

Artificial Intelligence-Driven ASIC Physical Design: A Review of PPA Prediction, Design Space Exploration, and Cross-Stage Optimization

Shubhashri Joshi¹, Shrikant Honade², Ganesh Patil³, Ganesh Dongare⁴

^{1,2,3,4} *Department of Electronics and Communication Engineering, CSMSS Chh. Shahu College of Engineering, Chhatrapati Sambhajinagar, India*

Abstract—Modern application-specific integrated circuit (ASIC) design has grown considerably more complex, intensifying long-standing challenges around power, performance and area (PPA) optimisation, timing closure, routing congestion and overall design turnaround time. To ease this burden, machine learning (ML) and artificial intelligence (AI) are being increasingly adopted within electronic design automation (EDA) as a way to move beyond slow, computationally expensive, iteration-heavy physical design flows. This review surveys AI-driven approaches across the ASIC physical design pipeline, with emphasis on early-stage PPA prediction, design space exploration, placement and routing optimisation, congestion estimation, and multi-objective optimisation. It further examines how machine learning, deep learning, reinforcement learning, graph neural networks and surrogate modelling techniques are being applied at distinct stages of the RTL-to-GDSII flow.

Index Terms—ASIC physical design, design space exploration, machine learning, multi-objective optimisation, PPA prediction, RTL-to-GDSII flow

I. INTRODUCTION

Application-specific integrated circuit (ASIC) physical design is a difficult and computationally demanding procedure due to the growing complexity of contemporary integrated circuits (ICs). Logic synthesis, floorplanning, placement, clock tree synthesis (CTS), routing, and physical verification are some of the interdependent steps in the RTL-to-GDSII sequence. Repeated optimisation across these steps is necessary to achieve acceptable power, performance, and area (PPA) while concurrently meeting time and routability restrictions. As a result, considerable design space exploration (DSE) is

frequently required by traditional electronic design automation (EDA) approaches, which raises computational costs and design turnaround times [1], [2].

Machine learning (ML) and artificial intelligence (AI) have shown promise in speeding up EDA and lowering reliance on repetitive full-flow runs. ML models can drive optimisation decisions and provide early estimates of physical-design metrics by learning correlations between circuit properties, design parameters, and implementation outcomes. For PPA prediction, placement, time analysis, congestion estimates, routing, and design-space exploration, previous research has examined supervised learning, deep learning, reinforcement learning (RL), and graph-based methods [3], [4]. As a result, these methods have laid a solid basis for clever and more automated physical-design approaches.

Particular focus has been placed on placement and routability optimisation. Floorplanning is presented as a sequential decision-making problem in Mirhoseini et al.'s learning-based chip-placement methodology [5]. While RouteNet demonstrated convolutional neural network-based routability prediction using placement information prior to detailed routing [7], DREAMPlace used deep learning frameworks and GPU acceleration for large-scale VLSI placement [6]. These methods demonstrate how learning-based strategies can significantly enhance specific phases of the physical design process. In order to handle increasingly complicated design choices, more recent studies have also looked into reinforcement learning, graph-based circuit representations, and AI-assisted EDA automation [8].

Despite these developments, there are still a number of significant restrictions. While practical ASIC implementation involves strong connections among synthesis, placement, timing, congestion, and routing, many existing techniques concentrate on a particular physical-design stage or optimisation target. Additionally, models that have been trained on certain circuits or technological nodes might not be able to generalise well to new designs. It is still difficult to simultaneously optimise competing goals like area, power, timing, and congestion. Other obstacles to practical deployment include dataset availability, model interpretability, cross-technology portability, reproducibility, and integration with industrial EDA tools [3], [4].

Thus, recent advancements are moving AI-assisted EDA away from isolated prediction and toward multi-objective, cross-stage, and increasingly autonomous design optimisation. Furthermore, new possibilities for intelligent circuit representation, automated tool interaction, and design-flow optimisation are being created by graph neural networks, transfer learning, reinforcement learning, and massive language models [9]. These advancements encourage a comprehensive analysis of how AI methods are changing ASIC physical architecture and identifying important research gaps..

With a focus on PPA prediction, design space exploration, placement and routing optimisation, congestion prediction, multi-objective optimisation, and cross-stage learning, this study examines AI-driven methods for ASIC physical design. The learning strategy, targeted design stage, optimisation goal, benchmark environment, benefits, and drawbacks of representative approaches are compared. Along with discussing new research approaches toward feedback-driven and autonomous EDA systems, the review highlights outstanding issues with generalisation, scalability, explainability, dataset availability, and technology portability.

