© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722423

AI-Native Semiconductor Engineering: Reinforcement Learning, Graph Neural Networks, and Digital Twins as Core Design Infrastructure

ARNAV BUTAIL
Euro International High School

Abstract- ***At sub-5 nm process nodes, traditional semiconductor scaling is coming up against physical and economic limits, with Moore's Law no longer applicable. At the same time, the need for computation power is increasing and growing every day with the advent of Artificial Intelligence (AI), 5G/6G communications and autonomous systems. In this survey paper, we suggest that the semiconductor design industry is in a structural shift to an AI native paradigm, where reinforcement learning (RL), graph neural networks (GNNs), digital twins, and agentic orchestration frameworks are key and not auxiliary parts of the chip design flow. This paper: (i) reviews the physical, economic and workforce pressures driving the transition; (ii) summarizes the results of a survey of demonstrated work in RL for floor planning and placement, GNN for EDA prediction and digital-twin enabled pre-fabrication validation; (iii) categorizes placement methodologies and their evaluation protocols; (iv) presents a detailed technical decomposition of each AI-native pillar; (v) analyses the economic case for investing in AI-native design; (vi) discusses open research challenges and four tractable directions along which this research will progress; and (vii) suggests a roadmap for a four-stage curriculum for preparing the next-generation semiconductor engineer. The analysis shows that learned placement methods can be competitive with, but do not yet consistently outperform, well-designed classical placement algorithms, confirming that the need for well-designed classical placement algorithms remains crucial as AI-native tools approach production use.***

Index Terms—artificial intelligence, chip design, deep reinforcement learning, digital twin, electronic design automation, floor planning, graph neural networks, hardware-software co-design, moore's law, semiconductor engineering, simulated annealing, vlsi placement, yield prediction.

I. INTRODUCTION

Moore's Law is the empirical law of over half a century that the number of transistors per unit area in an integrated circuit doubles approximately every two years. That path has led to predictable improvements in performance, power consumption and cost, and has been the basis for the design processes, tools and skills that make up the industry today. All the circuit simulation, static timing analysis (STA) and rule-based place and route algorithms were tuned for a design world in which improvements were based on a physical dimension, that is, the size of the gates, the length of the wires, and the speed of the clock. This environment is now in the process of change.

At the very latest, the geometry of transistors introduced in the latest nodes has shrunk to the point of imposing strict physical limits on continued miniaturisation due to quantum tunnelling, manufacturing variability at the atomic level, and limitations on thermal dissipation. They carry on the scaling trend, albeit in a way that partially extends it, but not without physically moving and controlling them: Gate-all-around (GAA) nanosheet transistors are the next type to come after FinFETs at the 3 nm generation and below. The investment requirement for cutting-edge manufacturing facilities has risen in proportion, and a single state-of-the-art "fab" requires now investment in the tens of billions of dollars.

In this context, rather than slowing down, demand for computation continues to grow. The global semiconductor sales amounted to $630.5 billion in 2024, which was the first year in history to top the $600 billion mark, and total outturn for 2025 was $791.7 billion up 25.6 percent from the previous year [2] [3]. Now, 34.9 percent of global chip demand is from AI and computer applications, and industry analysts estimate that global sales of these chips will reach $1 trillion a year by 2026 [3], [4]. In 2025, the

logic product category alone accounted for $301.9 billion of the total, which is up 39.9 percent year-over-year, with the overwhelming majority of this growth attributable to demand for AI accelerators [3].

The technical difficulty is further complicated by the work force. SIA, along with Oxford Economics, estimates the U.S. semiconductor industry will require about 115,000 people to be added by 2030. With the current rate of graduates produced, about 67,000 of those jobs (58 percent of the projected growth and 80 percent of the projected new technical jobs) may remain vacant [5]. The unfilled positions include 35% engineers who have four-year degrees and 26% who have master's or PhD—these are the positions where physical design innovation occurs.

The answer to both the physical scaling challenge and the lack of an available workforce is not just about getting tools faster, it is a paradigm shift in their design approach, an AI-native design approach in which AI, specifically machine learning (ML), is built into the tool, not tacked on as an ad hoc and afterthought accelerant. This survey contributes in the following way. Section II defines the four converging forces that are fuelling the transition. In Section III, the four technology pillars of AI-native design are explored with peer reviewed evidence. In Section IV a timeline of relevant activities is presented for the 2017–26 period. Section V provides a more in-depth technical evaluation of RL-based floor planning. The placement methodologies are analyzed and compared in a structured way in Section VI. The technical architecture for the EDA prediction based on GNN is explored in greater detail in Section VII. In section VIII, the digital-twin ecosystem at the fab and system level is analysed. The economics of designing with AI is discussed in Section IX. Section X describes the changing skill needs and provides a mapping of a course. The strength of the evidence and open research challenges are discussed in section XI. Section XII provides implications for industry, academia and policy. Section XIII concludes.

II. LIMITS OF TRADITIONAL SEMICONDUCTOR SCALING

Conventional design approaches are showing their limits to the effectiveness of the design process under four converging pressures, all alone enough to call for a change in design methodology and all together asking for a radical paradigm shift.

A. Physical Limits at Advanced Nodes

For sub 5nm process nodes, quantum tunnelling currents, gate leakage and electrostatic coupling between neighbouring transistors set severe physical limits to continued scaling [6]. Increased drain-induced barrier lowering (DIBL) and threshold voltage roll-off (TVO) cause short-channel effects to become harder to control using traditional device manipulation. Similarly, current power densities are another limiting factor to consider: interconnect resistance and capacitance are square-cubic products of geometry, and are increasingly eating up the power and delay budget. As performance increases at advanced nodes, the power density is coming close to the limits of air cooling, which is prompting the industry to adopt new architectures to overcome physical limitations: 3D-IC stacking and chiplets.

The move from planar MOSFETs to FinFETs and now to gate-all-around (GAA) nanosheet transistors is indicative of the industry's need to constantly switch to 3D device architectures to continue scaling. Every such architectural change adds complexity and new simulation/verification approaches and adds to closure time to achieve tape-out closure. Rules-based tools have a structural disadvantage compared to AI-native methods because they are individually recalibrated for each of the transitions.AI native methods are structurally better than rules-based methods, which need to be recalibrated for each of the transitions.

