IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

Machine Learning-Assisted Optimization of CMOS VLSI Circuits for Low-Power and High-Speed Applications

Ankit Soni
Assistant Professor,
Gyan Sagar College of Engineering, Sagar, (MP)

Abstract: *Scaling of complementary metal-oxide-semiconductor (CMOS) technology into the deep-nanometre regime has made the simultaneous minimisation of power and propagation delay one of the hardest problems in very-large-scale integration (VLSI) design. Supply voltage, threshold-voltage assignment, transistor sizing, body biasing and buffer insertion interact non-linearly, and the design space grows combinatorially with circuit size, so exhaustive SPICE-based exploration is intractable beyond small blocks. Classical metaheuristics such as simulated annealing and genetic algorithms handle the non-convexity but remain simulation-bound: every candidate must be evaluated by a costly simulator. This paper presents a machine learning-assisted optimisation framework that replaces most simulator calls with a learned surrogate, decoupling search cost from simulation cost. A graph neural network that consumes the netlist as an attributed graph is fused with a gradient-boosted regressor over scalar descriptors through a deep ensemble, which also yields a calibrated predictive uncertainty. A constrained NSGA-II search runs over the surrogate; an active-learning loop returns only high-uncertainty candidates to the simulator, and a verification stage re-simulates the Pareto set under process corners and Monte-Carlo variation. On eight digital benchmarks spanning arithmetic, control and datapath structures, the framework attains a mean power reduction of 27.4 % and a mean critical-path delay reduction of 18.6 % against a reference sizing — a 40.8 % improvement in power-delay product — while requiring roughly twenty times less wall-clock time than NSGA-II applied directly to SPICE. Surrogate accuracy reaches 2.7 % mean absolute percentage error for power and 2.4 % for delay. Ablation studies isolate the contribution of the graph encoder, the uncertainty-guided sampling policy and the ensemble, and a SHAP attribution analysis shows that supply voltage, critical-path sizing and threshold-voltage assignment dominate the learned power model, consistent with established device physics.*

Keywords: CMOS VLSI, low-power design, high-speed circuits, machine learning, graph neural networks, design space exploration, surrogate modelling, multi-objective optimisation, power-delay product, electronic design automation.

I. INTRODUCTION

1.1 Motivation

For nearly three decades the economics of integrated-circuit design rested on the scaling rules formalised by Dennard and co-workers, under which proportional reduction of device dimensions and supply voltage kept power density approximately constant while transistor count and switching speed improved [1]. That regime ended in the mid-2000s. Once the supply voltage could no longer be reduced in step with the feature size — because the threshold voltage could not be lowered without an exponential penalty in subthreshold leakage — power density began to rise at each successive node [2]. Modern systems-on-chip are therefore power-limited rather than area-limited, and the fraction of a

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

die that can be switched simultaneously at full frequency within a fixed thermal envelope has been shrinking node over node [3], [4].

Power dissipation in a static CMOS gate has three components. The dynamic component arises from charging and discharging load capacitance and scales with the square of the supply voltage [5]. The short-circuit component arises during the finite input transition window in which both networks conduct. The static component is dominated by subthreshold conduction, gate-oxide tunnelling, gate-induced drain leakage and junction leakage, and grows exponentially as the threshold voltage falls [6]. Because delay depends inversely on overdrive voltage, reducing dynamic power by lowering the supply degrades speed, and recovering that speed by lowering the threshold voltage inflates leakage. The designer therefore faces a coupled trade-off surface rather than a single objective [7].

What makes this difficult in practice is dimensionality rather than qualitative structure. A realistic problem for a moderately sized datapath block exposes several hundred simultaneously adjustable quantities — per-gate transistor widths, a discrete threshold-voltage flavour per cell, one or more supply domains, body-bias levels per well, buffer insertion points and the target clock period — which interact through the delay and power models within a region further constrained by setup and hold requirements, slew limits and area budgets. Evaluating one point to sign-off accuracy requires a transistor-level simulation costing seconds to minutes; evaluating the space exhaustively is impossible.

