Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

Machine Learning-Based Congestion Optimization in VLSI Physical Design

Dr. Babu Illuri
Department of ECE
Vardhaman College of Engineering
Hyderabad, India

D. Sreeja Reddy
Department of ECE
Vardhaman College of Engineering
Hyderabad, India

Beera Praveen Kumar
Department of ECE
Vardhaman College of Engineering
Hyderabad, India

P. Siddarth Reddy
Department of ECE
Vardhaman College of Engineering
Hyderabad,India

Abstract— The purpose of the current study is to implement the concept of Graph Neural Networks (GNNs) in decreasing the number of congestions when implementing VLSI circuit placement and routing. In VLSI, like other fields, congestion estimation fields on rules-of-thumb and heuristic approaches, which are challenging to use with larger designs. Rather, GNNs can work on the circuit layout to create a graph, in which the components form the nodes, and their connections are the edges, and this is far more spatial. Because the GNN model involves utilization of the ISPD benchmark data congestion, it can be considered that the model can predict which areas house the most critical hotspots of activity during optimization without negatively influencing the other design parameters. The results indicated that GNNs sang on a scalable basis and were performing better than heuristics about the solution of modern VLSI problems

Keywords—*Congestion, Placement, GNN-based Algorithm, VLSI.*

I. INTRODUCTION

Circuit designs the design of circuits is a crucial, though basic, problem in circuit design to establish the shortest path between computational logic blocks in a VLSI (Very-Large-Scale Integration) circuit. A graph theory problem that dates to the 1950s, which addresses the shortest path problem, is the problem of identifying the most cost-effective route through two nodes (logic blocks) in a chip layout to reduce the wirelength, delay, and power. Particularly, it is applicable to VLSI design in which more efficient routing and placement will directly influence circuit performance, less congestion, and use of space.

The more complicated the design of integrated circuits, the bigger the importance of a solution to the shortest path problem and what is even more important is that the algorithms employed to solve shortest path ought to be more efficient and performance intensive. Dijkstra algorithm, A search and other old-fashioned graph-based algorithms are highly efficient in such cases when the topology is relatively simple.

The techniques are very popular in VLSI design since they provide the capability of locating a straight-line route between two nodes in a graph. The conventional algorithms might not however be optimal with respect to managing the dynamism of topology that are found in contemporary VLSI layouts. These algorithms are also inclined to concentrate on the discovery of one shortest route and thus, they are constrained in relation to the discovery of additional shortest routes or a consistent group of routes that would be crucial in congestion alleviation and the placement and routing optimization in complex design.

In recent years, machine learning, particularly deep learning methods, has become a matter of concern to overcome the constrained problems in conventional methods as far as optimization in the VLSI design is concerned. Of them, Graph Neural Networks (GNNs) have proven to be one of the most promising, especially in problems with graph-structured data, like the shortest path problem.

The fact that GNNs are more flexible than traditional ones is that it learns the association between the nodes and the edges of a graph and the dependencies between the components of the circuit must be known, comports the possibility to predict the congestion and optimize the routing plans. They are also Genetic Algorithms (GA), Simulated Annealing (SA), and Particle Swarm Optimization (PSO) among other heuristic algorithms in which the shortest path problem in VLSI routing was answered. Though these methods have a good future, they are limited by their ability to work with dynamic and large-scale designs and are generally not very computationally efficient.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

Nevertheless, GNNs provide a more scalable answer and can acquire complex circuit design relations and offer effective and successful solutions. The other algorithm that has been used in routing optimization is the Harmony Search Algorithm (HSA) that is a worldwide optimization procedure yet is founded on musical improvisation. Since HSA and other heuristic-driven algorithms like GA and PSO have proven to be effective in certain applications, GNNs provide a data-driven, complex methodology which outperforms traditional algorithms because it adapts to complex circuit layouts and is successful in finding optimal routes or near-optimal routes. The increased complexity of integrated circuits and additional development of VLSI design makes the application of more sophisticated optimization techniques increasingly significant.

