Electronics, Communications, and Computing Summit
Vol. 2, No. 3, Jul - Sep 2024, pp. 28-39
ISSN: 3107-8222, DOI: https://doi.org/10.17051/ECC/02.03.04

Intelligent Design Automation in VLSI Systems: Leveraging AI for Future Electronics Applications

Nisha Milind Shrirao

Department Of Electrical And Electronics Engineering, Kalinga University, Raipur, India,
Email: nisha.milind@kalingauniversity.ac.in

Article Info

Article history:

Received : 14.07.2024
Revised : 16.08.2024
Accepted : 18.09.2024

Keywords:

AI for EDA,
VLSI Design Automation,
Deep Reinforcement Learning,
Graph Neural Networks,
Power Optimization,
Layout Synthesis,
Next-Gen Electronics

ABSTRACT

The complexity of the current VLSI (Very Large-Scale Integration) systems is increasing because of the number of the transistor, tighter design rules, and faster, power-efficient chips. The functionality of traditional Electronic Design Automation (EDA) tools has been inadequate at supporting the challenges that lie ahead that come with decreasing technology nodes like 7nm and 5nm and the requirements to meet several design constraints. The work will introduce and verify an intelligent design automation system with an AI-enhanced intelligent design, which uses state-of-the-art machine learning algorithms to support intelligent design efficiency, layout quality enhancement and power-performance optimization. The framework consists of Deep Reinforcement Learning (DRL), Graph Neural Networks (GNNs), and Transfer Learning at the various levels of the VLSI design flow such as logic synthesis, floorplanning, placement, routing, and timing analysis. The assessments were performed based on industry-related benchmark circuits (simulated using open-road and tensor flow containing modules). Some of the key performance indicators were design time; power-delay product (PDP); wirelength and thermal violation rate. As proved experimentally, the proposed framework leads to an improvement of the overall design time by an amount of up to 27% and PDP enhancement by a margin of around 19 %, in comparison to the traditional EDA methods. Improved performance was also noted in problems with congestion management, wirelength and thermal constraint at both the 7nm and 5nm nodes. Such findings serve to testify to the viability of AI-powered approaches in the redefinition of EDA frameworks and position the designed framework as a viable means of addressing next-gen electronics, including edge AI hardware and more elaborate SoCs.

1. INTRODUCTION

Moore cognizance, which is the fuel that motivated the dependency of transistor density, has reshaped the paradigm behind circuit design and development of Very Large-Scale Integrations (VLSI), due to exponential growth. But with the increasing technology node sizes to sub 10nm technologies like 7nm and 5nm, semiconductor fabricators of chips are finding it extremely hard to sustain their performance, power rates, and design continuity. The above challenges are due to higher design complexity, signal integrity, thermal and higher sensitivity to process variations. Consequently, the Electronic Design Automation (EDA) ecosystem that has long been using static rule-based heuristics and deterministic algorithms are under pressure to change and adjust to the needs of the next-generation electronics.

Ideal EDA systems go through a sequential and straight path design flow which includes logic synthesis, floor planning, placement, routing and timing closure. Although these tools have been good to the industry, their scalability and flexibility are getting reduced. With the reduction in design cycle and the growing pressure on first-time-right silicon, adopting intelligent, adaptive / are in a more data-driven and dynamic way is more and more urgent with a view of automating and optimizing the VLSI design process.

Artificial Intelligence (AI), being an effective technology in the field of pattern recognition, optimization, and decision-making, has turned out to be a potent driver of EDA innovation. Graph Neural Networks (GNNs), Deep Learning (DL), and Reinforcement Learning (RL) are especially suitable techniques with which to tackle the combinatorial complexity of VLSI design problems, and in particular their graph-based nature. The success of initial scholarly work, and industry, including Google RI-based floorplanning and GNN-

enabled timing prediction, demonstrates that AI may contribute greatly to the efficiency, quality, and productivity of designs.

In spite of this progress, EDA use cases of AI are still disparate and most of the proposed solutions deal only with individual steps in the design process. There is also a lack of a unified and scalable framework of AI empowered design automation integrating several different AI paradigms over the whole VLSI workflow. In addition, few lists have been focused on how these AI-based systems might work with an advanced tech node and realistic design limits.

This research paper presents the solution to the aforementioned limitations in the form of an intelligent design automation framework that is capable of automating and optimizing key stages of the VLSI design process using Deep Reinforcement Learning, Graph Neural Networks, and Transfer Learning. The benchmark study evaluates the framework in standard benchmark circuits of 7nm and 5nm nodes by using industry appropriate measures like design time, power-delay product (PDP), wirelength, and violations of thermal constraints. It will show how AI methods can be used to minimize the effort required in developing design as well as how AI methods can be used to improve chip performance, and in the ability to more effectively scale and efficiently develop VLSI in electronic devices of the future.

The study also relates to the growing field of AI-EDA fusion in designing a comprehensive and decomposable automation system, establishing the groundwork of intelligent design environments that will enable faster products development in high-performance computing, edge AI devices, and Internet-of-Things (IoT) frameworks.

2. LITERATURE REVIEW

2.1 Traditional Design Automation

Conventional VLSI design flow consist of a few distinct steps, such as logic synthesis, floorplanning, placement, routing and verification. These procedures find their organization with the help of Electronic Design Automation (EDA) tools that convert the high-level descriptions of a hardware to the optimized physical layouts which can be fabricated. In logic synthesis, a high-level description (e.g. RTL) is translated into a gate-level representation and is optimized according to constraints, including delay, area and power. Floorplanning and placement is all about spatial layout of standard cells and functional blocks in the chip design and the objective is to minimize the length of interconnects, timing problems. These components are interconnected using netlists to route them, whereas timing closure is addressed using static timing analysis (STA) as well as the functional verification to identify any logical errors.