1.2 Limitations of Conventional Optimisation Strategies

Three families of approach have historically been used. Analytical methods based on logical effort or convex posynomial approximations are fast and provably optimal within their model, but their accuracy degrades at nanometre nodes where velocity saturation, short-channel effects and layout-dependent stress make simple closed forms unreliable. Heuristic search over the true simulator — simulated annealing [9] or evolutionary algorithms such as NSGA-II [10] — makes no modelling assumptions and handles non-convex, mixed discrete-continuous spaces gracefully, but is sample-inefficient: a search requiring twenty thousand evaluations at four seconds each consumes more than twenty hours per block. The internal heuristics of commercial tools are highly tuned but opaque, hard to retarget, and their quality varies substantially with tool-option settings [8], [11].

The observation motivating this work is that heuristic search is not intrinsically expensive — the evaluation oracle is. If a sufficiently accurate and cheap approximation of that oracle can be learned from a modest number of true simulations, the search becomes nearly free and the simulation budget can be spent selectively where the approximation is least trustworthy. This is the role machine learning has come to play in design automation over the past decade [11]–[13].

1.3 Machine Learning in Electronic Design Automation

Recent results are qualitatively different from earlier attempts. Reinforcement learning has been applied to macro placement with quality comparable to or better than human floorplans on production designs [14]. Analytical placement has been reformulated as a differentiable optimisation problem executed on deep-learning toolkits, yielding order-of-magnitude speedups through GPU acceleration [15]. Graph convolutional agents transfer transistor-sizing knowledge across topologies and technology nodes [16]. Convolutional models predict dynamic IR drop roughly thirty times faster than commercial analysis tools and transfer between designs [17]. Public datasets such as CircuitNet have begun to relieve the data-scarcity problem [18], and comprehensive surveys now organise these contributions along the design hierarchy [12], [13].

Within this landscape, joint power-and-delay optimisation at the circuit and gate level has received less systematic treatment than placement and routing. Most existing work either predicts a single metric without closing the optimisation loop, optimises a single objective, or targets analog sizing where the design space is far smaller. This paper addresses that gap.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

1.4 Contributions

A unified formulation that treats supply voltage, threshold-voltage flavour assignment, transistor sizing, body bias, buffer insertion and target frequency as a single mixed discrete-continuous design vector, with power, delay and area as constrained objectives over it.

A hybrid surrogate combining a graph neural network over the netlist with a gradient-boosted regressor over scalar descriptors, fused through a deep ensemble that supplies a calibrated predictive uncertainty alongside the point estimate.

An uncertainty-guided active-learning loop that allocates the scarce simulation budget where the surrogate is least confident rather than sampling uniformly.

Embedding of the surrogate in a constrained NSGA-II search with a full-simulation verification stage covering process corners and Monte-Carlo variation, so that no reported design point rests on a model prediction alone.

An empirical evaluation on eight benchmarks quantifying power, delay, power-delay product and runtime against four baselines, with ablation studies, cross-node transfer, non-parametric significance testing and SHAP-based interpretability analysis.

1.5 Organisation of the Paper

Section 2 reviews the physical background and machine learning preliminaries. Section 3 surveys related work and identifies the research gap. Section 4 develops the proposed framework. Section 5 describes the experimental setup, benchmarks and baselines. Section 6 reports and discusses results. Section 7 examines remaining limitations, Section 8 outlines future directions, and Section 9 concludes.

II. BACKGROUND AND PROBLEM FORMULATION

2.1 Power Dissipation in CMOS Circuits

The total power dissipated by a synchronous CMOS circuit is conventionally decomposed as

$$P_{total} = P_{switching} + P_{short-circuit} + P_{static} \quad (1)$$

where the switching component for a node driving load capacitance C_L at clock frequency f_{clk} with activity factor α is

$$P_{switching} = \alpha \cdot C_L \cdot V_{DD}^2 \cdot f_{clk} \quad (2)$$

This expression, established in the foundational treatment of low-power CMOS design [5], explains why supply-voltage reduction has always been the most effective single lever available: the dependence on V_DD is quadratic, whereas every other term is at best linear. The short-circuit component arises because, during an input transition of duration τ, both networks conduct while the input lies between $V_{TH,n}$ and $V_{DD} - |V_{TH,p}|$:

$$P_{short-circuit} \approx \left(\frac{\beta}{12}\right) \cdot \tau \cdot f_{clk} \cdot (V_{DD} - 2V_{TH})^3 \quad (3)$$