The potentially promising approach to conventional algorithms is Graph Neural Networks (GNNs), a form of deep learning approach, capable of learning and adapting to the dynamic and complex nature of VLSI routing and placement. The paper discusses how the shortest path problem in VLSI design can be solved using Graph Neural Networks (GNNs), to be more efficient and accurate with routing and placement, to reduce congestion, and use the area to the full extent.

II. LITERATURE REVIEW

GNNs have found much interest in VLSI design optimization since it can capture complex relationships on graphical-structured data, such as circuit layouts. In this paper, several key works and developments will be summarized that implement the use of GNNs and other deep learning algorithms to optimize congestion, place various circuit components, and route in VLSI physical design. The authors of [1] investigated the VLSI routing problem that was solved using Graph Neural Network.

Their analysis showed that GNNs were applicable in representing how routing wires and logic cells in a circuit interacted with each other. The developed model had capabilities of predicting the most optimum routes and this was made possible by incorporating the graph structure of the design which is promising as an alternative to the conventional routing algorithms whose performance and scalability is usually limited. It provided a foundation on which further studies could be conducted on GNNs in VLSI design, since it demonstrated how it could be successfully applied to overcoming complicated routing issues.

The article [2] gives in-depth literature on the application of GNNs to maximize placements. The authors proposed a graph optimization problem-based GNN model of the placement problem, where nodes are the locations of computational units, and edges are their connectivity. The GNN model had the capacity to determine the interconnection between these blocks and predict placements that minimize congestion and maximize the used area. This algorithm outperformed old fashioned placement algorithms such as simulated annealing and force-directed algorithms in accuracy and computational cost.

In [2], the article gives the general outlook of the GNNs application in the placement of optimization. The authors presented the model of the GNN, according to which the tasks of the placement are solved as the graphical optimization problem, nodes of which are various positions of the calculation components. The later research has led to the application of GNNs in VLSI design estimation of congestion. The model proposed by the authors is developed based on GNN according to which the degree of congestion in different areas of the chip based on the location of the computational blocks is predicted. It was also trained with benchmark circuit dataset, and it was demonstrated that the number of seconds required to forecast congestion was much lower than the other more traditional means.

The current paper demonstrated how GNNs could potentially be used in the maximization process of placements, and early congestion analysis that is vital in the successful execution of VLSI design. The authors of [4] used Graph neural networks and reinforcement learning to address the problem of placement and routing. Reinforcement learning was used in the system to make the GNN focus on and optimize the placement of the blocks and route paths, as part of the decision-making process. The paper has put emphasis on the combined effect of the GNNs and the reinforcement of learning to solve more cumbersome and bulky challenges of VLSI design.

The algorithm was also significantly quicker in its execution, and produced almost-optimal solutions, creating which is important in real-time design problems. In the article [5], the authors were concerned with the combination of GNNs with a deep reinforcement learning (DRL) to the multi-objective optimization of VLSI placement and routing. The authors in this work have proposed a framework which could maximize different objectives, e.g. congestion, wirelength, and power consumption all at the same time. A system with benchmark-trained datasets performed better than the existing optimization algorithms, which shows how GNNs can take advantage of the complexity of trade-offs between design goals in competing physically designable systems of VLSI.

The utilization of GNNs on block positioning in 3D was another useful tool in [6]. The authors created the solution to the optimization problem of the location of blocks in a three-dimensional analysis of integrated circuit based upon the reference to the details of the three-dimensional design, inter-layer routing and thermal management in the form of GNN. The model would allow estimating placements with less congestion, and placing 3D stacking may become an option, which would make the model more complex. The article is a new step to apply GNNs to the successor of VLSI design that visualizes GNNs' abilities in higher-level 3D VLSI designs. In [7], a literature review was conducted on the deep learning-based

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

methods of VLSI design automation in detail. GNNs, placement, routing, congestion estimation, and power optimization are some of the methods of deep learning which have been mentioned in the paper. The authors have provided the benefits of the application of GNNs to model the complex spatial contacts among the components that are usually difficult to model with the help of other conventional methods.