The rest of this paper is structured as follows. DSE principles and ASIC physical design are introduced in Section II. Section III examines the main AI and ML methods applied to physical design. PPA optimisation, cross-stage approaches, and their applications are covered in Sections IV thru VI. In Section VII, exemplary studies are compared. After discussing research obstacles and future directions in

Sections VIII and IX, respectively, Section X concludes.

II. ASIC PHYSICAL DESIGN AND DESIGN SPACE EXPLORATION

A. ASIC Physical Design Flow

ASIC physical design converts a synthesised RTL/netlist representation into a manufacturable GDSII layout, typically progressing through synthesis, floorplanning, placement, clock tree synthesis (CTS), routing, timing optimisation, and physical verification [1], [2]. Because these stages are tightly interlinked, decisions taken early in the flow can substantially affect the quality of results obtained later on. Open-source platforms such as OpenROAD now bring these processes together within automated RTL-to-GDSII pipelines, giving researchers ready access to fast, exploratory design work and broader physical-design studies [1], [2].

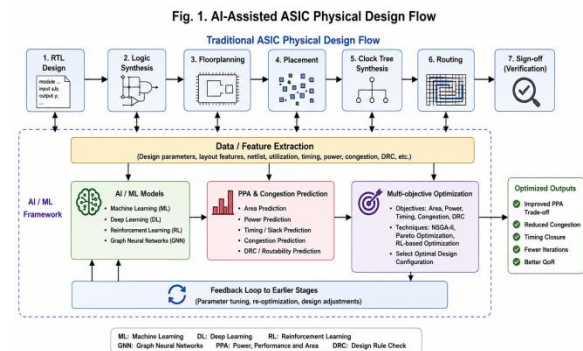


Fig. 1. AI-assisted ASIC physical design flow with feedback and optimization.

Floorplanning establishes the die/core dimensions, utilisation, macro placements, I/O positions, and power distribution structure. In order to minimise wirelength, congestion, and timing deterioration, placement then chooses appropriate places for standard cells and macros. While global and detailed routing create physical connections in accordance with technology-specific design guidelines, CTS builds the clock distribution network and manages clock latency and skew. Lastly, to guarantee implementation viability, parasitic extraction, static timing analysis, and physical verification are carried out. Floorplanning, global and detailed placement, CTS, global routing, detailed routing, parasitic

extraction, and final verification are just a few of the modules that OpenROAD offers.

Power, performance, and area (PPA), together with routability and design-rule compliance, are the main quality goals throughout this procedure. These goals are incompatible by nature. While aggressive time optimisation may add more buffers or larger cells, increasing both area and power, increasing placement density may increase routing congestion but decrease the necessary core area. Because no single implementation parameter can typically be optimised individually, physical design is really a multi-objective optimisation issue.

B. Design Space Exploration

The methodical assessment of various design configurations to find solutions that meet implementation restrictions and achieve desired PPA characteristics is known as design space exploration, or DSE. Clock period, core utilisation, placement density, synthesis technique, floorplan dimensions, routing limits, and optimisation effort are examples of parameters that may be included in the design space. Therefore, instead of optimising only one metric, multi-objective DSE approaches look for solutions that offer suitable trade-offs among competing objectives [19], [20].

For a design configuration x , the physical-design optimization problem can be expressed conceptually as

$$\max_{x \in X} [A(x), P(x), D(x), C(x)] \quad (1)$$

where $A(x)$, $P(x)$, $D(x)$, and $C(x)$ represent area, power, timing delay, and routing congestion, respectively, and X denotes the feasible configuration space.

Synthesis and physical-design techniques must be repeatedly executed for various candidate configurations in traditional DSE. Exhaustive investigation becomes computationally costly as the number of adjustable parameters increases because each evaluation may require placement, CTS, routing, and time analysis. This encourages the application of surrogate-modeling and intelligent search techniques that can predict implementation results prior to full physical realisation.

C. Need for AI-Assisted Design Space Exploration

AI-assisted DSE uses predictive models acquired from previously assessed implementations to either

replace or supplement recurring tool execution. An ML model can exclude low-quality configurations before to costly physical implementation by estimating values like area, power, time, congestion, or design feasibility given circuit characteristics and implementation factors [3], [4].