B. Economic Limits and Capital Concentration

The up-front investment required for cutting-edge manufacturing plants has increased significantly. According to SIA, the private sector has already committed more than $500 billion in investments in semiconductors for more than 100 projects in the United States alone [2]. In fact, the consequence of

this investment focus is a structural one – fewer companies are able to operate at the bleeding edge and those that do have a huge pressure to reduce design cycle time and avoid costly respins. Respin is very costly, and can push the time-to-market by six months or more, at the 5 nm node. This imbalance of sizable tooling cost versus possibly huge rasping cost is the economic driver for AI-native verification and validation tooling.

C. Design Complexity: Combinatorial Intractability

The chip design problem of floor planning, placement and routing for a billion transistor chip is a combinatorial problem that is in principle not solvable by complete search or by purely rule-based optimization. Despite 50 years of research, chip floorplanning remains a problem that has been "defying automation" [7] according to Mirhoseini et al. For a contemporary SoC that has hundreds of macros, complicated power grid, and various scenarios of timing constraints, the state space of the floorplanning problem is too huge to search exhaustively and too irregular to use a standard convex optimisation approach. At the same time, the number of design constraints has increased, new design rules coming from advanced nodes for double patterning, local interconnect or electromigration are all to be co-optimized with power-performance-area (PPA) goals.

D. Workforce Constraints and Productivity Imperatives

The SIA/Oxford Economics report 'Chipping Away' (2023) estimates that around 67,000 specialised workers in the semiconductor industry will be lacking by 2030, 35 percent of which will be four-year degree holders and 26 percent will be master's or PhD engineers [5]. This is the skillset that fuels physical design and verification innovation. Importantly, this deficit does not close on time scales determined by merely increasing the number of students enrolled in universities – the training of a PhD usually takes four to six years, and the flow of young people into PhD programs needs to come from a limited STEM talent pool. That means AI-native design methods that augment the productivity of current engineers—which would allow a smaller group to feature a broader design area—are not simply a technical advance, but an economic must.

III. FOUR TECHNOLOGY PILLARS OF AI-NATIVE SEMICONDUCTOR DESIGN

The four pillars of technology required for the design of an AI-native semiconductor. AI-native semiconductor engineering means that the fundamental component of chip design is an AI system, not just an add-on. This transformation has four technology pillars that are grounded in peer-reviewed evidence.

A. AI-Assisted EDA and Graph Neural Networks

Circuits, netlists, and register-transfer-level (RTL) descriptions are all graph-structured data that are natural fit for Graph Neural Networks (GNNs). Survey papers list the GNN application in the EDA flow in various areas such as logic-synthesis quality-of-result (QoR) prediction, timing and power estimation, testability analysis, and analog symmetry-constraint extraction [8] and [9]. A task-aligned analysis reveals a key design principle that this holds true when the graph representation for the computation is the same as the algebraic structure of the task, such as using timing graphs for slack prediction or hyper Graphs for wirelength estimation [9].

It marks a stage of maturity in research that is noticeable in the commercial adoption of ML-based EDA. Synopsys DSO.ai uses ML-assisted design-space optimisation to find Pareto-optimal design synthesis and implementation strategies in multi-dimensional objective spaces [10]. Cadence Cerebrus uses a reinforcement-learning approach to automatically adjust the parameters for the EDA tools, providing reported power-performance-area gains from manual expert tuning [11]. Both systems are new from a tool that performs the engineer-specified recipe to one that learns what recipes are likely to work on this design—a qualitative change in the relationship of human to tool that typifies the AI-native EDA.

B. Reinforcement Learning for Floor planning, Placement, and Routing

The floor planning of chips involves arranging the various functional macro blocks on a chip in such a way that the area, power consumption, and timing are optimised and is a sequential combinatorial optimisation problem. The first demonstration of RL applied to this problem is the 2021 paper by Mirhoseini, Goldie et al. in Nature [7] which represents the floor planning problem as a Markov Decision Process (MDP) and uses edge-based graph convolutional policy (GCP) to train the policy. As reported, design of their so-called Tensor Processing Unit (TPU) accelerator chips at Google uses the method to create floorplans in less than 6 hours which are superior or comparable to floorplans produced by humans, according to proxy PPA metrics.

Complementary work shows that RL is one of the ML approaches for physical design. For analytical global placement, Lin et al.'s DREAMPlace treats placement as a problem of training a neural network to produce a placement solution, and obtains large speedups over CPU-based analytical placers by utilizing GPUs parallelism [13]. Cheng and Yan's DeepPR are a joint placement and routing system which reduce routed-wirelengths by 1.4 to 20.1 percent over a sequential placement-then-routing pipeline on eight benchmark circuits of ISPD-2005 [14].

C. Digital Twins for Pre-Fabrication and Manufacturing Validation

The semiconductor industry or its use of digital twins can be divided into two categories. Functional simulation and formal verification have been a type of digital twinning for chip logic for decades at the design level. The use of digital-twin modelling in manufacturing is what's new. The Siemens Calibre platform is a fusion of fabrication history and design content to develop predictive models based on ML for yield and process monitoring [15]. Digital twins of the process equipment are employed by equipment vendors such as Tokyo Electron (TEL) and Lam Research to explore the process-condition search space that has increased from about $5x10^3$ candidate recipes to $10x10^6$ to $10x10^8$ [16] and [17].

D. Agentic Semiconductor Workflows

The most promising evolution is agentic workflows: coordinated AI-agent systems that can not only manage design and verification but also optimise and take charge of some manufacturing planning across the design flow. Existing studies in this area are largely exploratory: the frameworks that have been developed for combining generative and predictive AI for manufacturing digital twins suggest that agentic coordination is a planned evolution rather than a present skill [19]. This paper proposes agentic workflows as the logical next step after the individual maturity of RL-based physical design, GNN-based prediction, and digital-twin validation with novel trade-offs still under the control of human engineers.

IV. CHRONOLOGY OF AI-NATIVE SEMICONDUCTOR DESIGN

Table I The most recent year of agentic exploration has now added a new paradigm to our suite of methods, namely AI-native approaches, and this is a decade-scale research trajectory that can be traced back from the earliest experiments in RL.

TABLE I SELECTED MILESTONES IN AI-NATIVE SEMICONDUCTOR DESIGN (2017–2026)

Year	Milestone / Reference
2017	Mirhoseini et al. frame device placement for neural-network computation graphs as a reinforcement learning problem—an early precursor to RL-based physical design [7].
2019–20	Lin et al. introduce DREAMPlace, casting analytical global placement as equivalent to neural-network training and exploiting GPU acceleration for significant speedup over CPU-based placers [13].
2021	Mirhoseini, Goldie et al. publish “A Graph Placement Methodology for Fast Chip Design” in Nature (vol. 594, pp. 207–212), demonstrating an RL policy generating TPU floorplans in under six hours [7].
2021	Google open-sources the methodology as Circuit Training, enabling community reproduction and scrutiny [12].

