

AI-Powered EDA Tools: Current Trends and Future Directions

Abu Sayem

Electronic Engineering

Hochschule Hamm-Lippstadt

abu.sayem@stud.hshl.de

Abstract—AI-powered electronic design automation (EDA) is emerging as a key enabler for handling the growing complexity of modern integrated circuits and systems-on-chip. This paper surveys how machine learning, reinforcement learning, graph neural networks, and generative models are being integrated into different stages of the integrated-circuit design flow. After a brief review of the classical RTL-to-GDSII process, the paper discusses AI techniques for digital physical design, including reinforcement-learning-based macro placement and machine-learning-based congestion, timing, and power prediction. It then presents graph-based models and open datasets that enable learning on netlists and layouts, followed by applications in functional verification and high-level design assistance using large language models. AI-driven methods for analog and mixed-signal circuit sizing and optimization are also described. The paper concludes by outlining key benefits in design turnaround time and power-performance-area (PPA), and by highlighting open challenges in data availability, generalization, model interpretability, and integration toward truly AI-native EDA flows.

I. INTRODUCTION

The rapid advancement of semiconductor technology has led to integrated circuits (ICs) and systems-on-chip (SoCs) containing billions of transistors, pushing the boundaries of design complexity and performance requirements [1], [2]. Electronic Design Automation (EDA) tools have become essential for transforming high-level specifications into manufacturable chip layouts, encompassing the entire design flow from register-transfer level (RTL) design through physical implementation and verification [1]. However, traditional EDA algorithms face significant challenges with modern designs due to exponentially growing design spaces, stringent power-performance-area (PPA) constraints, and shrinking design cycles [3]. Artificial Intelligence (AI), particularly machine learning (ML), reinforcement learning (RL), and graph neural networks (GNNs), is emerging as a transformative force in EDA [4], [5]. Recent breakthroughs demonstrate that AI-driven approaches can achieve superior results in critical tasks such as chip placement, routing congestion prediction, and analog circuit optimization compared to conventional heuristic methods [3], [5], [6]. For instance, reinforcement learning-based placement algorithms have demonstrated 2-5% improvements in PPA metrics while reducing design time from weeks to hours [3], [7]. This seminar paper provides a comprehensive overview of AI-powered EDA tools suitable for bachelor-level hardware engineering students. It examines current applications across digital and physical design,

verification, and analog/mixed-signal domains; analyzes key enabling technologies such as GNNs and public datasets; and discusses persistent challenges including data scarcity and model trustworthiness [4], [5]. The paper concludes with future directions toward “AI-native EDA, where machine intelligence becomes integral to the design flow rather than an auxiliary component [4].

II. BACKGROUND: CLASSICAL EDA FLOW AND AI BASICS

A. Classical Digital EDA Flow

Electronic Design Automation (EDA) tools traditionally support the process of designing complex integrated circuits by automating tasks from high-level specifications to physical chip layout [1].

Figure 1 shows the classical digital EDA flow, which typically involves several main stages: specification and Register-Transfer Level (RTL) design, logic synthesis, placement and routing (physical design), and verification and sign-off [2].

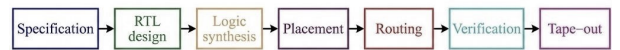


Fig. 1: Classical Digital EDA Design Flow from RTL to Tape-out [2].

During RTL design, hardware functionality is described using hardware description languages like Verilog or VHDL. Logic synthesis translates these behavioral descriptions into gate-level netlists. Placement assigns circuit components to physical locations on silicon, while routing connects these components with interconnects following design rules [1]. Finally, extensive verification ensures correctness, timing closure, power, and area constraints before tape-out [1].

Despite advancements, placement, routing, and verification remain computationally expensive as chip complexity grows exponentially [4].

B. AI and Machine Learning Concepts Relevant to EDA

Machine learning addresses these challenges by learning patterns from data [5].