Effective AI-assisted DSE, however, needs to go beyond forecast accuracy. Models should optimise several competing goals at once, account for interactions between physical-design stages, and generalise across heterogeneous designs. Research on supervised learning, deep learning, reinforcement learning, graph-based representations, and multi-objective optimisation for physical design has been spurred by these requirements [3]–[8]. These AI and ML methods are reviewed and their applicability to various ASIC design issues is examined in the next section.

Table I. Major ASIC physical design stages and optimization objectives

Stage	Primary Function	Major Optimization Objective
Synthesis	RTL-to-gate transformation	Area, timing, power
Floorplanning	Define die/core and macro organization	Area, utilization, routability
Placement	Position cells and macros	Wirelength, timing, congestion
CTS	Construct clock network	Skew, latency, power
Routing	Establish physical interconnections	Congestion, wirelength, DRC
Sign-off	Timing and physical verification	Timing closure, design correctness

The current OpenROAD flow follows this general progression, including floorplanning, placement, CTS, global/detailed routing, parasitic extraction, and final verification.

III. AI AND MACHINE LEARNING TECHNIQUES FOR ASIC PHYSICAL DESIGN

I

In order to decrease computationally costly design iterations and increase optimisation efficiency, artificial intelligence is being used more and more in ASIC physical design. Conventional machine learning (ML), deep learning (DL), reinforcement learning (RL), and graph-based learning are the main methods [3], [4].

A. Learning by Machine

Supervised machine learning methods discover connections between physical implementation results and circuit/design parameters. While classification algorithms can estimate design feasibility or timing closure, regression models like Random Forests can forecast continuous quantities including area, power, timing, and congestion [3], [21]. Because candidate configurations may be quickly assessed prior to full physical deployment, these surrogate models are especially helpful in DSE.

B. Deep Learning

The ability to represent intricate nonlinear and spatial interactions in physical-design data is enhanced by deep learning approaches. For instance, congestion and routability prediction have been effectively implemented using convolutional neural networks (CNNs). While DREAMPlace used GPU acceleration and deep learning computational frameworks for large-scale placement [6], RouteNet demonstrated CNN-based early routability estimation [7]. Although deep models often demand larger datasets and more processing power than traditional machine learning techniques, they can achieve good prediction performance.

C. Reinforcement Learning

Physical-design optimisation is formulated as sequential decision-making in reinforcement learning. After observing the present state of the design, an agent acts and is rewarded according to the quality of the implementation. Using PPA-related

objectives, Mirhoseini et al. showed RL-based chip placement [5]. Because RL can gradually acquire optimisation strategies, it is appealing for complex DSE; nonetheless, training cost, reward formulation, repeatability, and generalisation continue to be significant issues [8].

D. Graph-Based Learning

Graph structures are naturally present in circuit netlists, where nodes are represented by cells or modules and edges by connections. Thus, structural and connection information that traditional feature vectors might miss can be captured by graph-based learning [11]. These representations offer great potential for transferring learned information across heterogeneous circuit designs and are being studied more and more for timing, placement, congestion, and other EDA prediction applications.

E. Comparative Perspective

For physical design, each AI method has unique benefits. RL facilitates sequential optimisation, graph-based learning maintains circuit structure, deep learning catches intricate nonlinear patterns, and conventional machine learning offers low-complexity and comprehensible prediction. As a result, the desired design stage, available training data, computational resources, and optimisation goals all influence the learning approach selection.

Table II. Comparison of ai techniques for asic physical design

Technique	Typical Applications	Main Advantage	Major Limitation
ML	PPA prediction, DSE	Fast and data-efficient	Feature dependence
DL/CNN	Placement, congestion	Captures complex patterns	High data/training cost
RL	Placement, parameter tuning	Sequential optimization	Expensive training
Graph learning	Timing, routing, placement	Captures circuit topology	Model complexity

IV. AI APPLICATIONS ACROSS THE RTL-TO-GDSII FLOW

In order to forecast design quality, minimise optimisation iterations, and increase implementation efficiency, AI-based techniques are being used more frequently at various phases of the RTL-to-GDSII flow [3], [4]. Synthesis optimisation, placement, time analysis, congestion prediction, and routing are only a few of their uses.