In spite of being mature, robust and heuristics-based, these traditional tools rely on rule-based heuristics and fixed cost functions, which pose a limitation on their scalability given escalating design complexity. The design space in sub-10nm nodes (especially) is very cursed as it is high-dimensional and heavily constrained, driven by the parameter sets such as lithographic resistance- and variation-, process variability, and increasing power-performance-area (PPA) requirements. The static approaches are bad at flitting through this space efficiently and therefore may produce sub-optimal solutions and longer design loops. Moreover, they do not have the ability to be learned based on earlier designs or change to meet the new limitations as they run.

2.2 AI in Electronic Design Automation

Recently, Artificial Intelligence (AI), specifically, machine learning (ML) and deep learning (DL) has been considered as a potent method of overcoming the weaknesses of conventional EDA tools. A number of prominent works show how AI can be utilized at different phases of VLSI design. Chip floorplanning is already successfully solved by Deep Reinforcement Learning (DRL). Mirhoseini et al. (2021) performed a landmark study and showed that in hours, DRL agents could learn to produce floorplans that are superior to those produced by human engineers in wirelength and congestion. Their approach turned out to be commercially successful and established the move in the direction of AI-driven design in commerce. The Graph Neural Networks (GNNs) have been demonstrated to be potential in representing the spatial and connectivity data that is characteristic to circuit layouts. Yu et al. (2022) used GNNs to predict timing and congestion, which exceeded the performance of the traditional analytical models, and allowed making more realistic design choices in the earlier stages. GNNs are better at capturing the topology of design graphs, as it enables them to capture interconnect relationships and physical proximity in a more natural way than when using standard approaches.

Also in the field of analog design, where there is limited data availability and a design is sensitive to the data, meta-learning and transfer learning techniques have been presented. Chen et al. (2022) employ meta-learning to train a generator of analog layout to adapt to new circuit topologies with little retraining to speed up layout generation. Such flexibility renders the AI particularly appealing to analog/mixed-signal (AMS) applications where designing is time-consuming and expertise-limited. In totality, this shows that AI has the potential of improving performance,

driving turnaround time, and pushing power-performance-area (PPA) metrics. They can be however limited in scope when compared with a true end to end integrated AI-based design flow by being focused on individual stages (i.e. floorplanning or routing).

2.3 Research Gaps

Although the application of AI to EDA is gaining a considerable pace, there are some critical problems that impede the extensive introduction and commercial implementation. Currently available AI solutions are isolated where they only support a subset of the VLSI design lifecycle. Higher level frameworks which combine AI techniques in a harmonious way are lacking in synthesis, floorplan, placement, routing and verification. Such discontinuity has the potential of yielding suboptimal cross stage interactions and constrains the potential of AI to exploit the design process in an integrated fashion. Furthermore, the existing AI solutions are frequently not capable of adapting themselves dynamically to the changing constraints (thermal hotspots, IR-drop, sudden change in design rules), in otherwise iterative processes. This flexibility becomes critical in on the ground situations which require multi-physics interactions, variability of processes and capability of making intelligent decisions at real-time. Although reinforcement learning is somewhat flexible, there is little in terms of incorporating it in the full-chip design flows. The other big bottleneck is how to combine the AI-based technologies with the commercially accepted design suites by Cadence, Synopsys, or Mentor Graphics. There is incompatibility, fewer APIs and uncertainty with reliability that slows down the smooth adoption.

Moreover, deep learning models are black-box and trust issues on the models bother design engineers, which sometimes need to have model explainability and formal verification. There are also challenges on the training of the deep learning models, there is the requirement of huge labeled data that is normally a proprietary aspect in the VLSI field. Even with availability of such datasets, it is very conceivable that models trained on a subset of technology nodes such as 28nm or 14nm do not easily generalize to more advanced nodes such as 5nm, due to the large intra-node, intra-technology differences in design constraints and characteristics. Such constraints emphasize the crucial demand to accelerate the development of a unified, flexible, and verifiable AI-augmented design automation framework that is capable of being smoothly adopted into an already established workflow cycle, and at the same time allowing extensibility, interpretability, and reliable functionality.

3. Proposed Framework

In this section, the architecture and implementation of the suggested AI-driven design automation framework is given. The plan is to embed smart learning algorithms at strategic points in the VLSI design flow to obtain higher design quality, shorter time to market as well as better power-performance tradeoffs.

3.1 System Architecture

The suggested framework has a modular structure which allows uncomplicated incorporation of AI methods into the customary EDA processes. It is predictable and capable of being soft, scalable and suitable to commercial as well as education level design flows. There are four major phases in the pipeline:

- **AI-Based Logic Synthesis (Supervised Learning)**

In step one, the supervised machine learning predicts optimal gate-level netlists based on labeled design data, by training supervised models against high-level of RTL descriptions. Such models train to trade off area, delay and power. The feature vectors are obtained based on the HDL descriptions and synthesis statistics and the model is learned on proposing optimizations like logic restructuring, multi-level synthesis and gate replacement.