Figure 2 illustrates the main AI/ML paradigms used in EDA:

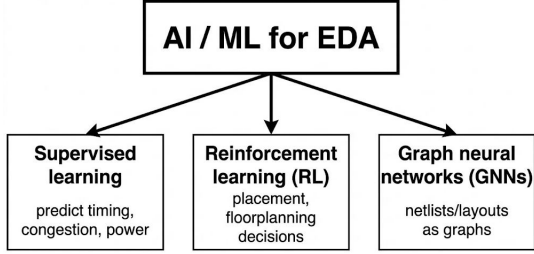


Fig. 2: AI/ML Taxonomy for EDA Applications [4].

Supervised Learning predicts outcomes like routing congestion or timing delays [5]. **Reinforcement Learning (RL)** learns optimal policies for sequential decisions like chip placement [3]. **Graph Neural Networks (GNNs)** operate on circuit graphs where nodes represent cells and edges represent connections [5].

Modern AI-powered EDA integrates these paradigms to complement classical algorithms [4].

III. AI FOR DIGITAL PHYSICAL DESIGN

Digital physical design includes key back-end steps such as floorplanning, placement, routing, and sign-off, all of which strongly impact power, performance, and area (PPA). As modern designs grow to millions of cells and macros, traditional heuristic algorithms struggle with runtime and solution quality, which has led to growing interest in AI-based methods for these stages [8], [9].

A. AI in Placement and Floorplanning (RL-Based Methods)

Chip placement can be formulated as a sequential decision-making problem where macros or cells are placed one by one onto a canvas, making it a natural candidate for reinforcement learning (RL) [10]. In a pioneering work, Mirhoseini *et al.* used deep RL with graph-based state representations to place macros for large accelerator chips; their method achieved comparable or better PPA than human experts and classical tools while reducing placement time from days to hours [10]. Follow-up research has explored hierarchical RL frameworks and RL-based macro refinement, showing that learning from expert layouts or refining existing placements can further improve wirelength and regularity on standard benchmarks [11], [12].

These studies indicate that RL can act either as a full placer or as a “regulator” that fine-tunes placements produced by traditional algorithms, providing an additional optimization layer in industrial flows [12].

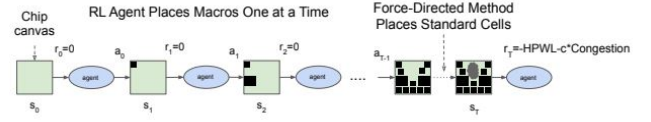


Fig. 3: Conceptual overview of an RL-based macro placement system: the agent observes the current partial placement and netlist graph, chooses macro locations, and receives rewards based on wirelength and other layout costs, inspired by Mirhoseini *et al.* [10].

B. Congestion and Routing Prediction Using Machine Learning

Routing congestion and poor routability are major causes of design iterations, since they may only become apparent late in the flow after detailed routing. Machine-learning-based predictors aim to estimate congestion much earlier, for example at the placement or even high-level synthesis stage [13]. One line of work trains regression and tree-based models to predict horizontal and vertical congestion on FPGA or ASIC designs using features extracted from placement and routing grids, achieving low prediction error and enabling early identification of congested regions [13].

More recent approaches, such as the Circuit-as-Points model and CircuitNet-based predictors, encode layouts as grids or sets of points and apply convolutional or graph-based neural networks to predict congestion and design-rule violation hotspots [14], [15]. These predictions help guide placement optimization and reduce overall turnaround time by steering designs away from highly congested solutions [15].

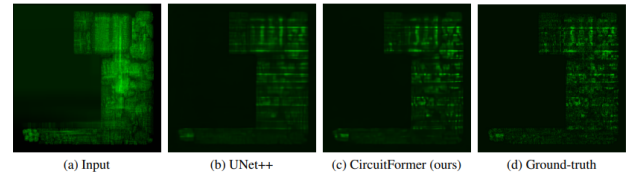


Fig. 4: Example of layout prediction results: (a) input layout representation, (b) prediction from a baseline UNet++ model, (c) prediction from an improved transformer-based model, and (d) ground-truth map, illustrating how modern ML models can better match true congestion or density patterns [14].