A. Logic Synthesis

ML models can predict how synthesis settings affect area, power, and timing prior to physical implementation. Therefore, the number of synthesis attempts needed during design exploration can be decreased and parameter selection can be guided by historical implementation data [3], [4].

B. Floorplanning and Placement

One of the most researched uses of AI in physical design is placement. While DREAMPlace uses GPU-accelerated optimisation to increase placement efficiency [6], RL-based methods have been studied for floorplanning and macro placement [5], [8]. While adhering to time and space restrictions, AI-assisted placement seeks to minimise wirelength and congestion.

C. Timing and PPA Prediction

Early area, power, and timing estimation is made possible using ML-based surrogate models without carrying out the full physical-design flow. Because underperforming configurations can be removed before costly placement and routing steps, such prediction is especially useful during DSE. Additionally, learning-based timing models for quick path-level timing estimation have been created [15].

D. Congestion and Routing

Placement density, connectivity, and accessible routing resources all have a significant impact on routing congestion. Before comprehensive routing, CNN-based techniques like RouteNet forecast routability and possible congestion hotspots [7]. When forecasting routing behaviour, graph-based and learning-assisted methods offer additional chances to take use of circuit connection [11]. Infrastructure for creating routing data and assessing learning-assisted

optimisation techniques is also provided by open-source routers like TritonRoute [17], [18].

E. Toward Integrated Optimization

The majority of current AI techniques focus on specific phases of the physical design process. However, because synthesis, placement, timing, and routing are closely related, optimising one step on its own might not result in the best final PPA. A shift toward multi-objective and cross-stage optimisation, in which data from later phases of implementation is utilised to inform earlier design choices, is being driven by this constraint.

Table III. AI applications across ASIC physical design

Design Stage	AI Application	Primary Objective
Synthesis	Parameter/PPA prediction	Area, power, timing
Floorplanning	Macro placement	Wirelength, area
Placement	Placement optimization	Timing, congestion
Timing	Early timing prediction	Timing closure
Routing	Congestion/routability prediction	DRC, wirelength
DSE	Surrogate-based optimization	Overall PPA

V. ML-DRIVEN PPA PREDICTION AND MULTI-OBJECTIVE OPTIMIZATION

The key parameters used to assess the quality of ASIC implementation are power, performance, and area (PPA). Comprehensive design space exploration is computationally costly since conventional PPA evaluation necessitates multiple synthesis and physical-design executions. By learning correlations between circuit attributes, implementation factors, and physical-design outputs, machine learning-based surrogate models offer an effective substitute [3], [4].

A. PPA Prediction

From RTL, synthesis, or placement-level information, supervised regression models can

estimate continuous metrics like cell area, power consumption, timing delay, and slack. Because they can capture nonlinear interactions with comparatively low training and inference complexity, algorithms like Random Forests are appealing for this usage [21]. Unsuitable configurations can be eliminated before to costly downstream implementation stages thanks to early prediction..

B. Multi-Objective Optimization

Conflicting goals are a fundamental part of ASIC physical design. For instance, high placement density may decrease core area but increase routing congestion, whereas aggressive time optimisation may increase area and power. Therefore, rather of optimising PPA metrics separately, optimisation should take into account several objectives at once. Multi-objective approaches look for configurations that offer acceptable compromises between conflicting design goals. For multi-objective exploration, genetic algorithms and Pareto-based methods like NSGA-II have been extensively employed [19], [20]. These optimisation techniques can be used in conjunction with projected PPA metrics in AI-assisted DSE to assess numerous candidate configurations without carrying out full physical implementation for each option.

A generalized multi-objective formulation can be represented as

$$F(x) = \{A(x), P(x), D(x), C(x)\} \quad (2)$$

where $A(x)$, $P(x)$, $D(x)$, and $C(x)$ denote area, power, delay, and congestion for design configuration x , respectively. The objective is to identify configurations that achieve an appropriate balance among these competing metrics while satisfying design constraints.

C. Limitations of Existing Approaches

ML-assisted PPA prediction can significantly speed up DSE, however how well it works depends on the calibre and variety of training data. Models that were trained on small circuit families might not generalise well to new designs or various technology nodes. Additionally, routing congestion and feedback from subsequent physical-design stages are not sufficiently taken into account by many current methods, which estimate various PPA metrics individually [3], [4]. Cross-stage and feedback-driven optimisation, which

are covered in the next section, are motivated by these constraints.