Year	Milestone / Reference
2021	Cheng and Yan present DeepPR (NeurIPS 2021), jointly training placement and routing policies and reporting 1.4%–20.1% routed-wirelength gains over a sequential pipeline [14].
2021–22	Multiple surveys formalise GNN applications across the EDA flow spanning logic synthesis, placement, timing, and power estimation [8], [9].
2023	Cheng, Kahng et al. circulate "An Updated Assessment of Reinforcement Learning for Macro Placement," subsequently accepted in IEEE TCAD, reporting that a strengthened SA baseline outperforms Circuit Training on true post-route PPA metrics [20].
2024–25	Commercial ML-driven EDA products (Synopsys DSO.ai, Cadence Cerebrus) enter broader adoption; digital-twin platforms extend from fab equipment into system-level design co-simulation [10], [11], [15]–[18].
2025–26	Research shifts toward diffusion-based and LLM-assisted EDA workflows; task-aligned analyses systematise when GNNs genuinely outperform classical proxies [9], [21].

V. CASE ILLUSTRATION: RL-BASED FLOORPLANNING (MIRHOSEINI ET AL., 2021)

To ground the survey in a concrete technical example, this section provides a detailed account of the best-documented public case of AI-native physical design.

A. Problem Formulation as a Markov Decision Process

Mirhoseini et al. represent the chip floor planning problem as a sequential decision-making problem which can be solved with reinforcement learning [7]. Each state is a partial layout of the chip canvas (as a graph over the netlist). The agent makes decisions at each step, to choose the location of one macro block, a large functional unit of either a memory block or a processing core. The length of the episode is terminated when all of the macros are placed, and a reward signal is computed based on surrogate measures of wirelength (a proxy for performance), congestion (a proxy for routability) and density (a proxy for area). The reward is only provided at the end of the episode, much like the standard sparse-reward MDP that makes learning an efficient policy a difficult task, and justifies the use of pre-training methods to start the policy from other designs they may be familiar with.

B. Graph-Based State Representation and Transfer Learning

One of the key ideas of the paper is an edge-based graph convolutional neural network (GCN) that uses the chip netlist as the state representation for the policy [7]. Each node in the graph is a circuit macro or a group of standard cells; edges represent electrical connections. The GCN combines the information within the graph to create a fixed dimensional embedding of the graph in the form of a netlist to which the placement policy will feed to determine the position of the next macro. This design is also important as it allows the learned policy to be transferred between the various chip designs: the policy can be transferred to new netlists without retraining; it can take advantage of structural priors embedded in the weights of its network. This is used to say that the policy is capable of improving itself as it continues to get more experience on more chip blocks, which is fundamentally different than per-instance optimisation, as done by simulated annealing or analytical placement.

C. Reported Outcomes and Production Deployment

The method produces the floorplans in less than 6 hours, which is better or at least comparable with the floorplans created by people on PPA proxy metrics, while the traditional method would take months of effort from a specialist [7]. The authors say they used the methodology to design the Google's Tensor Processing Unit (TPU) accelerator chips, one of the few publicly revealed copious uses of an RL-based approach in a production chip. Google later released the methodology as an open source project called Circuit Training, which was eventually renamed AlphaChip, which allowed community reproduction and scrutiny [12]. The contested comparative claims

are not the only important contribution that this open release will make to the field.

D. Pedagogical and Research Significance

Regardless of the final fate of the methodological debate (Section VI-A), this case is noteworthy for its ability to show that physical design can be formally represented as an RL problem at scale, as it lays the foundation for graph-based state representations to be the natural formalism for circuit-structured data in learned policies, and as it opens the Circuit Training / AlphaChip policy release to the research community for study, extension, and critique without requiring access to proprietary EDA infrastructure [12]. The Circuit Training repository also contains the Ariane RISC-V benchmark that was widely used in the following evaluation studies, with which it is directly comparable in the debate reported in Section VI.

VI. COMPARATIVE ANALYSIS OF PLACEMENT METHODOLOGIES

Table II assembles reported, verifiable results from four peer-reviewed studies that evaluate placement methods against each other, enabling a cross-protocol reading that reading individual papers in isolation does not permit.

TABLE II COMPARATIVE PLACEMENT METHODOLOGY RESULTS ACROSS STUDIES

Study	Method(s) Compared	Benchmark / Metric	Reported Outcome
Mirhoseini et al., Nature 2021 [7]	RL policy (edge-based GCN) vs. human expert layouts	Google TPU-related blocks; proxy PPA metrics (pre-route)	Floorplans generated in <6 h; comparable to or better than human layouts on power, performance, and area proxies.
Cheng, Kahng et al., IEEE TCAD 2025 [20]	Strengthened SA baseline vs. pre-trained Circuit Training / AlphaChip	Ariane RISC-V core, 7 nm TSMC and ASAP7; true post-route PPA via commercial P&R tool	SA outperforms pre-trained CT on both proxy and true post-route metrics; finding stable across from-scratch and fine-tuned settings.
Cheng & Yan, NeurIPS 2021 (DeepPR) [14]	Joint RL placement+routing vs. separate pipeline	ISPD-2005 benchmark suite (8 circuits); final routed wirelength	Routed-wirelength reductions of 1.4%–20.1% (mean ≈8.4%) versus separate pipeline across 8 circuits.
MacroDiff+ (2025/26) [21]	Generative diffusion vs. RL (MaskPlace) and BBO (WireMask-BBO)	Standard macro placement benchmarks; Half-Perimeter Wirelength (HPWL)	Diffusion reduces HPWL by ≈6% vs. RL and BBO baselines; RL and BBO fail to converge on largest test circuits.

A. Cross-Protocol Findings

Table II provides three patterns. First, how one evaluates, determines the conclusions, more than the choice of algorithms. The metrics are proxy metrics for the comparison from Mirhoseini et al., and true post route metrics from a commercial tool from Cheng, Kahng et al. for the comparison of the same RL system and a deliberately strengthened SA baseline. These are different experiments: the results are scope limited, not directly contradictory. The 2021 Nature answer to the question “can an RL policy produce layouts competitive with an experienced human engineer, measured by pre-route proxies?” is “yes”. Both answers correctly address the question “does pre-trained Circuit Training outperform a well-tuned SA baseline on true post-route metrics for an open benchmark circuit?” within the confines of their respective experiments.