C. Timing and Power Estimation

Accurate timing and power analysis is essential for sign-off but can be computationally expensive when repeated many times during optimization. Recent works and surveys report a variety of ML models, including gradient-boosted trees and neural networks, that approximate cell and wire delays or path slacks using netlist and layout features [8], [16]. For instance, ML-based timing models have been explored to replace or augment traditional non-linear delay models and analytical wire-delay formulas, yielding faster evaluations

while maintaining acceptable accuracy for advanced technology nodes [16]. Similarly, ML techniques have been applied to estimate power and IR-drop from partial implementation data, enabling earlier PPA feedback and more efficient design space exploration [9].

D. Benefits of AI in Speed and Quality

Across placement, congestion prediction, and timing/power estimation, AI-based methods share two main benefits. First, they can significantly reduce turnaround time by learning surrogate models or decision policies that avoid repeated expensive simulations and analyses [10], [16]. Second, they often improve solution quality by capturing complex non-linear relationships in modern designs that are difficult for hand-crafted heuristics to exploit, leading to better PPA or more routable layouts on a wide range of benchmarks [9], [14]. As a result, AI for digital physical design is becoming a core research direction and a key component of emerging AI-powered EDA flows.

IV. GRAPH-BASED MODELS AND OPEN DATASETS

A. Circuits as Graphs and Graph Neural Networks

Digital circuits and physical layouts have a natural graph structure: cells or macros can be represented as nodes, and nets or physical interconnects as edges [9], [17]. This observation has motivated the use of graph neural networks (GNNs), which are designed to operate directly on graph data by iteratively aggregating information from neighboring nodes and edges [17]. In EDA, GNNs have been applied to tasks such as predicting routability, timing, power, and even guiding placement decisions, because they can capture both local connectivity and global structural properties of the design [5], [14]. Compared to hand-crafted features, graph-based representations allow models to generalize better across different netlists and layouts [17].

B. Open Datasets for AI in Physical Design

Progress in AI for EDA strongly depends on high-quality datasets and benchmarks that allow fair comparison of different methods. CircuitNet is one of the first open-source datasets specifically created for machine learning in VLSI CAD; it contains thousands of designs implemented with commercial tools and provides labels for tasks such as routability, IR-drop, and timing prediction [5]. The dataset includes rich netlist, placement, and routing information, enabling researchers to train and evaluate supervised and graph-based models on realistic industrial-like data [5]. Extensions and tutorials further demonstrate how CircuitNet can be used to build congestion and timing predictors, lowering the entry barrier for academic groups [15].

More recent work has proposed advanced datasets and end-to-end benchmarks that focus on evaluating the full impact of AI algorithms on design quality. For example, the ChiPBench framework benchmarks AI-based placement algorithms by measuring final PPA and routability using open-source flows, rather than only surrogate metrics [18]. Together, such datasets

and benchmarks are essential for reproducible research and for understanding how AI techniques perform across a wide range of designs and technology nodes [8].

V. AI FOR VERIFICATION AND HIGH-LEVEL DESIGN

A. Functional Verification Support

Functional verification is one of the most time-consuming phases of a chip project, often accounting for the majority of total engineering effort and simulation resources [9]. Traditional verification flows rely on manually written testbenches, constrained-random stimulus, and coverage-driven methodologies, which can require many regression cycles to reach acceptable coverage. Machine learning techniques have been proposed to make this process more efficient by learning from past simulation results and coverage data [19]. For example, ML models can rank or prioritize test cases based on their predicted probability of exposing new failures, thereby reducing the number of simulations needed to find bugs [19]. Other approaches attempt to predict coverage or failure locations from partial runs, enabling earlier termination of unpromising tests and focusing compute resources on the most informative scenarios [9].