Table IV: ML-based PPA optimization approaches

Approach	Application	Advantage	Limitation
Regression models	PPA prediction	Fast estimation	Dataset dependent
Random Forest	PPA/DSE prediction	Low complexity	Feature dependent
Genetic algorithms	Multi-objective DSE	Global exploration	High search cost
NSGA-II	Pareto optimization	Handles conflicting objectives	Computational overhead
RL	Adaptive optimization	Sequential decision-making	Training complexity

VI. CROSS-STAGE AND FEEDBACK-DRIVEN PHYSICAL DESIGN OPTIMIZATION

Traditional AI-assisted physical design methods typically optimise each step separately. However, ASIC implementation is intrinsically sequential and interdependent: decisions about synthesis have an impact on floorplanning and placement, placement has an impact on timing and congestion, and routing outcomes may reveal issues that started earlier. As a result, isolated optimisation could yield locally effective solutions without attaining the highest level of ultimate design quality [3], [4].

A. Cross-Stage Optimization

To enhance design choices, cross-stage optimisation makes use of data produced at various RTL-to-GDSII flow phases. Intermediate metrics like utilisation, wirelength, timing slack, congestion, and routing violations can be communicated between stages rather than assessing synthesis, placement, CTS, and routing separately. This makes it possible for optimisation algorithms to take into account the long-term effects of previous parameter choices.

B. Feedback-Driven Optimization

This idea is further expanded by feedback-driven methods, which bring downstream implementation outcomes back to earlier phases. For instance, placement density, floorplan utilisation, or other relevant design characteristics can be changed in later iterations based on routing congestion found after placement or routing. In a similar vein, time data can direct modifications to placement or synthesis techniques. Such closed-loop optimisation can enhance convergence toward workable PPA solutions and minimise needless design iterations.

A generalized feedback process can be represented as $x_{k+1} = G(x_k, m_k)$ (3) where x_k represents the design configuration at iteration k , m_k represents the observed implementation metrics, and $G(\cdot)$ denotes the optimization strategy used to generate the next configuration.

C. Importance for Autonomous EDA

When developing autonomous EDA systems, cross-stage feedback is especially important. Feedback, when paired with multi-objective optimisation, reinforcement learning, and surrogate prediction, allows the design flow to adjust its choices based on real implementation results instead of solely depending on predefined heuristics. For gathering cross-stage design data and researching ML-assisted EDA approaches, open-source infrastructures like OpenROAD and CircuitOps offer helpful settings [1], [12].

Cross-stage learning is still difficult despite its potential because data produced at different physical-design stages have different dimensions and representations. Therefore, preserving model generalisation, balancing competing PPA and congestion objectives, and effectively transferring information between phases continue to be significant research challenges.

Table V. Single-stage and cross-stage AI optimization

Characteristic	Single-Stage Approach	Cross-Stage Approach
Optimization scope	Individual stage	Multiple stages
Information	Mostly local	Inter-stage

flow		
Downstream feedback	Limited/absent	Incorporated
Objectives	Often isolated	Multi-objective
Adaptability	Limited	Higher
Main challenge	Local optimization	Integration complexity

VII. COMPARATIVE ANALYSIS OF EXISTING APPROACHES

A comparative analysis of representative AI-assisted physical-design approaches is presented in Table VI. Existing studies demonstrate substantial progress in placement, routability prediction, timing estimation, and multi-objective optimization. However, most techniques remain specialized for particular stages of the physical-design flow rather than providing unified RTL-to-GDSII optimization [3], [4].

Table VI. Comparative analysis of representative AI-based EDA approaches

Ref.	Approach	Target/Application	Major Strength	Limitation
[5]	RL + graph representation	Chip placement	Learns placement policy	High training cost
[6]	GPU-based optimization	VLSI placement	High placement speed	Placement-specific
[7]	CNN	Routability prediction	Early hotspot prediction	Depends on training data
[8]	Reinforcement learning	Macro placement	Adaptive optimization	Generalization concerns
[9]	Multi-objective optimization + DREAM Place	Macro placement	PPA-aware automated tuning	Limited to placement
[1]	CNN-	Routability	Congest	Stage-

[0]	based prediction	y-driven placement	ion awareness	specific
[11]	Graph learning	EDA representation	Capture s circuit topology	Model complexity
[12]	OpenROAD + CircuitOps	ML-EDA infrastructure	Cross-stage data accessibility	Requires ML integration
[15]	Representation learning	Timing analysis	Fast timing prediction	Timing-specific
[21]	Random Forest	Prediction/regression	Low computational complexity	Feature dependent

VIII. RESEARCH CHALLENGES AND OPEN ISSUES

Even tho AI-assisted ASIC physical design has made great strides, a number of obstacles still prevent its widespread use. The growing IC design space and unsolved ML/EDA integration concerns are also identified as significant research issues in existing surveys.