Secondly, classical baseline strength is a key variable. If a study is spending resources on a well-tuned SA baseline, with a contemporary metaheuristic, multi-threading, and multiple restarts (Cheng/Kahng), then that baseline is competitive to or better than learned methods. For the non-adversarial classical comparison or the weak one, learned methods (RL in DeepPR, diffusion in MacroDiff+) clearly outperform the classical

approaches. This is indicative that the apparent disagreement in the literature may be more due to the level of engineering effort at baseline than to the difference between the learning-based approach and the classical approach.

Third, with genuine measured improvements, there are only incremental changes. DeepPR achieves gains of 10 to 20 percent over a reasonable but not adversarial baseline, while MacroDiff+ achieves gains of ~6 percent over strong learned and search-based baselines. These are engineering-related margins which may actually have a relevant impact on chip PPA in production but are not to the magnitude of the popular AI 'revolutionising' chip design story.

B. A Revised, More Defensible Claim

The defensible conclusion from the aggregate evidence is narrower and more useful than the headline framing of the 2021 Nature result: ML-based placement methods—including RL, joint learning, and generative approaches—are competitive with strong classical baselines on several benchmark families and metrics, and can outperform weaker or non-adversarial baselines by meaningful margins, but do not yet show a settled, protocol-independent advantage over the best available classical methods when those methods receive comparable engineering investment. This is still a genuinely important finding—it establishes learned methods as legitimate, competitive tools in the physical designer's toolkit—but it demands rigorous benchmarking as the condition for making stronger claims.

VII. TECHNICAL ARCHITECTURE OF GNN-BASED EDA PREDICTION

Section III-A introduced GNNs for EDA at a survey level. This section examines the technical architecture and design choices that determine where GNNs add genuine value versus where classical methods remain preferred.

A. Graph Representations in EDA

The choice of graph representation determines what structural information a GNN can exploit. Three representations appear most frequently in EDA literature. (1) Hypergraphs, in which a single net can connect more than two nodes, are the natural representation of a circuit netlist; they are typically converted to clique or star expansions for processing by standard GNN layers. (2) Directed acyclic graphs (DAGs) represent the logical data-flow in an RTL design and are the natural input for synthesis QoR prediction: a GNN operating on a synthesis DAG can propagate delay estimates forward from inputs to outputs, mimicking the structure of STA. (3) Timing graphs, in which edges carry resistance, capacitance, and transition-time annotations, support interconnect-delay prediction and timing slack estimation. The task-aligned analysis of [9] demonstrates empirically that GNNs achieve their largest improvements over classical proxies when the graph structure used for computation matches the algebraic structure of the task: a GNN operating on a timing graph for slack prediction outperforms a GNN operating on a structural netlist for the same task, because the timing graph encodes the causal flow of signal delay.

B. Key GNN Architectures for EDA

Graph Convolutional Networks (GCNs) aggregate neighbour features with a learned linear transformation and are the most common baseline architecture in EDA applications [8]. Graph Attention Networks (GATs) extend this with learned attention weights over neighbours, allowing the network to focus on the most informative connections—a property particularly useful in congestion prediction, where a small number of critical nets dominate the routing difficulty. Heterogeneous GNNs model circuits with multiple node and edge types (gates, pins, nets, macros, standard cells), maintaining separate message-passing functions for each type. Edge-based GCNs, as used by Mirhoseini et al. [7], assign features to edges rather than solely to nodes, allowing the model to encode connectivity properties directly rather than inferring them from node aggregation.

C. Open Challenges: Stage Leakage and Proxy-to-Signoff Gap

The task-aligned analysis of [9] identifies two challenges that limit GNN deployment in production EDA flows. Stage leakage refers to the error that propagates between pipeline stages: a GNN trained to

predict post-synthesis wire length may receive as input a netlist that the synthesis tool has optimised in ways the GNN's training data does not reflect, causing distribution shift. The proxy-to-signoff gap refers to the observation that a model that accurately predicts a proxy metric (e.g., pre-route estimated wire length) does not necessarily rank designs correctly by the final signoff metric (e.g., post-route timing closure with full design-rule checking). Both challenges are active research areas, and neither has been solved in a way that generalises robustly across PDKs and design families. Until they are resolved, GNN-based prediction is best deployed as a fast screening and ranking tool rather than as a replacement for full signoff analysis.

VIII. THE DIGITAL TWIN ECOSYSTEM IN SEMICONDUCTOR MANUFACTURING

This section examines the digital-twin ecosystem in greater depth, distinguishing between design-level, equipment-level, and system-level deployment contexts and discussing the technical infrastructure that enables each. D. Caution on Reported ROI Figures

A. Design-Level Digital Twins

In the design realm, however, there is a type of digital twinning that has been around for decades, using hardware description languages (HDLs) and pre-silicon simulation. By definition a register transfer level (RTL) simulation model is a digital twin of the functional behaviour of the chip – it has the same state transitions as the silicon, and any differences between the model and the silicon represent design and/or verification errors. The key difference between the formulation of classical digital twins and modern AI-native design-level digital twins is the use of machine-learning elements to generate physical properties – such as timing, power, area, thermal hotspots – from RTL or gate levels before full physical implementation. These predictive models can enable design teams to take implementation decisions much earlier in the design flow, which means they'll have fewer expensive iteration rounds between implementation and sign-off.

B. Equipment-Level Digital Twins

The most advanced industrial applications are at the equipment level with digital twins. Tokyo Electron (TEL) has developed digital twins of the process equipment to explore process-condition search spaces, ranging from about 5,000 candidate recipes to 10^6 to 10^8 [16]. Physical experiments are not feasible at this scale: it costs thousands of dollars to run a single process on a 300 mm wafer, and takes several hours. The benefit of a digital twin that can accurately predict the etch rate, uniformity and defect density from a proposed recipe without using wafer time is massive. Lam Research also claims to have digital twins that incorporate real-time sensor data from production equipment to allow predictive maintenance and detection of process drift before it impacts wafer yield for etch and deposition chambers.

An alternative approach is demonstrated by the Siemens Calibre Fab Insights platform, which takes use design content together with manufacturing history (yield maps, metrology measurements, process control etc.) to create predictive yield models and process monitoring models using ML (see figure 2) [15]. An important strength of this method is the ability to determine which characteristics of the chip design are most susceptible to the historical variation of a particular fab, so that the design team can make proactive decisions about layout choices, avoiding yield issues after first silicon.