B. Generative AI and High-Level Design Assistance

Recent progress in large language models (LLMs) and code generation has inspired their application to hardware design tasks. In this context, LLMs are used as assistants that can generate or refine register-transfer level (RTL) code, assertions, and EDA scripts from natural language descriptions [20]. Benchmark suites such as VerilogEval and related studies show that LLMs can produce syntactically correct RTL for many small modules, but logical correctness and corner cases remain challenging, especially for complex designs [20]. Beyond code generation, LLM-based tools have also been explored for debugging assistance, documentation, and automatic generation of verification properties from informal specifications [19]. Overall, generative AI has the potential to speed up high-level design and verification collateral creation, but it must be combined with rigorous formal and simulation-based checks to ensure correctness in safety-critical designs [9].

VI. AI FOR ANALOG AND MIXED-SIGNAL DESIGN

A. Challenges in Analog and Mixed-Signal Design

Unlike digital design, analog and mixed-signal (AMS) circuits such as amplifiers, filters, and RF front-ends still depend heavily on manual sizing and layout by experienced designers [21]. The design space is high-dimensional and strongly non-linear: performance metrics such as gain, bandwidth, noise, linearity, power consumption, and area interact in complex ways, and each design point typically requires expensive SPICE-level simulations [22]. Furthermore, process variations and layout-dependent parasitics must be considered early, which further increases design time and makes full automation difficult [21]. These challenges motivate the use of AI and machine learning to assist with circuit sizing, performance prediction, and design-space exploration.

B. ML-Based Analog Optimization and Surrogate Modeling

A common AI strategy for analog design is to build surrogate models that approximate the relationship between circuit parameters (e.g., transistor widths, lengths, and bias currents) and performance metrics [22]. Once trained on simulation data, these models—often neural networks or Gaussian processes—can be used inside global optimization frameworks such as evolutionary algorithms, Bayesian optimization, or reinforcement learning to search for high-quality designs with far fewer full SPICE evaluations [22], [23]. For example, recent work has demonstrated a machine-learning-driven global optimization framework that combines regression-based surrogates with genetic algorithms to efficiently optimize several benchmark analog circuits, achieving comparable or better performance than manual designs while significantly reducing simulation cost [22]. Other approaches integrate sensitivity information or gm/ID-based design methodologies with RL agents that automatically tune device sizes for operational amplifiers and RF blocks [23].

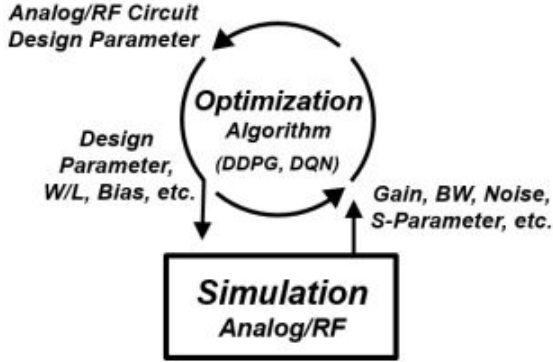


Fig. 5: Illustrative analog/RF optimization loop, where an AI-based optimization algorithm (e.g., deep deterministic policy gradient (DDPG) or deep Q-network (DQN)) selects circuit design parameters such as device sizes and bias conditions, receives performance metrics (gain, bandwidth, noise, S-parameters, etc.) from analog/RF simulations, and iteratively improves the design [22], [23].

C. Design Automation Flows for Analog and AMS

Beyond sizing, ML has also been applied to other stages of the analog and AMS flow. Some works propose end-to-end frameworks that start from a template schematic and iteratively adjust parameters using ML-guided optimization until all specifications are met [21]. Others train models to predict key performance metrics or feasibility directly from partial layouts or simplified parasitic information, enabling early screening of candidate topologies and layouts before running full extraction and simulation [21]. Surveys on ML-based AMS design report successful applications to low-noise amplifiers, comparators, and voltage references, but also emphasize that creating sufficiently rich training datasets and handling process variation remain open problems [24].

Overall, AI techniques show strong promise for reducing design iterations and enabling more systematic exploration of analog and mixed-signal design spaces, complementing rather than replacing expert knowledge [24].