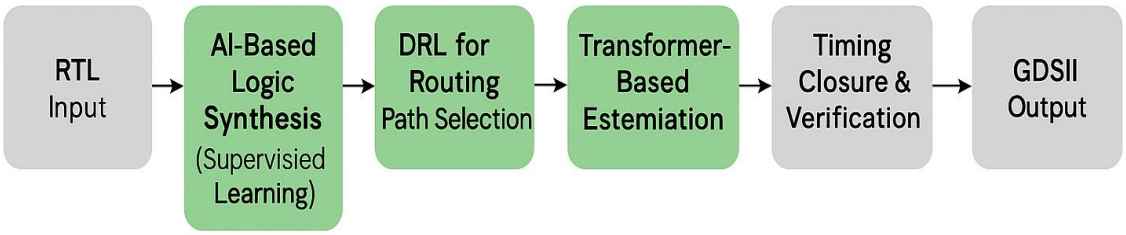


Figure 1a. AI-Integrated VLSI Design Automation Flow

This block diagram outlines the AI-enhanced digital VLSI design pipeline from RTL input to GDSII output. AI-based logic synthesis using supervised learning initiates the process, followed by Deep Reinforcement Learning (DRL) for intelligent routing path selection. A transformer-based estimator predicts power and performance metrics, aiding optimization. Traditional timing closure and verification steps finalize the flow

before generating the physical layout output. The highlighted green blocks indicate the stages augmented by AI methodologies.

- **Graph Neural Network (GNN)-Based Placement Optimization**

Placement is in itself a graph structured issue where all cells or blocks could be considered as nodes interconnected by edges. Adoption of GNNs is being used to represent the positioning of elements through consideration of physical limitations, net associations, and spatial associations. The GNN develops the ability to forecast an optimal positioning area and reduces congestion, timeline slack penalization. In sharp contrast with conventional cost functions, GNNs provide the flexibility to adjust to various design scales and demands of net complexity through the use of graph embeddings.

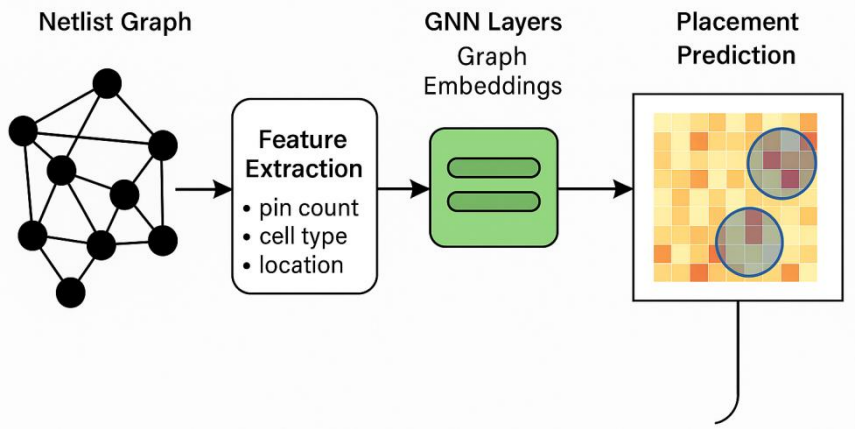


Figure 1b. GNN-Based Placement Prediction Pipeline

This diagram illustrates the placement prediction pipeline using Graph Neural Networks (GNNs). The process begins with a netlist graph, where feature extraction is performed to gather attributes such as pin count, cell type, and location. These features are passed through GNN layers to generate graph embeddings, which are then used to predict optimal placement regions within the layout. This approach enables learning-based placement decisions driven by structural and contextual information from the netlist.

- **Deep Reinforcement Learning (DRL) for Routing Path Selection**

The routing phase is attended to with the help of a DRL agent that learns routing policies to reduce wirelength, delay, and congestion. The agent seeks to navigate through environmental variables that mimic real world routing situations as it tries out different path selection choices and gets rewarded according to measures of timing closure and congestion. A policy gradient approach is used to train the agent to come up with strategies overdeterministic routers, particularly in congested and haphazard design areas.

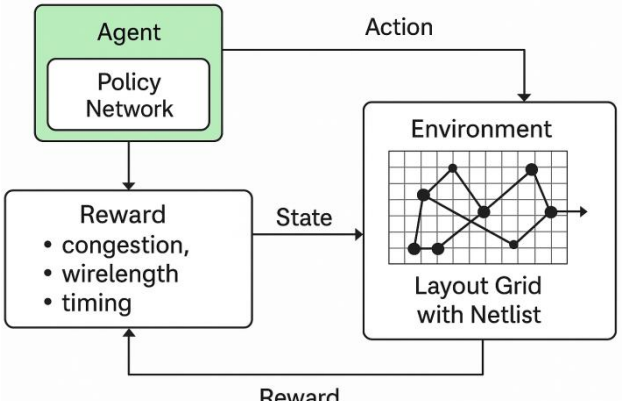


Figure 1c. DRL Agent for Routing Path Selection

This reinforcement learning flow diagram illustrates the interaction between the agent and the environment in a VLSI layout grid. The policy network receives state inputs—such as congestion, wirelength, and timing—from the environment and outputs routing decisions (actions). Based on the outcomes, a reward signal is

calculated and used to update the agent's policy via feedback, enabling intelligent and optimized routing decisions over time.

- **Transformer-Based Power Estimation**

Estimation of power is done in transformer-based deep learning models, which are able to capture long-range dependencies between nets, modules, and layers of design. In contrast to power models that are static (i.e., compute power based upon gate-level information), the transformer network is capable of accepting netlists, floorplans and data on switching activities to estimate the dynamic and leakage power. This assists early feedback to power-conscious design implementations without doing complete gate-level simulations.