C. System-Level and Cross-Lifecycle Digital Twins

Digital-twin concepts are being pushed beyond the realm of digital-twin electronic design into system-level electronics digital twins involving interactions among hardware, software, and environment throughout the product lifecycle, by Synopsys and other EDA vendors [18]. A system-level digital twin not only includes the logic and the physical implementation of the chip, but also how it interacts with the rest of the firmware, operating systems drivers, thermal management controllers and the rest of its system board environment. This degree of integration allows pre-silicon software bring-up, allowing the boot and test of an operating system on a virtual prototype of a chip, even before it has been fabricated in silicon, reducing the time to market for

more complex SoC products by months. These frameworks increasingly incorporate generative and predictive AI and are planned to be extended with agentic coordination [19].

IX. ECONOMIC DIMENSIONS OF AI-NATIVE DESIGN

A. Design-Cycle Time Compression

Previously months of intensive work by the physical design engineers were needed to get a chip floorplan completed [7]. While engineering review and integration may bring the actual six-hour figure down by a factor of ten or more, the fact that the number of months on tape-out has been reduced by an order of magnitude is a direct time-to-market impact. The ability to reduce the design cycle by 6 months or more, compared to a competitor, equates to billions of dollars in revenue in consumer and AI-accelerator chip segments. Cadence says Cerebrus has enabled customers to find feasible rapid exploration of large design-parameter spaces, cutting from weeks to hours the time typically needed for PPA optimisation.

B. The Cost of Late-Stage Design Errors

For advanced process nodes, the price of a chip re-spin after a design mistake can be in the millions of dollars and be the source of months of delay. Digital-twin based pre-fabrication validation and AI-assisted verification can most easily be viewed as an investment that avoids errors: It pays off in the extreme case of a large error, but not in the average case. This imbalance – modest recurring tooling and compute costs versus occasional, but large, losses – is a typical economic argument for investing in verification tooling in general – and is especially compelling for AI-native methods, because of the scale of modern SoC designs [15] and [16]. The Expected Value (EV) of marginal improvements to pre-silicon verification coverage is the probability of a costly respin times the cost of a respin.

C. Workforce Productivity as an Economic Multiplier

The average U.S. semiconductor industry revenue per worker is more than $744,000 as of 2024 [2] and by 2030, the SIA will have a shortfall of 67,000 workers [5]. In the short term, AI native tools that enable the engineering team to push the boundaries of the design space with the same number of engineers or to do physical design and verification sooner are a productivity multiplier not a headcount replacement. This is not only important for the way in which AI-native infrastructure will be budgeted in the business world, but also crucial for the way in which the labour-market effects of AI will be discussed by policy makers. Bigger productivity improvements boost output per engineer rather than cutting engineers and, with the current shortage of them, these tools are more likely to help design teams work on larger projects rather than lay off engineers.

TABLE III ECONOMIC VALUE DRIVERS FOR AI-NATIVE SEMICONDUCTOR TOOLING

Value Driver	Mechanism	Magnitude Indicator	Notes
Design-cycle compression	RL/ML-based physical design reduces iteration time	Hours vs. months for floorplan generation [7]	Commercially critical in high-velocity AI-chip segments
Respin avoidance	AI-assisted verification and digital-twin pre-silicon validation	Tens of M$ per respin at advanced nodes	Asymmetric payoff: modest tooling cost vs. large avoided loss
Engineer productivity	ML tools expand design-space coverage per engineer	Industry revenue/employee >$744K (2024) [2]	Productivity multiplier, not near-term headcount reduction
Yield prediction and ramp	Fab-level digital twins predict recipe outcomes without wafer runs	Recipe search space: 10^6–10^8 candidates [16]	Direct reduction in wafer cost for process development
Training cost offset	AI-native tools reduce need for scarce senior-expert time	Industry shortfall: 67,000 engineers by 2030 [5]	Structural economic argument for workforce-multiplying tooling

D. Caution on Reported ROI Figures

The efficiency data publicly reported (e.g. specific improvements for wafer virtual metrology or sustainability associated with digital-twin deployments) come from individual case studies and pilot programmes, not from an industry-wide average. Users of this paper as a guide to readers and policy makers should not interpret such numbers as definite returns that can be applied to any particular design team or any particular fabrication facility. The right level of generalisation is: Generalisation of tooling by AI has shown tangible benefits in specific, documented use cases, and the underlying processes (ML-based prediction, RL-based optimisation, digital-twin simulation) are sufficiently well understood that similar benefits are expected to be achievable in similar settings with facility engineering investments.

X. EVOLVING ENGINEER SKILL REQUIREMENTS AND CURRICULUM DESIGN

As design workflows become AI-native, the required skill profile of the semiconductor engineer must evolve substantially. Table IV summarises the transition from traditional to AI-native competency requirements.

TABLE IV EVOLUTION OF REQUIRED COMPETENCIES IN SEMICONDUCTOR ENGINEERING

Traditional Competency	AI-Native Competency
Digital/analog circuit design	Circuit design augmented with RL-based design-space exploration and learned optimisation of PPA trade-offs
Manual floorplanning and place-and-route	RL- and GNN-driven floorplanning, placement, and routing; evaluation of learned versus classical baseline results
Static timing analysis (STA)	GNN-based netlist representation and timing/power prediction; understanding of proxy-to-signoff gap
Rule-based verification and testbench design	AI-assisted verification, coverage-guided RL test generation, anomaly detection, and formal-ML hybrid approaches
Standalone EDA tool proficiency	Orchestration of multi-tool AI-native design flows; ML-driven design-space optimisation (DSO.ai, Cerebrus)
Post-silicon validation on physical prototypes	Digital-twin pre-fabrication testing, yield prediction, and equipment-level process optimisation
Single-domain expertise (logic or physical)	Hardware-software co-design: joint reasoning about silicon architecture and AI workloads running on it
Sensitivity analysis and Monte Carlo methods	ML-based corner and variation modelling; uncertainty quantification in process variation and reliability

The areas of ML fundamentals (supervised, unsupervised, and reinforcement learning algorithms and mathematics); reward design, policy optimisation, and sequential decision-making for physical design problems; GNN representation learning over netlists and circuit topologies for prediction and optimisation problems; and hardware-software co-design for joint reasoning over the AI workloads and silicon architecture it will eventually execute are emerging as central to the discipline.

A. A Four-Stage Curriculum Roadmap

Based on the evidence cited above, there is a four-stage "roadmap" to a curriculum for next-generation semiconductor engineers. This sequence is planned to develop skill in a progressive manner, building on skills learnt in the previous steps.