A. Insufficient Training Information

The creation of datasets is computationally costly since high-quality training data necessitates repeated synthesis, placement, and routing runs. For complicated or untested circuits, small datasets can result in overfitting and lower forecast accuracy.

B. Technology Portability and Generalisation

Under various circumstances, models trained with particular benchmark designs, EDA tools, or technology nodes might not retain the same accuracy [3], [4]. Thus, creating transferable models that apply to different circuit layouts and process technologies continues to be a crucial area of study.

C. Multi-Objective Trade-offs

Congestion, timing, power, and area are closely related and frequently at odds. Enhancing one goal could undermine another. Therefore, rather than optimising each statistic separately, effective AI-based optimisation must find balanced solutions.

D. Reliability and Explainability

Many RL and deep learning techniques function as "black-box" models, making it challenging to understand their conclusions. Reliable predictions and adequate justification for certain configurations or optimisation operations are necessary for designers to apply EDA practically.

E. Cross-Stage Integration

The majority of current methods continue to concentrate on specific design phases. Because each stage generates distinct features and optimisation goals, it is still difficult to efficiently exchange information between synthesis, placement, CTS, routing, and sign-off. Therefore, cross-stage learning offers a significant chance to decrease the number of design iterations..

Table VII. Major challenges in AI-driven physical design

Challenge	Impact	Research Need
Limited datasets	Overfitting	Data-efficient learning
Generalization	Poor unseen-design accuracy	Transferable models
Technology portability	Node-dependent performance	Cross-node learning
Multi-objective trade-offs	Conflicting PPA goals	Pareto-aware optimization
Explainability	Low designer confidence	Explainable AI
Cross-stage integration	Local optimization	Feedback-driven learning

IX. EMERGING TRENDS AND FUTURE DIRECTIONS

Beyond isolated prediction, AI-assisted EDA is evolving toward more intelligent, transportable, and increasingly autonomous design approaches. Graph-based representations, foundation models, AI agents, and integrated optimisation across several stages of the chip-design flow are becoming more and more popular, according to recent research.

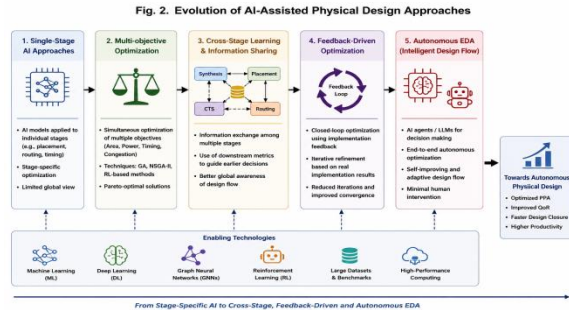


Fig. 2. Evolution of AI-assisted physical design toward autonomous EDA.

A. Circuit Learning Based on Graphs

Graphs are a natural way to depict integrated circuits, with logic cells or modules acting as nodes and electrical connections acting as edges. Thus, structural and connection information that traditional feature-based machine learning models would miss can be captured by graph neural networks (GNNs). Although scalability and EDA-tool integration are still issues, future GNN-based methods are anticipated to enhance timing, placement, routing, and congestion prediction.

B. Generalisable Models and Transfer Learning

The majority of machine learning models currently in use are trained for particular circuits, datasets, or technological nodes. By transferring knowledge between designs and technologies, transfer learning and domain adaptation may lessen the requirement for repeated model training. These methods could greatly increase AI-based DSE's scalability.

C. AI Agents and LLMs for EDA

With applications ranging from RTL creation, verification, design support, optimisation, and interaction with EDA tools, large language models are developing as another significant EDA approach

[13]. Additionally, recent research is moving away from LLM-based assistants and toward AI agents that can coordinate several design tasks and reason about design objectives.