VII. CHALLENGES AND FUTURE DIRECTIONS

A. Data, Generalization, and Trust

Although many promising AI techniques have been demonstrated for placement, congestion prediction, and analog optimization, several fundamental challenges limit their widespread industrial adoption. A major issue is data availability: real industrial designs and process information are highly confidential, so most public datasets cover only a small number of open-source or academic designs [5], [18]. This makes it difficult to train large models that generalize well across different architectures, technology nodes, and design styles. Even when models perform well on the datasets they were trained on, their behavior on unseen designs may degrade significantly, raising concerns about robustness and reliability in production flows [8].

Another obstacle is trust and interpretability. Many AI models used in EDA, such as deep neural networks and reinforcement learning agents, behave as black boxes, making it hard for engineers to understand why a particular placement or sizing decision was chosen [4]. Since design errors can lead to costly re-spins, practitioners need strong evidence that AI-generated results are safe and meet all constraints. Recent surveys emphasize the need for explainable models, uncertainty estimation, and tight integration of AI outputs with existing rule-based and formal verification methods to build confidence in AI-assisted flows [4], [9].

B. Towards AI-Native EDA

Looking ahead, many researchers argue that the long-term goal is not just to bolt AI components onto existing tools, but to move toward “AI-native EDA,” where machine learning is a core part of the design stack [4]. In such flows, AI models may guide decisions from system-level architecture exploration down to detailed layout, with classical algorithms providing guarantees and fine-grained optimization. This vision includes integrating graph-based models, reinforcement learning, and generative AI into unified frameworks that optimize end-to-end metrics such as PPA, reliability, and manufacturability rather than isolated surrogate objectives [17], [18].

Future research directions highlighted in recent surveys include creating larger and more diverse public datasets, developing transfer-learning and domain-adaptation techniques to reuse models across technologies, and defining standardized benchmarks for both digital and analog tasks [5], [18]. In parallel, there is growing interest in combining AI with human-in-the-loop design, where engineers interactively steer optimization and verify AI suggestions, aiming to achieve the best balance between automation, safety, and designer productivity [4], [8].

VIII. CONCLUSION

This paper has reviewed how artificial intelligence techniques are being integrated into modern electronic design automation (EDA) flows. Starting from the classical digital design flow, we discussed how reinforcement learning enables macro placement, how supervised and graph-based models support congestion prediction and timing/power estimation, and how machine-learning-based frameworks assist with analog and mixed-signal circuit optimization [5], [10], [22]. We also highlighted emerging uses of AI in functional verification and high-level design assistance, including large language models for RTL and verification collateral generation that still require rigorous checking to guarantee correctness [19], [20].

Across these domains, AI-powered methods offer two key advantages. They can shorten design turnaround time by learning surrogate models or decision policies that reduce the number of expensive analyses, and they often improve solution quality by capturing complex patterns in large design datasets that are difficult to exploit with hand-crafted heuristics [9], [14]. However, important challenges remain, including limited public training data, weak generalization to new technologies, limited interpretability of black-box models, and the practical effort required to integrate AI components into existing industrial flows [4], [5].

Future work on “AI-native EDA” aims to move beyond isolated AI add-ons toward flows in which machine learning is a core element that cooperates with traditional algorithms to optimize end-to-end metrics such as power, performance, area, and reliability [4], [17]. For future hardware engineers, a solid understanding of both conventional EDA techniques and basic AI/ML concepts will therefore be increasingly important as these technologies continue to converge in next-generation design tools and methodologies [8].

A. Commercial AI-Powered EDA Tools

Major EDA vendors have already integrated AI capabilities into production toolchains. Synopsys offers the DSO.ai, VSO.ai, and TSO.ai families, which apply machine learning and reinforcement learning to design-space exploration, physical design optimization, and verification planning in advanced-node designs [4]. Cadence provides Cerebrus Intelligent Chip Explorer, an AI-driven system that tunes place-and-route flow parameters to improve PPA for large digital blocks. Siemens EDA offers Solido ML-based tools aimed at variation-aware analog and RF design, using machine learning to reduce the number of Monte Carlo simulations required for accurate characterization. These commercial solutions demonstrate that AI-powered EDA has moved from research prototypes into industrial practice, complementing the academic techniques surveyed in this paper [8], [21].