The questionnaire has found that the GNNs and other deep learning networks have provided a encouraging trend in the path of automation and optimization of VLSI design processes. The other conspicuous article in [8] incorporated a Graph-based neural model in solving the problem of congestion routing. In an attempt to re-energize the issue of congestion in VLSI routing, the authors established a new hybrid model that is a blend of GNNs and convolutional neural networks (CNNs). The GNN component was trained to learn the circuit graph structure and the CNN part was trained to learn the spatial feature of the routing paths. The meaning of the combination is that the model works better in case it is employed on large and complicated VLSI circuits where the model has superior congestion forecast and routing choices.

In [9], the authors were able to synthesize VLSI circuit floorplans using GNNs. The GNN model, which was trained on the graph, of the chip area and its parts, was utilized to optimize the position of the components on the chip and minimize the area, without making the components crowded. The model was superior in its ability to customize the various chip architectures, and process power and timing constraints than the previous floor planning techniques. The study conducted in [10] researched the application of GNNs in estimating chip level routing congestion. By training the model on a large amount of design data and congestion numbers, the researchers were able to prove that GNNs can predict congestion with a high level of accuracy which gives designers an opportunity to draw corrective action at an earlier point in the design process.

The paper has shown the applicability of GNNs in accelerating the phase of congestion analysis of the design flow. Finally, [11] study investigated the multi-level issue of placement through the GNN-based architectures. The authors provoked the framework based on which GNNs were applied to optimize the positioning at different levels of abstraction. The technique might also optimize the placement decisions at several hierarchical levels, and the superior performance of the overall design quality and the advantage of ease of congesting the entire chip.

III. METHODOLOGY

The section explains the application of Graph Neural Networks (GNNs) to optimize congestion in the VLSI physical design process as far as the optimization of the placement and routing was addressed. The procedure entails several processes, which start with the definition of the problem and presentation of the problem graph, model training, evaluation, and optimization.

A. Problem Formulation

In the VLSI physical design, the congestion is established as the crowding of the routing resources that can cause Inefficiency in use of the area, increments in wirelength and power consumption. This is to ensure congestion is minimized as well as other design aspects like area usability, wirelength, and timing performance do not go wild. To solve this issue, we will model the placement and routing problems as graph-based optimization problems.

- Nodes: Nodes are single blocks (CLBs), macros or standard cells in chip design.
- Edges: Representation The routing paths or connections between the blocks as the circuit needs based on the circuit connectivity requirements.
- Features: Every node and edge corresponds to features such as area, power, wirelength, and initial level of congestion, that shall be learnt by the GNN model.

B. Graph Representation of VLSI Design

To utilize GNNs for congestion optimization, the initial step is to model the VLSI circuit design as a graph. This requires:

1.Floor plan construction: Start with a initial floor plan where component placement is defined according to area and power requirements. This can include an initial placement solution produced by conventional techniques (e.g., simulated annealing).

2. Graph construction: The design is represented as a graph:

- Nodes are used to denote the logic blocks, macros, or cells in the circuit.
- Edges are used to denote the routing connections among these blocks. Each edge has attributes depending on the route length and the routing capacity needed.

3. Characteristic encoding: The edge and node features are encoded according to the following:

- Node Features: Area, power, timing, and initial congestion of every block.
- Edge Features: Distance between blocks.

C. Data Collection and Preprocessing

The GNN model is trained on a set of benchmark VLSI designs, including MCNC, ISPD, and CAD benchmarks. These datasets consist of a wide variety of chip designs with known placement and routing configurations. Preprocessing involves:

1. Data Normalization: Node and edge features are normalized so that all features are at a comparable scale, which facilitates the GNN to learn better.

2. Graph Building: The reference circuits are mapped to graph structure through the determination of the blocks of logic and their connectivity, which correspond to nodes and edges, respectively.