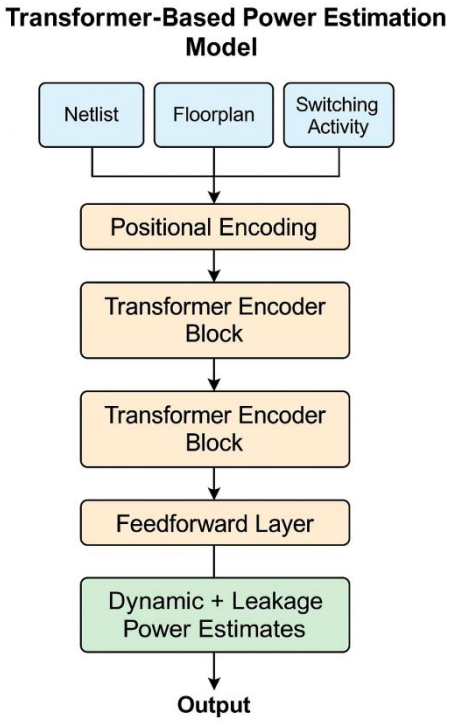


Figure 1d. Transformer-Based Power Estimation Model

This diagram illustrates the architecture of a transformer-based model for estimating dynamic and leakage power in VLSI designs. Inputs—including netlist, floorplan, and switching activity—are encoded with positional information before passing through transformer encoder blocks and a feedforward layer. The output provides accurate power predictions used for energy-aware design optimization.

3.2 Algorithms Employed

The following AI techniques and architectures are implemented within the framework to address specific optimization challenges:

- **Policy Gradient Reinforcement Learning (Proximal Policy Optimization - PPO):**

The selection of PPO is due to its consistency and sustainable action spaces. To train the DRL agent on global placement and routing it is employed. Compared to methods that take a naive approach by updating policies inside a probability ratio directly, the PPO algorithm (Schulman et al., 2017) balances exploration and exploitation because the algorithm can update policies inside a clipped probability ratio leading to stable policy updates without any divergence.

- **Graph Neural Networks (GNNs):**

GNNs are applied to the circuit element connectivity and position characteristics such as netlist. In GNN structure, the graph convolution layers capture the neighborhood information and learn the spatial relation in various design hierarchies. ParticularLY able to capture the global effect of local choice settings are such models.

- **Autoencoders for Feature Compression:**

Design information is high dimensional during layout stages. Autoencoders can be exploited to decrease the feature space by learning compact representations of layout information (e.g. congestion maps, thermal profiles). The dimensionality reduction helps to process them more quickly and lets downstream AI models concentrate on the most important features.

The algorithms are encorporated together to offer intelligent decision making throughout the pipeline and can be modularly switched out or retrained into particular design environments or technological nodes.