D. Autonomous and Cross-Stage EDA

Predictive models, optimisation algorithms, and feedback mechanisms are anticipated to be included into increasingly autonomous design cycles in future EDA systems. Cross-stage frameworks can employ downstream PPA, time, and congestion data to improve earlier design choices rather than optimising synthesis, placement, and routing separately. A potential path toward quicker design completion and less manual parameter adjustment is offered by this closed-loop method.

In general, rather of depending solely on one AI method, the future of AI-driven physical design is probably going to feature hybrid learning, graph-based representations, transferable models, intelligent agents, and cross-stage feedback. Translating these developments into useful ASIC design environments would require achieving dependable generalisation, trustworthy decision-making, and smooth integration with well-established EDA tools.

Table VIII. Emerging directions in AI-driven EDA

Emerging Direction	Potential Benefit
GNN-based learning	Topology-aware circuit modeling
Transfer learning	Cross-design and cross-node generalization
LLMs	Design assistance and EDA interaction
AI agents	Automated design-flow orchestration
Cross-stage learning	Globally informed optimization
Autonomous EDA	Reduced manual design iterations

X. CONCLUSION

By facilitating quicker prediction, effective design space exploration, and automated optimisation throughout the RTL-to-GDSII loop, artificial intelligence and machine learning are progressively

changing ASIC physical design. The main AI approaches—conventional machine learning, deep learning, reinforcement learning, and graph-based learning—as well as their uses in PPA prediction, placement, time analysis, congestion estimates, routing, and multi-objective optimisation were covered in this review.

The examined research shows that AI-assisted methods can increase the effectiveness of physical-design optimisation and decrease computationally costly design iterations. The majority of current approaches, however, continue to concentrate on certain design phases and frequently show drawbacks in terms of training-data accessibility, generalisation to unseen circuits, technology-node portability, interpretability, and concurrent optimisation of competing PPA and congestion targets.

It is anticipated that more autonomous and integrated EDA approaches will be the focus of future research. Graph-based circuit representations, transfer learning, AI agents, multi-objective learning, and cross-stage feedback, in particular, offer promising avenues for creating models that can exploit dependencies between synthesis, placement, timing, and routing and adapt across heterogeneous designs. Scalable, feedback-driven, and increasingly autonomous ASIC physical-design flows with shorter design turnaround times and higher design quality may eventually be made possible by integrating these methods with open-source EDA infrastructures.

REFERENCES

- [1] T. Ajayi *et al.*, “Toward an open-source digital flow: First learnings from the OpenROAD project,” in *Proc. 56th ACM/IEEE Design Automation Conf. (DAC)*, Las Vegas, NV, USA, 2019, Art. no. 76.
- [2] T. Ajayi *et al.*, “OpenROAD: Toward a self-driving, open-source digital layout implementation tool chain,” in *Proc. Government Microcircuit Applications and Critical Technology Conf. (GOMACTech)*, 2019, pp. 1105–1110.
- [3] G. Huang, J. Hu, Y. He, J. Liu, M. Ma, Z. Shen, J. Wu, Y. Xu, H. Zhang, K. Zhong, X. Ning, Y. Ma, H. Yang, B. Yu, H. Yang, and Y. Wang, “Machine learning for electronic design automation: A survey,” *ACM Trans. Design Autom. Electron. Syst.*, vol. 26, no. 5, Art. no. 40, 2021.
- [4] T. Chen, G. L. Zhang, B. Yu, B. Li, and U. Schlichtmann, “Machine learning in advanced IC design: A methodological survey,” *IEEE Design Test*, vol. 40, no. 1, pp. 17–33, Feb. 2023.
- [5] A. Mirhoseini, A. Goldie, M. Yazgan, J. W. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, A. Nazi, J. Pak, A. Tong, K. Srinivasa, W. Hang, E. Tuncer, Q. V. Le, J. Laudon, R. Ho, R. Carpenter, and J. Dean, “A graph placement methodology for fast chip design,” *Nature*, vol. 594, no. 7862, pp. 207–212, Jun. 2021.
- [6] Y. Lin, Z. Jiang, J. Gu, W. Li, S. Dhar, H. Ren, B. Khailany, and D. Z. Pan, “DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 4, pp. 748–761, Apr. 2021.
- [7] Z. Xie, Y.-H. Huang, G.-Q. Fang, H. Ren, S.-Y. Fang, Y. Chen, and J. Hu, “RouteNet: Routability prediction for mixed-size designs using convolutional neural network,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, San Diego, CA, USA, 2018, Art. no. 80.
- [8] C.-K. Cheng, A. B. Kahng, S. Kundu, Y. Wang, and Z. Wang, “Assessment of reinforcement learning for macro placement,” in *Proc. Int. Symp. Physical Design (ISPD)*, 2023, pp. 158–166.
- [9] A. Agnesina, P. Rajvanshi, T. Yang, G. Pradipta, A. Jiao, B. Keller, B. Khailany, and H. Ren, “AutoDMP: Automated DREAMPlace-based macro placement,” in *Proc. Int. Symp. Physical Design (ISPD)*, 2023, pp. 149–157.
- [10] Y.-H. Huang, Z. Xie, G.-Q. Fang, H. Ren, S.-Y. Fang, Y. Chen, and J. Hu, “Routability-driven macro placement with embedded CNN-based prediction model,” in *Proc. Design, Automation Test Europe Conf. Exhib. (DATE)*, Florence, Italy, 2019, pp. 180–185.
- [11] Y. Ma, Z. He, W. Li, L. Zhang, and B. Yu, “Understanding graphs in EDA: From shallow to deep learning,” in *Proc. Int. Symp. Physical Design (ISPD)*, 2020, pp. 119–126.