This term is normally modest in a circuit with controlled slew rates, but becomes significant when long nets are left unbuffered. The static component is the one whose relative importance has grown most with scaling; its dominant contributor at moderate temperatures is subthreshold conduction,

$$I_{sub} = \mu \cdot C_{ox} \cdot \left(\frac{W}{L}\right) \cdot v_T^2 \cdot e^{\left\{\frac{V_{GS}-V_{TH}}{n \cdot v_T}\right\}} \cdot \left(1 - e^{\left\{-\frac{V_{DS}}{v_T}\right\}}\right) \quad (4)$$

in which v_T is the thermal voltage and n the subthreshold slope factor. The exponential dependence on V_{TH} means a 100 mV reduction of the threshold increases subthreshold leakage by roughly an order of magnitude at room temperature. Gate-oxide tunnelling, gate-induced drain leakage and junction leakage contribute further terms whose weights depend on the gate stack and doping profile; a systematic account of all of these and of the circuit techniques used to mitigate them is given by Roy and co-workers [6].

2.2 Delay and the Power-Delay Trade-off

The propagation delay of a gate driving a capacitive load is customarily approximated by the alpha-power law,

$$t_{pd} \approx \frac{(K \cdot C_L \cdot V_{DD})}{(W/L \cdot (V\_DD - V\_TH)\hat{}\alpha)} \quad (5)$$

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

where α is a velocity-saturation index between roughly 1.2 and 1.6 in short-channel devices. Equations (2) and (5) together define the trade-off addressed here. Lowering V_{DD} reduces dynamic power quadratically but increases delay, and the increase becomes steep as V_{DD} approaches V_{TH} because the overdrive term collapses. The composite figures of merit are the power-delay product $PDP = P \bullet t_{pd}$, which measures energy per operation, and the energy-delay product $EDP = P \bullet t_{pd}^2$, which weights speed more heavily and is the more appropriate metric for high-performance applications.

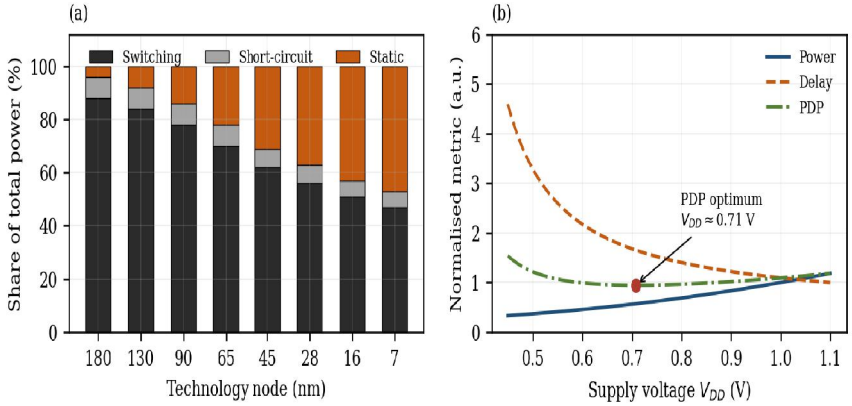

Fig. 1. (a) Representative evolution of the relative contributions of switching, short-circuit and static power with technology node; the static share grows steadily as the threshold voltage is scaled down. (b) Normalised power, delay and power-delay product against supply voltage for a representative logic path; the PDP exhibits a well-defined interior minimum whose location shifts with threshold voltage, activity factor and load.