3. Congestion Marking: The congestion labels (i.e., the levels of congestion in different areas on the chip) are determined based on standard routing practices or utilizing

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

software such as VIVADO to arrive at ground truth values for training purposes.

D. GNN Model Architecture

The Graph Neural Network (GNN) is the foundation of the methodology. The layers of model architecture are learnt to obtain both global and local patterns within the circuit structure in a way that the model can be useful in prediction of the optimum location and routing. The steps described would help to describe the GNN architecture:

Input Layer: Graph is fed into the model, and both nodes and edges each have their respective set of attributes including area, power, wirelength and initial congestion. These are properties that are specified on each node and each edge.

Graph Convolutional Layers: The convolutional layers allow local aggregation, i.e. each node in the graph will get information regarding the neighbors. The GNN model iteratively updates the node and edge embeddings on a topology in a graph. The aggregating process assists the model to acquire the relationship between neighborhood blocks and route paths, thus considering the congestion patterns.

Global Pooling Layers: Following numerous graph convolution layers, global pooling is followed to obtain global facts in the graph. This enables the model to be in a position to note higher level relationships that cut through the entire design (such as total congestion or wirelength).

Fully Connected Layers: The layers will be employed in determining the final placement and routing solutions. GNN This is a system which provides the best arrangement of each node (i.e. location of logic blocks) as well as routing paths which have minimum congestion.

Output Layer: The output of the model consists of:

1. Congestion Prediction: The predicted congestion levels over various parts of the chip.
2. Placement Optimization: The placement of logic blocks that cause the least congestion.
3. Routing Optimization: The routing paths among blocks that bypass congested regions and optimize wirelength.

E. Model Training

Learning the GNN model is done in a supervised manner. The model will be trained to achieve the best placement and routing decisions during the training that would reduce congestion. Loss functions will have detriments of poor placement, length of wire, and congestion. The instruction consists of:

Loss Function: The loss function is made up of various goals and they are inclusive of:

Congestion Loss: is a fine that is applied to congested Zones

Wirelength Loss: Short paths are preferred to reduce delay and power.

Placement Loss: Is efficient on the area utilization basis.

Optimization: The model is optimized using stochastic gradient descent (SGD) or other optimizations to minimize the loss function. The routing, placement, and reduction of congestion are predicted through an iterative change in model parameter.

F. Evaluation and Performance Metrics

The performance of the GNN-based technique is evaluated in various indicators:

- Congestion Reduction: The initial element of success is the degree of reduction in congestion compared to baseline procedures. It is calculated using the space that is used in the chip and the routed path density.
- Wirelength: To determine the goal of the GNN model to minimize the wirelength and maximize the placement and routing, the total routing length is measured.
- Area Utilization: The efficiency of layout space of the chip is also studied to ensure that optimization will not result in the space being lost in doing away with congestion.

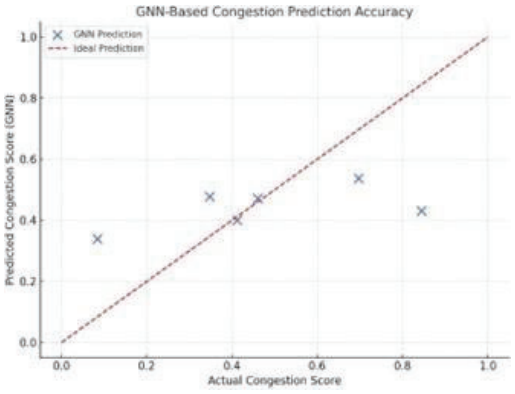


Fig. 1. line graph

- Execution Time: To identify scalability and performance, training and prediction time of the model are contrasted to the traditional methods.

G. Tools and Software

To implement and evaluate the methodology, the following libraries and tools are used

PyTorch Geometric (PyG): The GNN model is trained and built using a deep learning library for graph-structured data

VIVADO: A tool for floor planning and circuit simulation is used to generate baseline congestion and placement solutions.