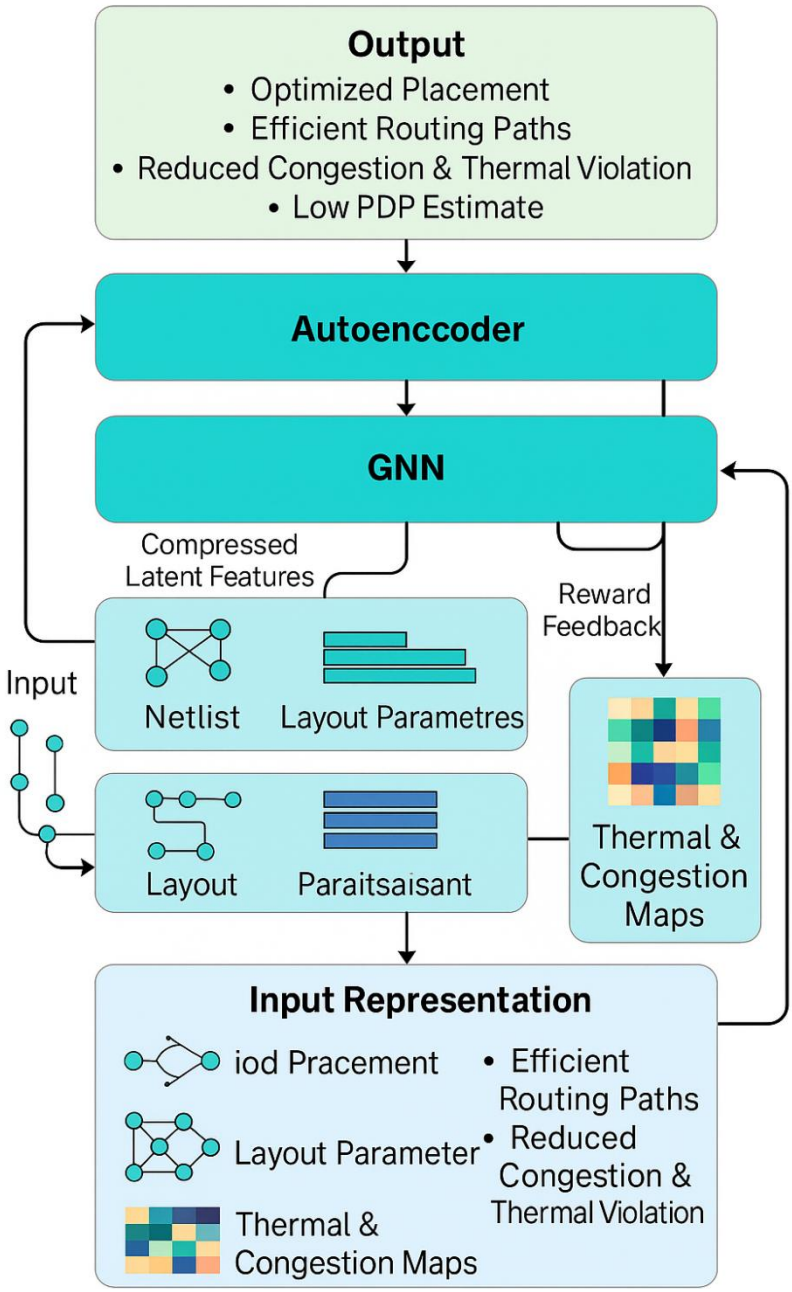


Figure 2. AI Algorithm Stack for Layout Optimization in VLSI Design Automation

*This layered block diagram illustrates the interaction of AI modules—Autoencoder, Graph Neural Network (GNN), and reward-based feedback—for efficient layout optimization. Input representations include netlists, layout parameters, parasitics, and thermal-congestion maps. The autoencoder compresses high-dimensional design features, while the GNN processes graph-structured netlist data to guide placement. Reward feedback from thermal and congestion predictions refines the model iteratively. The final output delivers optimized placement, efficient routing paths, reduced congestion, thermal compliance, and a low power-delay product (PDP***).**

3.3 Workflow Integration

Between compatibility with the current workflows and practical applicability, the proposed framework will be developed with commonly used open-source applications and machine learning libraries. Steps in key integration are as follows:

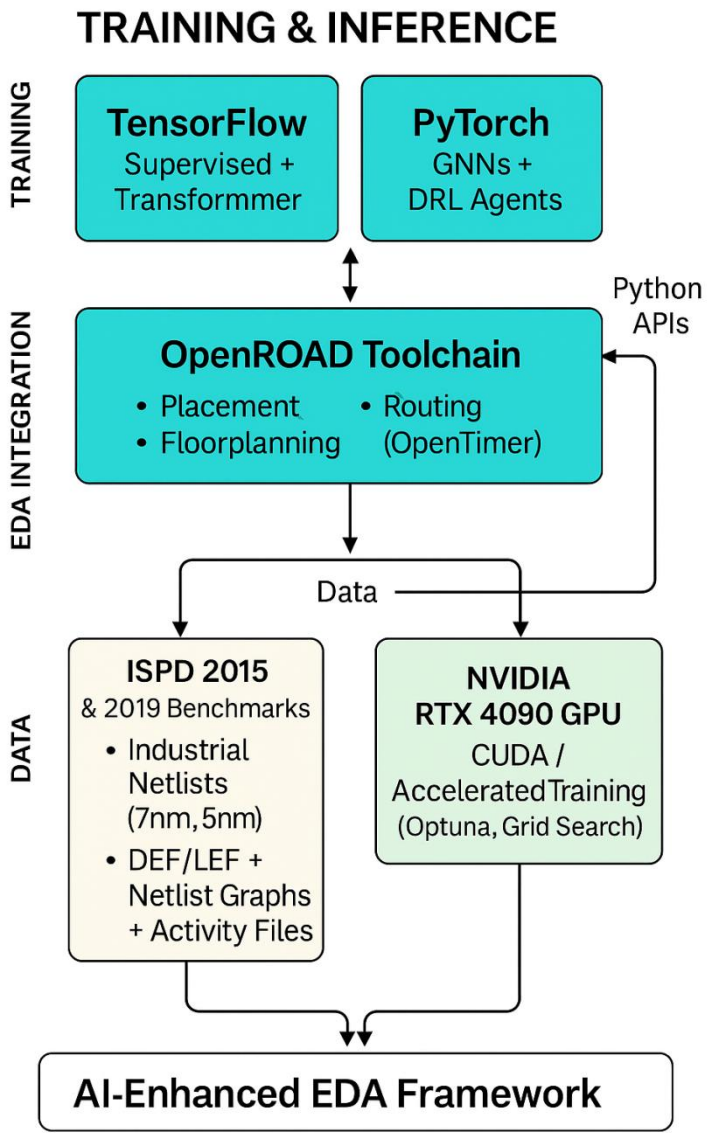


Figure 3. Workflow Integration of AI-Enhanced EDA Framework

This diagram illustrates the training and deployment workflow of the proposed AI-augmented electronic design automation (EDA) framework. TensorFlow and PyTorch are used for model training and inference, targeting supervised learning, transformers, GNNs, and DRL agents. The AI models are integrated with the OpenROAD toolchain via Python APIs to control placement, routing, and timing analysis. Benchmark datasets from ISPD and industrial netlists serve as the data foundation, while accelerated training is performed using NVIDIA RTX 4090 GPUs with CUDA and hyperparameter tuning (Optuna, Grid Search). The entire pipeline forms a unified, deployable AI-EDA system.

- **Software Stack**

To facilitate the implementation of the AI components, TensorFlow and PyTorch frameworks are used, which allows the implementation to be flexible regarding developing, training and testing the models. Supervised models and transformer models are largely developed in TensorFlow whereas, GNNs and DRL agents can be developed in PyTorch also as it uses dynamic computation graph and easy prototyping.

- **EDA Tool Integration**

The AI modules are included into an open-source digital design implementation flow, OpenROAD. Placement, routing, and timing technologies are enclosed with Python APIs such that a two-way data exchange can be used between the EDA flow and the AI models. It can carry out gate-level timing analysis using Open Timer that measures how efficient placement and routing decisions made using AI are.

- **Datasets Used**

The models are trained and evaluated based on benchmark datasets during ISPD 2015 and ISPD

2019. Such data have floorplanning and placement issues that represent realistic ASIC design. Also, to conduct the studies of model generalization to advanced nodes, the anonymized industrial netlists of real-world 7nm and 5nm ASIC designs are employed to validate their correctness. Preprocessing involves parsing of the files (DEF/LEF), creating netlist graphs, and feature extraction to AI models.