Figure 1(a) shows that at 180 nm the switching component dominates, whereas by 7 nm the static component is comparable to it in many designs. The methodological consequence is direct: an optimiser minimising dynamic power alone will systematically produce designs inferior in total power at advanced nodes, because it drives the threshold voltage down and the leakage up without accounting for the penalty. Figure 1(b) shows that the PDP-optimal supply voltage is an interior optimum whose location depends on parameters varying from block to block, which is why a single global supply chosen by rule of thumb leaves substantial energy efficiency unexploited and why V_DD must be treated as an optimisation variable rather than a constant.

2.3 The Design Space and Its Complexity

Let the design vector be

$$x = [V_{DD}, v_{th}, w, V_{BB}, b, f_{clk}] \in X \quad (6)$$

where $v_{th} \in \{LVT, SVT, HVT\}^N$ is a discrete per-cell threshold assignment over N cells, $w \in R^M$ a vector of width ratios for the M sizable devices, V_BB a body-bias level per well, $b \in \{0,1\}^K$ buffer-insertion decisions at K candidate sites, and f_clk the target frequency. The optimisation problem is

$$minimise\ [P(x), T(x)]\ \ subject\ to\ A(x) \le A\_max,\ slack(x) \ge 0,\ x \in X \quad (7)$$

The cardinality of the discrete part alone is 3^N · 2^K, which for two hundred cells and fifty buffer sites exceeds 10^110; the continuous part adds a real subspace of dimension M + 3. No exhaustive method can address a space of this size, and because P and T are available only through simulation, gradients are unavailable in closed form. Combinatorial size together with expensive black-box evaluation are exactly the conditions under which surrogate-assisted optimisation is advantageous.

A further complication is that the objectives are not deterministic. Random dopant fluctuation, line-edge roughness and oxide thickness variation induce mismatch whose standard deviation scales inversely with the square root of device area, as described by the Pelgrom model [19], and systematic across-wafer variation shifts the operating point of an entire die. A design optimal at the nominal corner may violate timing at the slow-slow corner or exceed the power

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

budget at the fast-fast corner. Any framework intended for practical use must therefore treat corner and statistical verification as an integral stage rather than an afterthought.

2.4 Machine Learning Preliminaries

2.4.1 Supervised regression models

he surrogate task is supervised regression: given $D = \{(x_i, y_i)\}$ in which y_i collects simulated power, delay and area, learn $\hat{f}$ approximating the simulator. Random forests aggregate decorrelated trees and are robust to irrelevant features and heterogeneous scales [21]; gradient-boosted trees, in the regularised implementation popularised as XGBoost, fit an additive sequence of trees to the residuals of previous stages and generally outperform bagged ensembles on tabular data of moderate size [22]. Support-vector regression performs well in low-data regimes at the cost of unfavourable scaling in sample count [23], and feed-forward networks approximate arbitrary continuous functions given sufficient capacity and data [20].

2.4.2 Graph neural networks

A netlist is naturally a graph: cells are vertices and nets are edges. Tabular models discard this structure, encoding it at best through hand-crafted statistics such as average fan-out. Graph neural networks operate on the structure directly, iteratively updating each vertex representation from its neighbours [24]. In the graph convolutional formulation the layer-wise update is

$$H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \bullet \tilde{A} \bullet \tilde{D}^{-\frac{1}{2}} \bullet H^{l} \bullet W^{l}\right) \quad (8)$$

where Ã is the adjacency matrix with self-loops, D̃ its degree matrix, H^(l) the vertex feature matrix at layer l and W^(l) a learned weight matrix [25]. After L propagation steps each vertex representation summarises an L-hop neighbourhood, and permutation-invariant pooling produces a graph-level embedding suitable for regression. This inductive bias matters here because power and delay are structural quantities: block delay depends on the topology of the critical path, and switching power on how activity propagates through the connectivity.