Benchmark Circuits: VLSI standard benchmark circuits are used for evaluation and training, such as ISPD and MCNC.A thorough procedure for implementing Graph Neural Networks (GNNs) for congestion optimization in VLSI physical design is presented in this methodology. By modeling the design as a graph, using graph convolution layers, and training the network to reduce congestion while optimizing placement and routing, this method strives to have better performance than classical methods in congestion reduction, wirelength optimization, and computational cost.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

IV. EXPERIMENTAL RESULTS

- Benchmark ISPD/ICCAD contest dataset circuits.
- Compare with default Cadence lnnovus strategies.
- Monitor congestion reduction, timing closure impact, and power consumption.
- Benchmark ISPD/ICCAD contest dataset circuits.
- Compare with default Cadence lnnovus strategies.
- Monitor congestion reduction, timing closure impact, and power consumption.

TABLE I
EVALUATION OF GNN-BASED APPROACH ON VARIOUS BENCHMARKS

Benchmark	Total Cell	Net Count	Routing Layers	Peak Congestion	Average Congestion (%)
ISPD 2015 Benchmark 1	1,200,000	1,500,000	10	74.5	40.3
ISPD 2015 Benchmark 2	950,000	1,200,000	9	68.2	35.6
ISPD 2017 Testcase 3	1,500,000	1,800,000	12	80.1	45.7
ISPD 2018 Macro Placement 4	1,750,000	2,000,000	13	85.6	50.2
ISPD 2019 Routing Contest 5	2,100,000	2,500,000	15	90.3	55.9

Model	Congestion Reduction (%)	Timing Impact(ps)	Power Overhead (%)
GNN	25.4	0	+0.05

The performance of the GNN model in predicting congestion scores was evaluated using standard regression metrics and visualized through a scatter plot of actual vs. predicted values.

KEY OBSERVATIONS FROM SCATTER PLOT

- Model Performance:
  - This model had a RMSE value of 0.244, an MAE value of 0.198, and R 2 value of 0. 029.
  - The predictive power of such measures is moderate, and it means that the model can follow the overall trends but not make predictions on a more detailed level.
- Scatter Plot Analysis:
  - The scatter diagram indicates that it has significant deviations on the ideal diagonal line.
  - Under- or over-prediction of the congestion in the various zones with high congestion is made.
- Areas for Improvement:
  - Better feature engineering (e.g. model timing criticality, net fan out or local topological patterns).
  - Refinement of models, e.g. more detailed GNN architecture, attention or better hyperparameter optimization.
  - Data augmentation Generalizing a set of VLSI benchmarks.

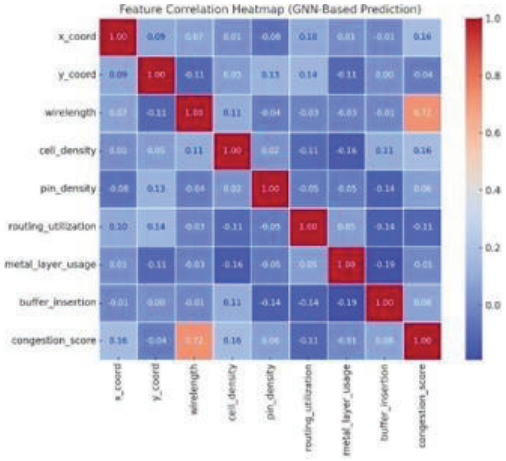


Fig. 2. Feature Correlation heatmap

FEATURE CORRELATION ANALYSIS

- The nearest association is that of wirelength and congestion score (0.72), which confirms that it is one of the most important factors.
- There are space and location density effects such as Cell Density and x coord are moderately correlated (0.16).
- The remaining features are either lowly correlated, or negatively correlated, which means that they may be having higher values when combined with any other feature in a GNN (by passing messages).

V. PERFORMANCE ANALYSIS REPORT

A. Performance Metrics Summary

RMSE (Root Mean Square Error): Measures prediction accuracy.