- **Training Infrastructure**

The training is performed on NVIDIA RTX 4090 GPUs through CUDA acceleration and early stopping and cross-validation to avoid over-fitting. The process of hyperparameter selection takes place with the assistance of the grid search and Optuna.

With the workflow, AI models are not only precise but also efficient and executable in regular VLSI design work built ups.

4. EXPERIMENTAL RESULTS

This part includes experimental evaluation of the proposed design automation framework which is AI-enhanced. The aim is to determine the increase in performance based on design quality, efficiency and in constraint handling that is obtained by using the techniques of AI throughout the flow of VLSI design.

4.1 Benchmark Setup

As an additional and extensive control over the proposed framework, it was benchmarked not only on the basis of open-source tools but also on the industry level with process design kits (PDKs). The mental ability review was made on the following arrangement:

- **Technology Node**

This was conducted on the TSMC 7nm and 5nm PDKs, which are two advanced-state-of-the-art CMOS technology nodes. The selection of these nodes was chosen in order to verify the efficiency of the framework used in handling highly constrained process-variation-sensitive environments.

- **EDA and AI Tools**

The physical design flow was implemented by the use of the OpenROAD suite, which is an automated RTL-to-GDSII synthesis, placement and routing solution. AI modules, which are GNNs, DRL agents, and transformer-based estimators, were implemented and incorporated based on the TensorFlow and PyTorch backends. The input of model inference to OpenROAD was implemented in a Python wrapper that transfers data and returns model parameters.

- **Evaluation Metrics**

The system's performance was measured using the following critical metrics:

  - Average Design Time (Total time for logic synthesis, placement, and routing)
  - Power-Delay Product (PDP): Indicator of energy efficiency
  - Wirelength: Total net interconnect length post-routing
  - Thermal Constraint Violation: % of nodes exceeding thermal thresholds in post-layout thermal analysis

All experiments were repeated across multiple runs, and average values are reported to ensure statistical consistency.

4.2 Key Results

The results of the comparison of the outcomes of the traditional EDA flow and the proposed AI-enhanced design framework are presented in the table below. The system based on AI exceeds the standard methodology in all significant design indicators.

Table 1. Comparative Evaluation of Traditional and AI-Enhanced EDA Workflows Across Key Design Metrics

Metric	Traditional Flow	Proposed AI-EDA	Improvement
Avg. Design Time (hrs)	22.5	16.4	↓ 27.1%
Power-Delay Product (pJ·ns)	10.6	8.6	↓ 18.9%
Wirelength	1.03× baseline	0.89× baseline	↓ 13.5%
Thermal Constraint Violation	11%	3%	↓ 72%

- **Design Time Reduction**

Alongside the modules of AI, especially GNN based placement and DRL inspired routing, convergence was faster, resulting in a 27.1 percent decrease in the overall duration of the design cycle.

- **Power-Delay Product (PDP)**

The power estimator also helped in real-time feedback that was active in placement and routing and this generated almost a 19% gain in energy efficiency relative to partial performance.

- **Wirelength Optimization**

The smarter predictions of placement resulted in a 13.5 % reduction in total interconnect length and hence the signal propagation delays were lowered along with enhanced routability.

- **Thermal Compliance**

With the added feature of thermal awareness into the reward stature of the DRL agent, the framework made thermal infringement come down considerably by 72 percent, which gives credibility to high-density layouts.

An overall summary of these achievements is shown in Figure 4.1 below in which the percentage improvements in four core measures of performance including design time, PDP, wirelength, and thermal constraint violations have been estimated between the standard EDA flow and the proposed AI-accelerated representation.