REFERENCES

- [1] Wikipedia, “Electronic design automation,” https://en.wikipedia.org/wiki/Electronic_design_automation, 2025, *accessed* : 2025 – 12 – 02.
- [2] Synopsys, “What is electronic design automation (eda)?” <https://www.synopsys.com/glossary/what-is-electronic-design-automation.html>, 2025, *accessed*: 2025-12-02.
- [3] J. Miras *et al.*, “Chip design with deep reinforcement learning,” *Nature*, 2021, deepMind AlphaChip breakthrough.
- [4] “The dawn of ai-native eda: Opportunities and challenges,” *arXiv:2403.07257*, 2024.
- [5] Y. Chai and D. Zhao, “Circuitnet: An open-source dataset for machine learning in vlsi cad,” *Science China Information Sciences*, 2022.
- [6] “Machine-learning-based design automation for optimizing analog circuits,” *IEEE Transactions*, 2023, iEEE Xplore document 10390878.
- [7] Z. Wang *et al.*, “Benchmarking end-to-end performance of ai-based chip placement algorithms (chipbench),” *arXiv:2407.15026*, 2024.
- [8] F. Author and Others, “Survey of machine learning for electronic design automation,” *ACM Comput. Surv.*, 2022.
- [9] Z. Dai *et al.*, “Machine learning for electronic design automation: A survey,” *ACM Trans. Design Autom. Electron. Syst. or arXiv:2102.03357*, 2021.
- [10] A. Mirhoseini, A. Goldie *et al.*, “Chip placement with deep reinforcement learning,” *Proc. ISPD*, 2020, also available as *arXiv:2004.10746*.
- [11] “Hierarchical reinforcement learning for chip-macro placement in advanced nodes,” *Pattern Recogn. Lett.*, 2024, scienceDirect S0167-8655(24)00035-7.
- [12] “Reinforcement learning policy as macro regulator rather than placer,” *arXiv:2412.07167*, 2024.
- [13] F. Author and Others, “Machine learning based routing congestion prediction in fpga high-level synthesis,” in *Proc. DATE*, 2019, also available as *arXiv:1905.03852*.
- [14] F. Yang *et al.*, “Circuit as set of points,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- [15] “Circuitnet tutorial: Congestion, ir-drop, and timing prediction,” https://circuitnet.github.io/tutorial/experiment_tutorial.html, 2023, *accessed* : 2025–12–17.
- [16] B. Yu *et al.*, “Machine learning in eda: When and how,” in *MLCAD*, 2023, cUHK technical report.
- [17] F. Author and Others, “A survey of graph neural networks for electronic design automation,” *IEEE Trans. or arXiv*, 2021.
- [18] Z. Wang *et al.*, “Benchmarking end-to-end performance of ai-based chip placement algorithms (chipbench),” *arXiv or conference proceedings*, 2024.
- [19] B. Yu *et al.*, “Machine learning in eda: When and how,” in *MLCAD*, 2023.
- [20] “Ai-driven design automation,” https://en.wikipedia.org/wiki/AI-driven_design_automation, 2025, *accessed* : 2025 – 12 – 17.
- [21] F. Author and Others, “Machine-learning based analog and mixed-signal circuit design automation,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 2021.
- [22] —, “Machine learning driven global optimisation framework for analog circuit design,” *Eng. Appl. Artif. Intell.*, 2024.
- [23] A. Smith and Others, “Reinforcement learning-based analog circuit optimizer using gm/id sensitivity analysis,” in *Proc. IEEE/ACM CAD Conf.*, 2023.
- [24] B. Lee and Others, “Machine-learning-based analog circuit design: a survey,” *IEEE/ACM J.*, 2021.