Stage 1—Foundations (Semesters 1-2): Digital circuit design, analog circuit design, computer architecture, linear algebra and probability, and an introduction to machine learning (supervised and unsupervised learning). Students will have to carry out at least one

laboratory-based circuit implementation project, gaining some familiarity with the design-rule, timing and power constraints which will be optimized by later AI-native methods.

Stage 2—Core AI Methods (Semesters 3-4): A separate course on RL that includes Markov Decision Processes, policy gradient methods (REINFORCE, PPO, A3C), and actor-critic methods. Together with a graph representation learning course on message passing neural networks, graph convolution networks, graph attention networks and application to combinatorial optimisation. Students need to develop a toy placement agent to serve a small benchmark circuit to correlate algorithmic ideas with physical design considerations.

Stage 3—Applied Systems (Semesters 5-6): Practical application of open source EDA and RL-for-EDA tools. The Concrete and Reproducible Entry Point is provided by reproducing results within the framework of Circuit Training / AlphaChip [12] and the MacroPlacement TILOS-AI-Institute benchmark suite [20] without the need of proprietary EDA access. Combined with a VLSI physical design course for students to link algorithmic ideas with real physical, timing and power limitations. Digital-twin concepts need to be introduced via a lab, with open-source simulation frameworks.

Stage 4—Capstone Integration (Semester 7–8): A capstone project for the application of RL or GNN methods to an open benchmark circuit, a critical evaluation of which is required and involves students comparing their approach to the use of a well-tuned simulated annealing implementation and measuring proxy and true post-route measures. This directly involves students in the methodological debate that is recorded in Section VI and develops rigorous benchmarking discipline. The students at the end of the sequence should be able to critically assess the claims of AI-native EDA, to design and implement prediction models with GNNs for selected EDA tasks, and to apply RL for the optimisation to an open-source physical design workflows.

XI. EVIDENCE STRENGTH, LIMITATIONS, AND OPEN RESEARCH CHALLENGES

A. The Placement Debate Is Peer-Reviewed on Both Sides

The 2021 Nature floorplanning by RL is really important and peer-reviewed [7]. First released as a preprint in 2023, the methodology re-evaluation by Cheng, Kahng et al., has been accepted for publication in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems [20] and authors report that the re-evaluation of the methodology confirms their original finding – a well-tuned SA baseline outperforms pre-trained Circuit Training on both proxy and true post-route metrics. The original claim now has the status of peer-review and the critique has also been given the status of peer-review. It should be noted that claims of RL 'outperforming' classical methods should be interpreted with regard to the protocol and baseline examined and not as general.

B. Manufacturing and Efficiency Gains Are Context-Dependent

Digital-twin numbers attributed by equipment vendors are from specific pilot programmes as opposed to industry averages. They are a representation of direction and possible size, they are not a sure thing for any particular manufacturing plant or design group. This disparity is even more pronounced when looking at the number of peer-reviewed, independently replicated manufacturing-outcome studies with AI-native digital twins, compared to vendor reports—from which readers should give far less credence to specific quantitative claims.

C. Agentic Workflows Remain Largely Prospective

There are no fully production-scale agentic semiconductors workflows that are fully autonomous as described in the peer-reviewed literature discovered for this paper. This is a planned future extension, and is not a validated capability, in published frameworks in this space [19]. The use of agentic workflows in this paper should be viewed within the context of the development of its constituent pieces of technology: RL-based physical design, GNN-based EDA prediction and digital-twin

simulation; all are individually demonstrated and the next logical step in integrating all these technologies for agentic orchestration is yet to be shown independently at the level of rigor required for peer-reviewed production claims.

D. Four Open Research Challenges

The following challenges represent tractable directions for future research; each grounded in gaps identified in the evidence reviewed above.

1) Benchmark Standardisation: It is important to have benchmark standardisation with open access, to be able to compare RL vs Classical on a level playing field across research groups and to be able to verify claims. The TILOS-AI-Institute Macro Placement benchmark suite can be a step towards this direction, but is currently limited to a few circuit families and process nodes.

2) Proxy-to-Signoff Gap Closure: Proxy-to-signoff gap closure has been documented in [9] and proxy-to-signoff gap closure is known to be unreliable for gNN models trained on pre-route proxy metrics. An open and commercially relevant research area is the use of methods that explicitly model the gap, for instance, by learning a model of the gap between proxy and signoff, or by training directly on post-route objectives with differentiable routing approximations.

3) Cross-Domain Generalisation: The vast majority of published RL and GNN results are shown on one or a few specific chip families, benchmarks and/or process nodes. So far, there is a lack of proper generalisation over foundries and design styles (and quantified degradation budgets in case the distribution changes) which is necessary for generalisation to be adopted by any industry.

4) Validated Agentic Orchestration: There is currently no peer-reviewed literature found in the surveyed literature that includes examples of multi-agent AI systems that make autonomous decisions in design, verification and manufacturing. This is the largest disconnect between the industry desired and the industry actual in the field, and the direction it's heading fastest.

XII. IMPLICATIONS FOR INDUSTRY, ACADEMIA, AND POLICY

A. For Industry

For Industry In addition to the core EDA technologies and the reorganization of design teams, semiconductor companies will require investment in AI infrastructure and staffing with AI experts to achieve the best results. Beyond the EDA technologies, the semiconductor companies will need to invest in AI infrastructure and hire AI experts to work collaboratively with circuit designers, rather than in isolation. The economic argument is strongest for applications where AI is a tool that eliminates errors at a later stage of design (Section IX-B), or where the application leverages a tight engineering team (Section IX-C). In Section VI, firms should use the methodological controversy as an indicator of needing to go into their own internal benchmarking efforts rather than accepting gains reported by vendors without independent validation.

B. For Academia

RL and GNN methods will have to be made part of the compulsory courses of the electrical and computer engineering curricula, not as elective specialisation. One possible sequence for the four stages is given in the roadmap in Section X-A. Students and researchers who do not have access to proprietary EDA infrastructure can use the open-source artefacts, like Google's Circuit Training / AlphaChip framework [12] or the TILOS-AI-Institute MacroPlacement benchmark suite [20] to enter into the process. The gap identified by Cheng/Kahng needs to be addressed by submission of evaluation-methodology papers to academic journals and programme committees of conferences that stress test the claims in the field of AI-native EDA against well-engineered classical baselines.