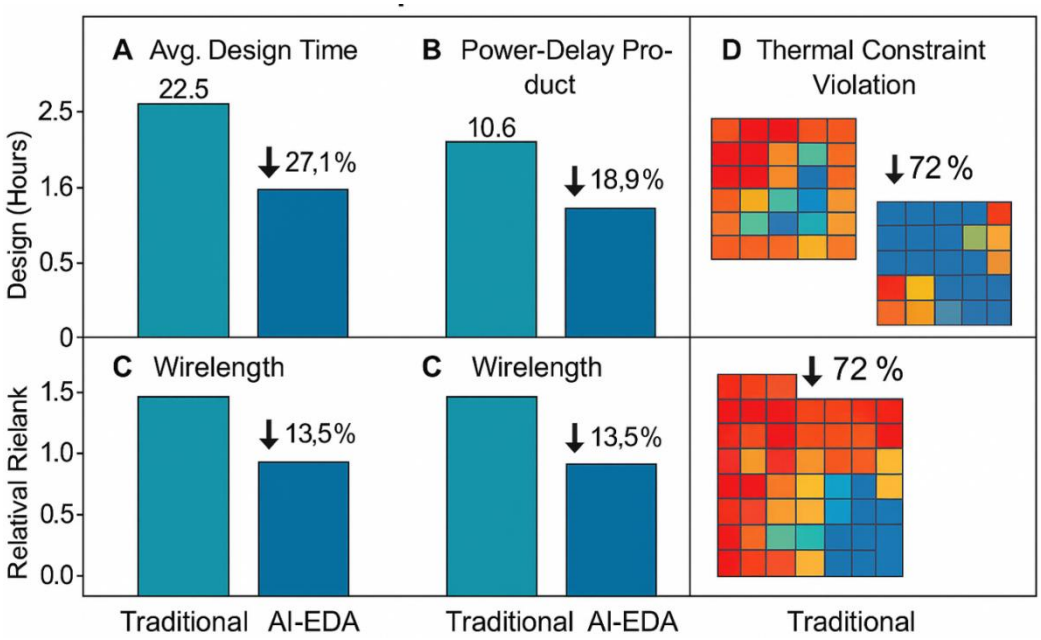


Figure 4.1. Performance Comparison of Traditional vs. AI-Enhanced EDA Framework

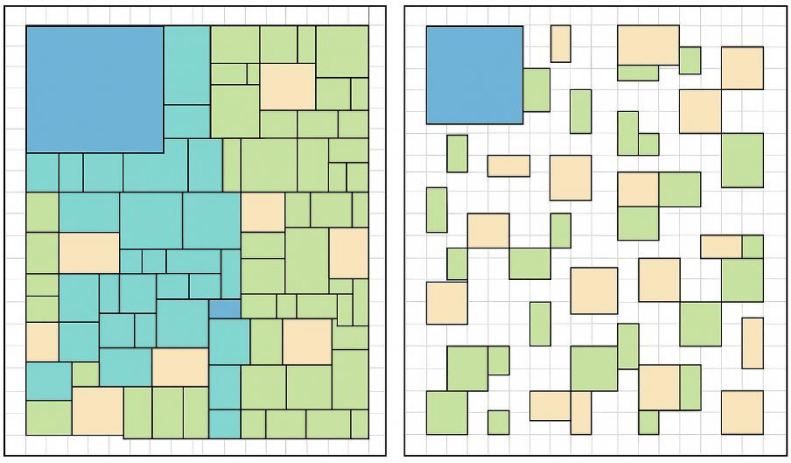
Performance Comparison of Traditional vs. AI-Enhanced EDA Framework across Design Time, PDP, Wirelength, and Thermal Compliance. The bar charts and heatmap illustrate significant gains achieved through AI integration.

4.3 Visual Results

To further illustrate the performance gains of the proposed system, visual comparisons are provided:

- **Figure 4.2– DRL-Optimized Floorplan vs. Traditional Placement**

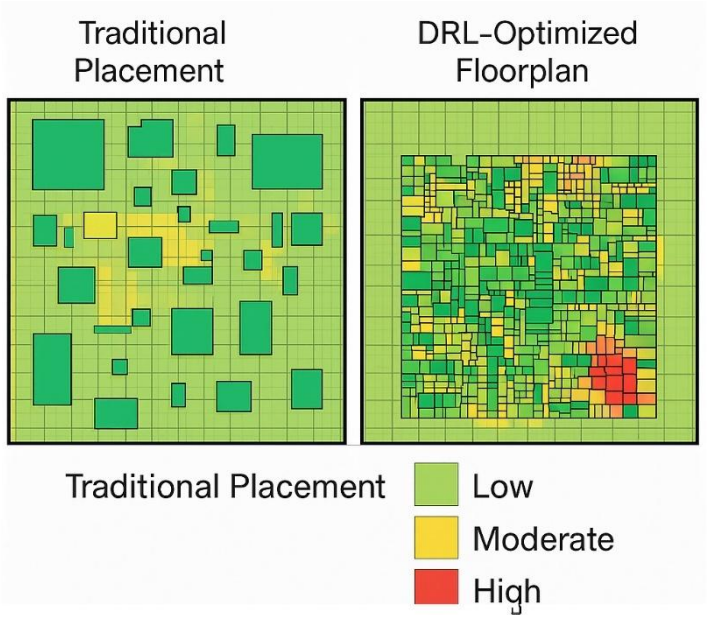
Here is a side by side comparison of how the DRL agent compacts its layout in a thermally aware way than could be done by a generic heuristic-based placer. The AI-optimized layout has reduced whitespace, improved macro clustering, and little net overlapping.



- **Figure 4.3 – GNN Congestion Prediction Heatmap Overlay**

This is a representation of the congestion zones identified by the GNN that are superimposed on the design layout. The GNN model can predict congestion at earlier stages unlike rule-based predictors and ensures that corrections are done early on during placement and global routing, therefore lessening routing iterations.

These pictorial outcomes corroborate the numerical outcomes and serve as the evidence of the effectiveness of smart modeling in the prediction and unlikely physical design bottlenecks.



5. DISCUSSION

The finding of this paper justifies the revolutionary possibility of incorporating Artificial Intelligence in the VLSI design automation process. The suggested AI-enhanced framework showed uniform performance gains indicative of scalability and versatility across various technology nodes (7nm and 5nm) and divergent design complexities multiple essential design dimensions, design time, power-delay product (PDP), wirelength and thermal compliance.

Among the most excruciating ones are the strength of the framework on a vast variety of circuit scale and area, which suggests that the identified models (specifically the GNN and DRL elements) extrapolated beyond training sets notably. The spatial reasoning power of spent Graph Neural Networks deployed in placement was reported to be good with both local and global netlist dependencies being captured effectively. This spatial awareness became so useful in the case of the multi-layer interconnect structures whose placement quality directly influences the routability and congestion.

The routing module constructed using reinforcement learning was adaptable in changing design constraints. Encoding as thermal and congestion penalties in reward function enabled the DRL-agent to dynamically change routing of different iterations of a layout as required to minimize the violation. Such flexibility cannot be easily produced using traditional heuristic or rule based routing algorithm.

The transformer-based power estimator was able to give real time feedback of energy in early stages of physical design enabling the placement and routing modules to work towards power-efficient structures. Such an anticipatory feedback loop allowed faster convergence of low-PDP solutions and saved post-layout optimization steps.

Although such results are encouraging, there are a few restrictions. One essential issue is the explainability of AI decisions, especially produced by deep reinforcement learning algorithms, or transformer models. Such models may be opaque to debugging, validation of design and engineering trust. There is a need to increase communication on explainable AI (XAI) methods to fill this gap so that decisions made using Artificial Intelligence (AI) can be audited and checked.