2.4.3 Surrogate-assisted and Bayesian optimisation

Bayesian optimisation formalises the use of a probabilistic surrogate to guide evaluation of an expensive black-box function [26]. A prior over functions, classically a Gaussian process, is updated with observations, and the next query maximises an acquisition function balancing exploitation against exploration; the expected-improvement criterion from the efficient global optimisation literature remains among the most widely used [27]. The framework proposed here adopts the same principle but replaces the Gaussian process — whose cubic scaling in observation count and whose dependence on a fixed-length feature vector are both problematic in this setting — with a deep ensemble whose disagreement serves as the uncertainty estimate.

III. RELATED WORK

3.1 Predictive Surrogates for Power, Performance and Area

The most widely deployed use of learning in design automation is cross-stage prediction: estimating a downstream quantity from upstream data to avoid a costly forward pass. Routability prediction is the canonical example. Convolutional models trained on placement images forecast global routing congestion and locate design-rule-check hotspots with accuracy comparable to a global router at a fraction of the runtime [28], and graph-based formulations improve on image-based ones by respecting netlist structure explicitly [29]. Related work applies deep biased learning to lithographic hotspot detection from layout clips [30]. On the power side, PowerNet estimates dynamic IR drop across a full chip about thirty times faster than a commercial tool and, importantly, transfers between designs rather than requiring per-design retraining [17]. These contributions establish that learned models can substitute for expensive analyses with acceptable fidelity, but they stop short of closing the loop: the prediction informs a human or a conventional heuristic rather than driving an automated search.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

3.2 Black-box Optimisation of Circuit Parameters

A second line of work uses learning to drive optimisation directly. In the analog domain, AutoCkt formulates transistor sizing as a sequential decision problem and trains a deep reinforcement-learning agent with sparse subsampling, reporting convergence on most target specifications at a substantial speedup over genetic search [31], while GCN-RL Circuit Designer augments such an agent with a graph convolutional encoder of the topology, enabling knowledge transfer across technology nodes and topologies — a capability flat parameter-vector agents lack [16]. Bayesian approaches have also been applied extensively to analog sizing, where the space is small enough for Gaussian-process surrogates to remain tractable. The digital gate-level problem differs in an important respect: the number of decision variables is one to three orders of magnitude larger and many are discrete, so methods that work well for a ten-parameter amplifier do not transfer directly.

3.3 Learning in Physical Design

The most visible recent results concern physical design. A deep reinforcement-learning agent with an edge-based graph convolutional architecture produces macro floorplans in under six hours that are comparable to or better than human layouts on power, performance and area for production accelerator designs [14], and DREAMPlace recast analytical global placement as neural-network training, applying standard toolkits and GPU hardware to a classical numerical problem with speedups exceeding an order of magnitude and no loss of quality [15]. Open-source flows such as OpenROAD have made the labelled data these methods require far easier to generate [32]. These results are important existence proofs, but they operate at the placement and floorplanning level, where the objective is largely wirelength and congestion; the coupled power-delay optimisation addressed here operates earlier in the flow and on different variables.

3.4 Datasets, Surveys and Reproducibility

Progress was long limited by the absence of public data: most published models were trained on small proprietary datasets, making benchmarking and independent reproduction impossible. CircuitNet addressed this by releasing more than ten thousand samples extracted from commercial tool runs on open-source RISC-V designs, with domain-specific evaluation metrics for routability, DRC and IR-drop tasks [18]. Surveys now organise the field along the design hierarchy [12] and along the design-time versus run-time axis [13], and an edited volume collects methodological treatments of the principal application areas [33]. Kahng's earlier position paper anticipated much of this development, framing it as the removal of unnecessary design margins and the acceleration of design convergence [11].

3.5 Research Gap

Table I summarises the positioning of representative prior work. Three gaps are apparent. First, the majority of published models are predictors rather than optimisers; where optimisation is closed-loop it is usually single-objective. Second, work that does address multi-objective power-delay optimisation operates either on very small analog design spaces or on placement-level objectives, not on the mixed discrete-continuous gate-level space of (6). Third, few frameworks quantify predictive uncertainty, so the simulation budget is allocated uniformly rather than where the model is weakest, and model errors propagate silently into the reported optimum. The framework of Section 4 is designed to close these three gaps.