- [12] V. A. Chhabria, W. Jiang, A. B. Kahng, R. Liang, H. Ren, S. S. Sapatnekar, and B.-Y. Wu, "OpenROAD and CircuitOps: Infrastructure for ML EDA research and education," in *Proc. IEEE 42nd VLSI Test Symp. (VTS)*, Tempe, AZ, USA, 2024.
- [13] J. Pan, G. Zhou, C.-C. Chang, I. Jacobson, J. Hu, and Y. Chen, "A survey of research in large language models for electronic design automation," *ACM Trans. Design Autom. Electron. Syst.*, vol. 30, no. 3, Art. no. 34, pp. 1–21, 2025.
- [14] K. I. Gubbi, S. A. Beheshti-Shirazi, T. D. Sheaves, S. Salehi, S. M. P. D., S. Rafatirad, A. Sasan, and H. Homayoun, "Survey of machine learning for electronic design automation," in *Proc. Great Lakes Symp. VLSI (GLSVLSI)*, 2022, pp. 513–518.
- [15] T. Chowdhury, A. Vakil, B. S. Latibari, S. A. Beheshti-Shirazi, A. Mirzaeian, X. Guo, S. M. P. D., H. Homayoun, I. Savidis, L. Zhao, and A. Sasan, "RAPTA: A hierarchical representation learning solution for real-time prediction of path-based static timing analysis," in *Proc. Great Lakes Symp. VLSI (GLSVLSI)*, 2022, pp. 493–500.
- [16] H. Yang, W. Zhong, Y. Ma, H. Geng, R. Chen, W. Chen, and B. Yu, "VLSI mask optimization: From shallow to deep learning," *Integration*, vol. 77, pp. 96–103, Mar. 2021.
- [17] A. B. Kahng, L. Wang, and B. Xu, "TritonRoute: The open-source detailed router," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 3, pp. 547–559, Mar. 2021.
- [18] A. B. Kahng, L. Wang, and B. Xu, "TritonRoute-WXL: The open-source router with integrated DRC engine," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 4, pp. 1076–1089, Apr. 2022.
- [19] M. Palesi and T. Givargis, "Multi-objective design space exploration using genetic algorithms," in *Proc. 10th Int. Symp. Hardware/Software Codesign (CODES)*, Estes Park, CO, USA, 2002, pp. 67–72.
- [20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [21] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [22] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, Feb. 2000.
- [23] I. L. Markov, F. Yang, L. Shang, and H. Zhou, "Guest editor's introduction: Machine learning for VLSI physical design," *ACM Trans. Design Autom. Electron. Syst.*, vol. 28, no. 4, Art. no. 48, 2023.
- [24] Z. Xie, J. Pan, C.-C. Chang, J. Hu, and Y. Chen, "The dark side: Security and reliability concerns in machine learning for EDA," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 4, pp. 1171–1184, Apr. 2023.