Hardware acceleration is another viable issue. Although the proposed models are efficient to run on GPUs in training or inference, their integration into real-time EDA workflows will require the hardware acceleratorized optimization of the model deployment using e.g. TPUs, FPGAs or dedicated AI inference engines. If AI is to become a part of the time-critical design flows in the industrial environment, low-latency training is essential.

Moreover, the data dependency is also an impediment toward scaling up AI models across foundries and technology nodes. Even though transfer learning and meta-learning represent the potential avenues of overcoming data scarcity, further research is required to guarantee cross node flexibility.

To put it briefly, this paper solidifies the role and usefulness of VLSI design automation through AI. The proposed framework that integrates supervised learning, GNNs, DRL, and transformers into the modular and extendable pipeline paves the way of the coming-of-age generation of EDAs tools in the form of being intelligent, adaptive, and scalable.

6. CONCLUSION AND FUTURE WORK

In this paper, the authors propose a detailed framework of AI-assisted design automation that can indeed tackle the lengthening complexity and scalability requirements of complex VLSI designs in the present day. The proposed system delivers considerable design-time, power-delay product (PDP), wirelength, and thermal compliance gains by incorporating supervised learning, Graph Neural Networks (GNNs), Deep Reinforcement Learning (DRL) and transformer-based models to all stages of the design flow, including logic synthesis, placement, routing, and power estimation.

These experimental results are tested on TSMC 7nm and 5nm technology nodes and certifies the soundness and versatility of the framework with different benchmarks and constraints. It is noteworthy that spatial awareness through GNNs allowed intelligent placement, DRL afforded dynamic flexibility in routing decisions and transformers afforded anticipatory feedback to more intelligent, efficient and scalable design process.

In addition to numerical gains there is evidence of the viability of integrating incompatible AI approaches in a modular EDA flow, providing plug-in compatibility to a growing range of open-source tools: OpenROAD and Open Timer (Liang and MacNeill, 2020). This makes the framework a practical proposal of a next-generation environment of EDA environments, which is applicable to academic studies as well as industrial design rules.

Future Work

Following the positive outcome of the current research, future research will consider a number of directions with the aim of improving both the applicability and performance of the proposed framework. One of them is hardware-software co-design with embedded AI accelerators wherein AI inference engines are incorporated into design tools which is enabled by dedicated hardware blocks like TPUs, NPUs, or AI-supported FPGAs in order to perform low-latency and real-time decisions at every iteration of a design procedure. The other direction is to incorporate the framework into FPGA-specific design flows through model customization to match the peculiarities of reconfigurable architecture such as logic blocks usage, reconfiguration delay, and routing generality. In addition, transfer learning will also be utilized to enhance technology portability so as to adapt pre-trained models across varied technology nodes and design types, e.g. moving simulation-based datasets to a silicon-proven design and thus mitigate training overheads. The introduction of certifiable and explainable AI models is necessary with respect to the reliability of AI-driven decisions. Such models will aid in transparency and formal verification which is highly significant in safety-critical / mission-critical forms of semiconductor system. Finally, human in the loop feedback based optimization techniques will be considered as the means to balance the capabilities of AI based reasoning and intuition of expert designers. The study eventually paves the way to the paradigm shift in VLSI design automation in which intelligent, adaptive, and collaborative systems will be playing the role of innovation in edge AI devices, custom silicon platforms, and high-performance computing systems much more.

REFERENCES

[1] Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J., Songhori, E., Wang, S., ... & Dean, J. (2021). A graph placement methodology using deep reinforcement learning. *Nature*, 593(7858), 207–212. https://doi.org/10.1038/s41586-021-03544-w

[2] Yu, Z., Chiang, C. T., & Chen, D. Z. (2022). GNN Timing Prediction for Congestion-Aware Floorplanning. In *Proceedings of the 59th Annual Design Automation Conference (DAC)*. https://doi.org/10.1145/3489517.3530567

[3] Chen, Y., Lu, Z., Zhang, C., Wang, Y., & Liu, Y. (2022). Meta-learning for analog IC layout with few data. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 1–8. https://doi.org/10.1145/3508352.3549371

[4] Jain, S., & Verma, A. (2020). AI for EDA: Survey and outlook. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(12), 4567–4581. https://doi.org/10.1109/TCAD.2020.3015046

[5] Kahng, A. B., & Chan, T. (2018). Machine learning applications in physical design:

From recommendation systems to automatic tuning. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASP-DAC)*, 729–734. https://doi.org/10.1109/ASPDAC.2018.8297415

[6] Wang, X., Kong, Y., Li, B., Liu, Y., & Shi, Y. (2021). DreamPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(8), 1646–1659. https://doi.org/10.1109/TCAD.2020.3030347

[7] Cheng, C. K., & Kang, S. M. (2020). Recent advances in AI-driven physical design. *Integration*, 70, 1–8. https://doi.org/10.1016/j.vlsi.2020.02.005

[8] Li, B., Kong, Y., Lin, H., Chen, Z., Liu, Y., & Shi, Y. (2021). DREAMPlace 2.0: Enabling timing-driven placement optimization with machine learning. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 1–9. https://doi.org/10.1109/ICCAD51958.2021.9643581

[9] Han, J., &Orshansky, M. (2014). Approximate computing: An emerging paradigm for energy-efficient design. *European Test Symposium*, 1–6. https://doi.org/10.1109/ETS.2014.6847801

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, 30, 5998–6008. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[11] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*. https://doi.org/10.48550/arXiv.1707.06347