C. For Policy and Research Funding

SIA's 2025 State of the Industry report recognizes that, compared to other key competitors, the U.S. R&D tax incentive rate is lower than China (25.8 percent) and Japan (14.8 percent), and calls for the

incentive credits to be broadened to cover chip design and research, in addition to manufacturing capacity [2]. The projected shortfall of 67,000 workers is disproportionately design-side engineering, which may make it possible to fill the workforce gap more efficiently than relying on policy to boost the number of workers in fabrication capacity. Research funding organizations must make strong statements in regard to AI and chip design as strategic areas for funding, along with funding for benchmark infrastructure and open-source EDA tooling for proper comparative studies.

D. For Students and Early-Career Researchers

From the point of view of students new to the field, the take-home message of this survey is straightforward: training in an integrated approach to circuit fundamentals, RL based on graphs, and simulation-based validation will be more useful than training in any of these areas individually. As the field evolves, what will differentiate trustworthy results from untrustworthy ones is the ability to rigorously benchmark AI-native methods and compare them against well engineered classical methods, in a way that relies on proper methods and metrics for evaluating both alternatives. In fact, as the field evolves, one of the most crucial meta skills to be developed is rigorous benchmarking, which is the capacity to evaluate AI-native methods against well engineered classical alternatives, using the right methods and metrics for evaluating both.

TABLE V SUMMARY OF KEY OPEN RESEARCH CHALLENGES AND TRACTABLE DIRECTIONS

Challenge	Description	Tractable Direction
Benchmark standardisation	Lack of common, open benchmarks prevents apples-to-apples comparison of AI-native vs. classical methods across groups [20].	Expand TILOS MacroPlacement suite to more circuit families, nodes, and foundries; standardise baseline implementations.
Proxy-to-signoff gap	GNNs accurate on pre-route proxies may rank designs incorrectly by final post-route signoff metrics [9].	Train correction models from proxy-to-signoff; explore differentiable routing approximations as training objectives.
Cross-domain generalisation	Most results are demonstrated on specific families/nodes; generalisation across PDKs and design styles unresolved.	Systematic transfer-learning studies across process nodes; PDK-agnostic GNN architectures.
Agentic orchestration	Multi-agent AI systems coordinating design, verification, and manufacturing decisions not yet validated in peer-reviewed literature [19].	Pilot agentic frameworks on open benchmark flows; formal evaluation protocols for agent coordination.
Human-in-the-loop boundaries	Unclear which design decisions AI agents should own versus escalate to human engineers for judgment.	Develop uncertainty-quantification methods that trigger human review when agent confidence falls below a threshold.

XIII. CONCLUSION

The semiconductor industry is at a turning point – physical scaling constraints, economic considerations and the growing skills gap are all pushing toward a major shift toward a future of AI-native designs. The research presented in this survey has confirmed and substantiated with peer-reviewed reports and verified industrial reports that the usage of RL for floor planning, usage of GNN for EDA prediction, and pre-fabrication validation using digital-twins is not speculation—this is the areas of research published in peer-reviewed journals and in selected cases already deployed in industry. The need for AI-native design productivity tools is less about a cycle and more about a structural need as the global semiconductor market is worth $791.7 billion in 2025 [3] and AI application accounts for 34.9 percent of global chip demand [2] and by 2030, there is an expected shortage of 67,000 engineers in the semiconductor industry. [5]

The comparative analysis in Section VI gives a calibrated and defensible result: ML-based placement methods are valid and competitive methods for the physical designer's toolbox, but they do not yet

clearly outperform the best available classical placement methods in terms of advantages over these latter methods after comparable engineering efforts. The current methodological debate, which has in its turn been peer-reviewed on both sides [7, 20] should not be interpreted as a justification to reject methods which are intimately connected to AI, but rather as a sign that the world of methods has reached the level of maturity where it is possible and even necessary to conduct hard-to-beat benchmarking.

Technical architecture for GNN based EDA prediction (Section VII), multi-level digital-twin ecosystem (Section VIII) and the nascent prospect of agentic workflow orchestration (Section III-D, XI-C) together present a decade-long evolution of growing AI usage in chip design. The four open research challenges listed in Section XI (benchmark standardization, proxy-to-signoff gap closure, cross-domain generalization and validated agentic orchestration) are the future of this path. One of the most important technical and educational needs of the semiconductor community over the next decade is the ability to close these gaps, and build up the engineering workforce with the right skills to do so, through the curriculum roadmap proposed in Section X.

APPENDIX: GLOSSARY OF KEY TERMS

Reinforcement Learning (RL): A machine learning framework with an agent that learns sequential decision-making based on reward feedback; the MDP is formulated as a sequence of macro-blocks placement and used for chip placement, where each placement of a macro-block is an action.

Graph Neural Network (GNN): A neural network architecture where nodes and/or edges in the network communicate with one another via messages, a graph-based structure such as circuits, netlists and RTL descriptions.

Floor planning: The physical design phase where functional macro blocks are placed on the chip die in order to maximize area, timing, congestion and power before detailed placement and routing is completed.

Digital Twin: Virtual, data-driven copy of any physical system (chip, process equipment, fab), to simulate, predict and optimise its behaviour before or during physical operation.

Electronic Design Automation (EDA Programs and tools for designing, simulating, verifying and manufacturing integrated circuits including synthesis, place-and-route, timing analysis and physical verification.

Signoff: The final pass through a process that determines whether a chip design will satisfy all timing, power and manufacturability parameters before it is sent to the fabrication process.

Hardware-Software Co-Design: An engineering strategy that optimises the design of the chip and the software or AI workloads to execute on it, to encourage each to inform each other's design constraints.

Half-Perimeter Wirelength (HPWL): A commonly-used proxy for interconnect quality in placement evaluation, calculated as the sum of all bounding-box half-perimeters across all nets; used when full routing is not feasible during optimisation, for example.

Power-Performance-Area (PPA): AI-native approaches are benchmarked against PPA trade-offs, as the three main goals of semiconductor physical design are minimising power consumption, maximising operating frequency, and minimising the size of the die.

Gate-All-Around (GAA): As a result, a new transistor architecture, known as surrounding gate geometry, was developed to provide improved electrostatic control over that obtained with FinFETs at sub-3 nm process nodes.

Simulated Annealing (SA): A probabilistic metaheuristic optimization algorithm that searches a solution space with accepting inferior solutions based on their probability, which decreases over time; classical algorithm used as a baseline against which the RL-based placement is compared.

Process Design Kit (PDK): A collection of files that define the design rules, device models and layout parameters of a specific foundry process node; generalisation of GNN across different PDKs is still an open challenge.

