1340	4.47%
	worse 1% MAE	0.6807	1.0317	-51.56%
	MAPE	31.33%	21.48%	31.43%
	worse 1% MAPE	217.25%	103.17%	52.51%
	R^2 **score**	0.5495	0.3490	-36.50%
1 to 10	**MAE**	0.3561	0.3296	7.42%
	worse 1% MAE	2.4771	0.4286	82.70%
	MAPE	13.58%	11.17%	17.75%
	worse 1% MAPE	60.41%	42.86%	29.05%
	R^2 **score**	0.9269	0.9414	1.56%
10 and beyond	**MAE**	2.0927	1.7632	15.74%
	worse 1% MAE	10.7077	0.3775	96.47%
	MAPE	12.53%	10.65%	15.02%
	worse 1% MAPE	39.82%	37.75%	5.19%
	R^2 **score**	0.9373	0.9565	2.05%

TABLE V: Total absolute errors distributed by post placement capacitance.

Capacitance Range (fF)	No. of nets	Baseline	Proposed Model	Improvement
0 to 0.1	403	28.43	130.51	-359.09%
0.1 to 1	48533	6,808.11	6,503.50	4.47%
1 to 10	53104	18,908.16	17,504.80	7.42%
10 and beyond	4873	10,197.49	8,591.97	15.74%
Total	106913	35,942.19	32,730.77	8.93%

case predictions, with the proposed methodology providing an average gain of 47.43% in MAPE and 14.31% in MAE over the baseline for the top 1% of nets with the largest error estimate. In addition, although there are model scenarios where the commercial tool outperforms the proposed models when considering the full dataset, specifically for wb_conmax and pci, the proposed architecture consistently outperforms the baseline across all circuit models for the 1% of nets with the greatest error estimates. Both the baseline and the proposed model provide R^2 scores greater than 0.95 for all training scenarios. However, the proposed model outperforms the baseline by a small margin, with an average improvement of 1%.

As shown in Fig. 1b and Fig. 1c, the distribution of the interconnect capacitance is heavily left-skewed, indicating that there are a greater number of interconnects with small capacitance as compared to interconnects with large capacitance. However, larger capacitive loads are generally more critical from the perspective of a designer. The dataset is, therefore, sub-divided by the post placement capacitance, where ranges of 0 to 0.1 fF, 0.1 to 1 fF, 1 to 10 fF, and 10 fF and above are considered. The R^2 score, total MAPE, total MAE, and MAPE and MAE of the top 1% of nets with the greatest errors are calculated for each range of capacitance as listed in Table IV. The proposed SGCN model predictions, when compared to the baseline predictions, result in greater absolute errors (total MAE and worse 1% of nets MAE) for smaller capacitances, which improves with increasing capacitance. The R^2 score also indicates that the model performs better for larger capacitances. The R^2 score for both the baseline and the proposed model is low for interconnects with smaller capacitance but improves with increasing capacitance. A negative R^2 score of -35.61 for the capacitance range of 0 to 0.1 fF indicates that the proposed model underperforms even as compared to the mean of the expected capacitances. In contrast, the improvements in the proportional errors (total MAPE and worse 1% of nets MAPE) are greater for smaller capacitances when utilizing the SGCN models over the baseline tool, but deteriorate as the capacitance increases. However, the SGCN model outperforms the baseline across all capacitance ranges in MAPE and worse 1% of nets MAPE. Considering the total absolute error across the capacitance ranges listed in Table V, although the proposed model performs 359.09% worse than the baseline for the smallest capacitance range of 0 to 0.1 fF, the prediction error only accounts for 3.17% of the total error of the dataset. The proposed model performs 15.74% better for the largest capacitance range (10 fF and beyond), which contributes 49.99% of the total error of the dataset.

VI. Conclusions

In this paper, a deep spatial graph regression model is proposed to predict the post routing interconnect parasitic impedance after completing placement. Interconnect graph representations of six post placement IWLS'05 benchmark circuits are generated. A commercial physical design tool is used to generate the datasets, and an analysis is performed to obtain a baseline error across the circuits. When validating the six trained models, each with data from one of the benchmark circuits as the test set and the rest as the training set, an average improvement of 23.39% in MAPE, 5.33% in MAE, and 1% in R^2 score is observed when predicting the post-routing interconnect parasitics after completing placement. When considering the worse 1% of predicted impedance errors, an average improvement of 47.43% in MAPE and 14.31% in MAE is observed as compared to the baseline. The proposed machine learning methodology largely reduces errors and outperforms the baseline provided by the commercial. The improvement is even more significant for interconnects with large errors in predicted impedance.

REFERENCES

[1] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, Vol. 1, pp. 57–81, Jan. 2020.

[2] W. Haaswijk, E. Collins, B. Seguin, M. Soeken, F. Kaplan, S. Süsstrunk, and G. De Micheli, "Deep learning for logic optimization algorithms," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, May 2018.

[3] A. Mirhoseini, A. Goldie, M. Yazgan, J. W. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, A. Nazi, *et al.*, "A graph placement methodology for fast chip design," *Nature*, Vol. 594, No. 7862, pp. 207–212, June 2021.

[4] Y.-C. Lu, J. Lee, A. Agnesina, K. Samadi, and S. K. Lim, "GAN-CTS: A generative adversarial framework for clock tree prediction and optimization," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, Nov. 2019.

[5] P. Shrestha, S. Phatharodom, and I. Savidis, "Graph representation learning for gate arrival time prediction," *Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD (MLCAD)*, pp. 127–133, Sept. 2022.

[6] Z. Wu and I. Savidis, "Transfer of performance models across analog circuit topologies with graph neural networks," *Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD (MLCAD)*, pp. 159–165, Sept. 2022.

[7] Z. Wu and I. Savidis, "Transfer Learning for Reuse of Analog Circuit Sizing Models Across Technology Nodes," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1033–1037, Nov. 2022.

[8] H. Ren, George F. Kokai, W. J. Turner, and T.-S. Ku, "ParaGraph: Layout parasitics and device parameter prediction using graph neural networks," *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, June 2020.

[9] V. A. Chhabria, W. Jiang, A. B. Kahng, and S. S. Sapatnekar, "From global route to detailed route: ML for fast and accurate wire parasitics and timing prediction," *Proceedings of the 2022 ACM/IEEE Workshop on Machine Learning for CAD (MLCAD)*, pp. 7–14, Sept. 2022.

[10] T. Danel, P. Spurek, J. Tabor, M. Śmieja, Ł. Struski, A. Słowik, and Ł. Maziarka, "Spatial graph convolutional networks," *Proceedings of the International Conference on Neural Information Processing (NIPS)*, pp. 668–675, Nov. 2020.

[11] C. Albrecht, "IWLS 2005 benchmarks," *In International Workshop for Logic Synthesis (IWLS): http://www.iwls.org*, June 2005.

[12] "IEEE standard for integrated circuit (IC) open library architecture (OLA)," *IEEE Std 1481-2019 (Revision of IEEE Std 1481-2009)*, pp. 1–641, Mar. 2020.

[13] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," *arXiv preprint arXiv:1803.08375*, pp. 1–7, Mar. 2018.

[14] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, pp. 1–14, Sept. 2016.

[15] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," *arXiv preprint arXiv:1903.02428*, pp. 1–9, Mar. 2019.