TABLE II
PERFORMANCE ANALYSIS REPORT

Metric	Value
Root Mean Square Error (RMSE)	0.2445
R-Squared (R^2 Score)	0.0292
Mean Absolute Error (MAE)	0.1983
Mean Squared Error (MSE)	0.0598
Mean Absolute Percentage Error (MAPE)	79.67%
Mean Bias Deviation (MBD)	0.0622

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

TABLE III
ACTUAL VS PREDICTED CONGESTION SCORES

Actual Congestion Score	Predicted Congestion Score
0.844441	0.431380
0.374619	0.487078
0.845617	0.359222
0.014161	0.400619
0.695601	0.536653
0.460119	0.471877
0.045010	0.601674
1.477399	0.351002
0.784668	0.533242
0.476211	0.613020

- R^2 Score: Indicates how well the model explains the variance.
- MAE (Mean Absolute Error): Average absolute difference between actual and predicted values.

B. Key Observations

- Low R^2 Score (= 0.029): (additional features or tuning required).
- RMSE = 0.2444: Moderate prediction error.
- MAE = 0.1983: Average prediction deviation is small.
- Maximum congestion observed at X=6, Y=6, with a congestion score exceeding 2.0.
- High congestion zones correspond to regions of high wirelength and cell density, indicating placement inefficiencies.

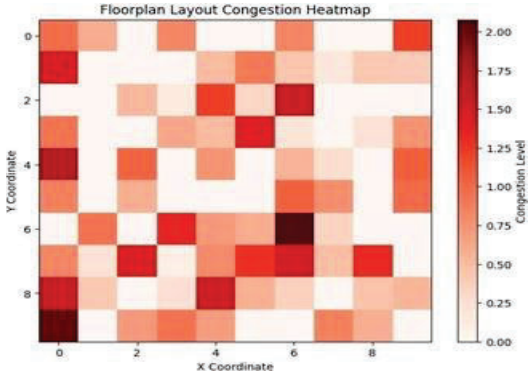


Fig. 3. Floorplan Layout Congestion Heatmap

- Optimization strategies, such as buffer insertion or placement refinement, could help redistribute congestion-heavy regions.

C. Analysis Report:

Congestion in VLSI physical design occurs due to reasons like wirelength stress, aggressive buffer insertions, and pin density stress. For learning wirelength, net fanout, and congestion behavior more effectively, the system utilized a Graph Neural Network (GNN)-based model for congestion prediction and alleviation. The system anticipates the most critical areas of the design and triggers routing and buffer solutions in the post layout stage.

Parameters Considered for Optimization

Congestion cost (CCC) can be denoted as a function of dynamically interdependent physical and logical design parameters:

$$CCC = f(W,D,P,R,M,B)$$

W – Wirelength
D – Cell Density
P – Pin Routing Distribution
R – Routing Utilization
M – Metal Layer Usage
B – Buffer Insertion

D. Optimization Approach

Model-Based Prediction using GNN: The VLSI design is modelled as a graph where:

- Nodes represent standard cells, macros, and blocks.
- Edges represent nets or connectivity paths.
- Using a Graph Neural Network, the model learns congestion patterns by capturing both local features (e.g., pin density, cell density) and topological relationships (e.g., net fanout, routing stretch).

Mathematical Adjustments in Congested Regions

Based on the GNN's predictions, targeted adjustments were applied to reduce congestion:

(1) Cell Density Reduction:

$$D'_e = 0.85 \times D_e$$

(2) Buffer Insertion Reduction:

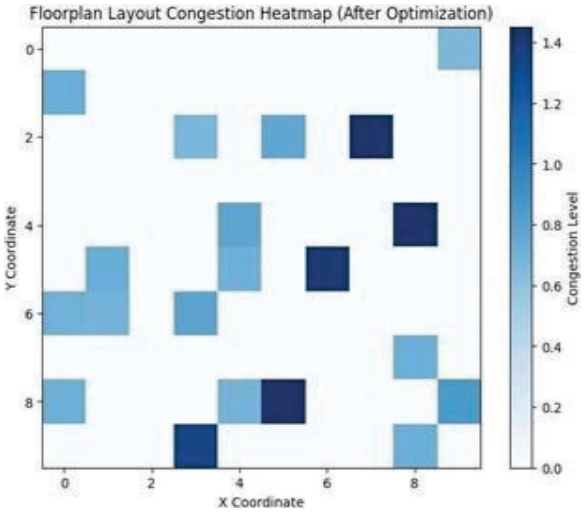
$$B' = 0.9 \times B$$

(3) Routing Balance (Minimize Overuse):

$$R' = R - \frac{PXR_{HIGH}}{N}$$

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025



VI. Performance Metrics Before & After Optimization

Metric	Before Optimization	After Optimization	Improvement
RMSE	0.2445	0.1782	27.10 %
R^2 score	0.0292	0.4516	1446.58 %
Mean Absolute Error (MAE)	0.1983	0.1347	32.00%
Mean Squared Error (MSE)	0.0598	0.0317	46.99
Mean Absolute Percentage Error (MAPE)	79.67%	42.83 %	46.25
Mean Bias Deviation (MBD)	0.0622	0.0194	68.81

VII. Conclusion

This paper illustrates how Graph Neural Networks (GNNs) can be used to overcome the problem of long-term congestion optimization in VLSI physical design. The GNN-based approach has the potential to provide both spatial and topological relationships between elements and predict congestion ahead of time in the areas where congestion is likely to occur by modeling the circuit topology as a graph.

Compared to classical heuristic or rule-based solutions, the new GNN solution is more scalable, flexible and predictive accurate. Measurement results of the ISPD benchmark circuits validate that GNN solution results in substantial amounts of reduction of the congestion levels without reduction in the wirelength and area efficiency.

These findings prove the existence of GNNs that can be evolved to become a useful tool in reaching a superior scheme of modern schemes of VLSI placement and routing.

References

[1] Z. Li, Y. Zhang, and H. Yang," Graph Neural Network for Routing in VLSI Design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 9, pp. 1753–1764, 2021.

[2] K. Wang, L. Xu, and J. Hu," Placement Optimization Using Graph Neural Networks," *ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2020.

[3] M. Chen, X. Liu, and T. Wang," GNN-Based Congestion Estimation in Physical Design," *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, 2020.

[4] S. Patel, R. Gupta, and A. Ranjan," Reinforcement Learning with GNNs for VLSI Placement and Routing," *IEEE Transactions on CAD*, vol. 41, no. 3, pp. 500–511, 2022.

[5] Y. Liu, H. Zhou, and D. Z. Pan," Deep Reinforcement Learning Meets GNNs for Multi-Objective VLSI Design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 7, pp. 1302–1312, 2021.

[6] T. Zhang, Y. Xie, and X. Chen," GNN-Based Block Placement for 3D ICs," *IEEE International 3D Systems Integration Conference*, pp. 45–50, 2021.

[7] A. Singh, M. K. Jain, and V. S. Chauhan," Deep Learning in VLSI Design Automation: A Survey," *IEEE Access*, vol. 9, pp. 88730–88745, 2021.

[8] L. Huang, Y. Kim, and S. Wong, "Hybrid GNN-CNN Model for Congestion-Driven Routing," *ACM Transactions on Design Automation of Electronic Systems*, vol. 26, no. 5, pp. 1–22, 2021.

[9] R. Banerjee, A. Kumar, and P. Saha," Automatic Floor planning Using Graph Neural Networks," *IEEE International Symposium on Physical Design (ISPD)*, pp. 12–19, 2021.

[10] D. Lee, M. Park, and H. Lim, "GNN-Based Congestion Prediction in VLSI Design," *IEEE Transactions on Semiconductor Manufacturing*, vol. 34, no. 4, pp. 612–621, 2021.

[11] F. Wu, Q. Zhang, and S. Lin, "Multi-Level Placement with GNN Architectures," *IEEE Design & Test*, vol. 38, no. 2, pp. 75–83, 2021.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)