REFERENCES

[1] International Technology Roadmap for Semiconductors (ITRS), "ITRS 2.0: 2015 Edition," ITRS, 2015. [Online]. Available: http://www.itrs2.net/

[2] Semiconductor Industry Association (SIA), "2025 State of the U.S. Semiconductor Industry," SIA, Washington, DC, Jul. 2025. [Online]. Available: https://www.semiconductors.org/2025-state-of-the-industry-report-investment-and-innovation-amidst-global-challenges-and-opportunities/

[3] Semiconductor Industry Association (SIA), "Global Annual Semiconductor Sales Increase 25.6% to $791.7 Billion in 2025," SIA, Washington, DC, Feb. 6, 2026. [Online]. Available: https://www.semiconductors.org/global-annual-semiconductor-sales-increase-25-6-to-791-7-billion-in-2025/

[4] Deloitte, "2026 Global Semiconductor Industry Outlook," Deloitte Insights, Feb. 2026. [Online]. Available: https://www.deloitte.com/us/en/insights/industry/technology/technology-media-telecom-outlooks/semiconductor-industry-outlook.html

[5] Semiconductor Industry Association (SIA) and Oxford Economics, "Chipping Away: Assessing and Addressing the Labor Market Gap Facing the U.S. Semiconductor Industry," SIA, Washington, DC, Jul. 2023. [Online]. Available: https://www.semiconductors.org/chipping-away-assessing-and-addressing-the-labor-market-gap-facing-the-u-s-semiconductor-industry/

[6] M. M. Waldrop, "The chips are down for Moore's law," Nature, vol. 530, no. 7589, pp. 144–147, Feb. 2016. doi: 10.1038/530144a

[7] 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. doi: 10.1038/s41586-021-03544-w

[8] D. S. Lopera, L. Servadei, G. N. Kiprit, S. Hazra, R. Wille, and W. Ecker, "A survey of graph neural networks for electronic design automation," in Proc. 3rd ACM/IEEE Workshop on Machine Learning for CAD (MLCAD), 2021, pp. 1–6. doi: 10.1109/MLCAD53597.2021.9531070

[9] R. Chen et al., "Graph computation meets circuit algebra: A task-aligned analysis of graph neural networks for electronic design automation," arXiv:2605.08291, May 2025. [Online]. Available: https://arxiv.org/html/2605.08291v1

[10] Synopsys Inc., "DSO.ai: Autonomous AI-driven design space optimization," Synopsys. [Online]. Available: https://www.synopsys.com/implementation-and-signoff/ml-ai-driven-design/dso-ai.html

[11] Cadence Design Systems Inc., "Cerebrus Intelligent Chip Explorer," Cadence. [Online]. Available: https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/silicon-signoff/cerebrus-intelligent-chip-explorer.html

[12] Google Research, "Circuit Training: An Open-Source Framework for Generating Chip Floor Plans with Distributed Deep Reinforcement Learning," GitHub, 2021. [Online]. Available: https://github.com/google-research/circuit_training

[13] 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. doi: 10.1109/TCAD.2020.3003843

[14] R. Cheng and J. Yan, "On joint learning for solving placement and routing in chip design," in Proc. 35th Conf. Neural Inf. Process. Syst. (NeurIPS), 2021. [Online]. Available: https://arxiv.org/abs/2111.00234

[15] Siemens Digital Industries Software, "Unlocking the Future with a Digital Twin for Semiconductor Manufacturing," Siemens Calibre Blog, Mar. 29, 2024. [Online]. Available: https://blogs.sw.siemens.com/calibre/2024/03/29/unlocking-the-future-with-a-digital-twin-for-semiconductor-manufacturing/

[16] Tokyo Electron Ltd. (TEL), "Digital Twins Accelerate Innovation in Semiconductor Technology," TEL Blog, Aug. 28, 2025. [Online]. Available: https://www.tel.com/blog/all/20250828_001.html

[17] Lam Research Corp., "Revolutionizing Chipmaking: The Power of Digital Twins in Semiconductor Manufacturing," Lam Research Newsroom. [Online]. Available: https://newsroom.lamresearch.com/how-digital-twins-can-revolutionize-chipmaking

[18] Synopsys Inc., "Digital Twins for the Semiconductor Industry," Synopsys Blog. [Online]. Available: https://www.synopsys.com/blogs/chip-design/digital-twins-semiconductor-industry.html

[19] R. Laubscher et al., "Generative and predictive AI for digital twin systems in manufacturing," PMC / NCBI, 2025. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC12753877/

[20] C.-K. Cheng, A. B. Kahng, S. Kundu, Y. Wang, and Z. Wang, "An updated assessment of reinforcement learning for macro placement," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., early access, Dec. 2025. doi: 10.1109/TCAD.2025.3545001. [Online]. Available: https://ieeexplore.ieee.org/document/11300304; code/benchmarks: https://github.com/TILOS-AI-Institute/MacroPlacement

[21] Anonymous authors, "Physics-guided geometric diffusion for macro placement generation (MacroDiff+)," arXiv:2605.16451, 2025/26. [Online]. Available: https://arxiv.org/pdf/2605.16451

[22] C.-K. Cheng, A. B. Kahng, I. Markov, and J. Hu, VLSI Physical Design: From Graph Partitioning to Timing Closure, 2nd ed. Springer, 2022. doi: 10.1007/978-3-030-96415-3

[23] Semiconductor Industry Association (SIA), "2025 SIA Factbook," SIA, Washington, DC, May 2025. [Online]. Available: https://www.semiconductors.org/wp-content/uploads/2025/05/2025-SIA-Factbook-FINAL-1.pdf

[24] G. Huang et al., "Machine learning for electronic design automation: A survey," ACM Trans. Design Autom. Electron. Syst., vol. 26, no. 5, pp. 1–46, 2021. doi: 10.1145/3451179

[25] J. Gu et al., "A comprehensive survey on electronic design automation and graph neural networks," ACM Trans. Design Autom. Electron. Syst., vol. 28, no. 2, 2023. doi: 10.1145/3543853

[26] Semiconductor Industry Association (SIA), "Semiconductor Workforce Policy Blueprint," SIA, Washington, DC, Apr. 2024. [Online]. Available: https://www.semiconductors.org/wp-content/uploads/2024/04/SIA-Workforce-Policy-Blueprint-4_8_24.pdf

[27] CHIPS and Science Act of 2022, Pub. L. No. 117-167, 136 Stat. 1366, Aug. 9, 2022. [Online]. Available: https://www.congress.gov/bill/117th-congress/house-bill/4346