TABLE I. POSITIONING OF REPRESENTATIVE PRIOR WORK

Reference	Learning model	Design stage	Objective(s)	Closed loop	Uncertainty
Kahng [11]	Various (position paper)	Physical design	Margin / convergence	Partly	No
Xie et al. [28]	CNN	Placement	Routability	No	No
Kirby et al. [29]	Graph NN	Placement	Congestion	No	No
Yang et al. [30]	CNN	Mask synthesis	Hotspot detection	No	No
Xie et al. [17]	CNN	Power analysis	IR drop	Partly	No

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

Mirhoseini et al. [14]	GCN + RL	Floorplanning	Wirelength, congestion	Yes	No
Lin et al. [15]	Differentiable opt.	Placement	Wirelength, density	Yes	No
Settaluri et al. [31]	Deep RL	Analog sizing	Spec satisfaction	Yes	No
Wang et al. [16]	GCN + RL	Analog sizing	Figure of merit	Yes	No
**This work	**GNN + GBT ensemble	**Gate/circuit level	**Power and delay	**Yes	**Yes

IV. PROPOSED MACHINE LEARNING-ASSISTED OPTIMISATION FRAMEWORK

4.1 Overview

The framework, shown in Fig. 2, is organised as six stages linked by two feedback loops. Stage 1 parameterises the design space. Stage 2 generates an initial sample and evaluates it with the reference simulator. Stage 3 converts each sampled design into a feature representation combining an attributed graph with scalar descriptors. Stage 4 trains an ensemble surrogate on these features. Stage 5 performs a constrained multi-objective search over the surrogate. Stage 6 verifies the resulting candidate set with full simulation under corners and statistical variation. The first feedback loop returns high-uncertainty candidates to the simulator so the surrogate is refined where it is weakest; the second returns constraint-violating candidates to the optimiser.

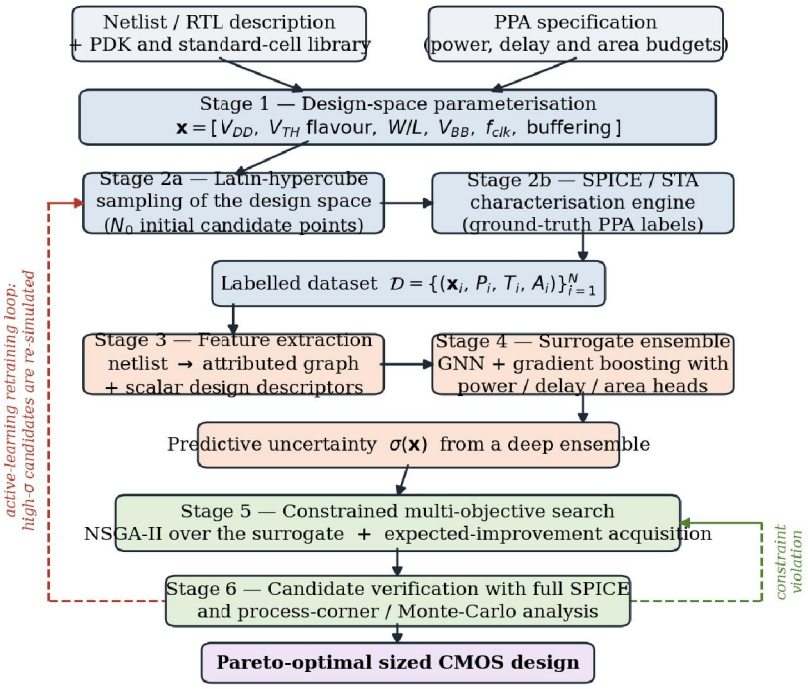

Fig. 2. Architecture of the proposed framework. Blue blocks denote data preparation, orange the learned surrogate, and green the optimisation and verification stages. The dashed red path is the active-learning retraining loop; the dashed green path returns constraint-violating candidates to the optimiser.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

4.2 Stage 1: Design-space Parameterisation

The design vector of (6) is instantiated per benchmark. Continuous variables are normalised to the unit interval using the bounds in Table II so no single variable dominates the sampler's distance metric or the tree ensemble's split criteria. Discrete variables use one-hot encodings at cell level, additionally summarised by aggregate statistics — the fraction of cells assigned to each threshold flavour, and the fraction of critical-path cells so assigned — which are much lower-variance features for the tabular branch than the raw per-cell assignment. To keep the sizing subproblem manageable, cells are grouped into clusters by topological depth and load class with a single width multiplier per cluster, reducing M from the cell count to a few tens of clusters at modest cost in achievable optimality and substantially improving conditioning by removing large numbers of nearly-equivalent design vectors.

TABLE II. DESIGN-SPACE VARIABLES, TYPES AND RANGES

Variable	Symbol	Type	Range or levels	Dimension
Supply voltage	V_DD	Continuous	0.55 – 1.10 V	1 – 3 domains
Threshold flavour	v_th	Categorical	{LVT, SVT, HVT}	per cell cluster
Width multiplier	w	Continuous	0.5× – 4.0× nominal	16 – 48 clusters
Body-bias voltage	V_BB	Continuous	−0.4 – +0.2 V	1 – 2 wells
Buffer insertion	b	Binary	{0, 1} per site	12 – 64 sites
Target clock period	f_clk	Continuous	0.6× – 1.6× nominal	1
Gate-oxide flavour	t_ox	Categorical	{thin, thick}	per cell cluster

4.3 Stage 2: Sampling and Ground-truth Characterisation

The initial sample is drawn by Latin-hypercube sampling, which stratifies each marginal and attains substantially better space-filling properties than independent uniform sampling at the same budget [34]: each axis is partitioned into N_0 equiprobable strata with one point per stratum and the assignment across axes randomly permuted, while categorical variables use stratified allocation proportional to the desired flavour mix. Designs violating hard constraints before simulation are rejected and resampled, so the budget is not spent on infeasible points.

Each sampled design is instantiated as a netlist with the corresponding cell variants, simulated at transistor level for power and characterised by static timing analysis for delay. Average power is measured over a pseudorandom input sequence of fixed length and activity factor, with leakage measured separately in the quiescent state so the two components can be attributed independently in the learned model. Critical-path delay is the worst arrival time across all endpoints and area is taken from the cell library, giving one training record (x_i, P_i, T_i, A_i).

4.4 Stage 3: Feature Representation

Each design is represented in two complementary ways. The graph representation constructs an attributed directed graph $G = (V, E)$ in which vertices are cell instances and edges are net connections. Vertex attributes comprise cell type, assigned threshold flavour, cluster width multiplier, input and output pin capacitance, logic depth from the nearest primary input, fan-in and fan-out degree and estimated switching activity; edge attributes comprise estimated wire capacitance and resistance and a critical-path flag. The scalar representation is a fixed-length vector holding the global variables, aggregate statistics of the per-cell assignments, and structural summaries such as cell count, maximum and mean logic depth, total pin count and fan-out distribution.

Both representations are retained because they fail in different regimes. The graph branch captures topology-dependent effects no fixed-length summary can express, but needs more data and is more sensitive to hyperparameters. The scalar branch is data-efficient and stable but blind to structure. Section 6.7 quantifies the contribution of each.

4.5 Stage 4: Surrogate Model

The graph branch is a five-layer graph convolutional network following (8), with hidden dimension 128, ReLU activations, batch normalisation after each propagation step and residual connections between consecutive layers. Vertex embeddings are aggregated by concatenating mean- and max-pooling, preserving both average behaviour and the influence of extreme vertices such as the most heavily loaded gate on the critical path, and the pooled embedding

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

passes through a two-layer perceptron. The tabular branch is a gradient-boosted tree ensemble in the regularised formulation of [22], with squared-error loss, maximum depth six, learning rate 0.05, row and column subsampling of 0.8 and cross-validated L2 regularisation. Tree ensembles were preferred to kernel methods because the scalar feature vector mixes continuous, ordinal and one-hot variables on very different scales, where tree-based splits are considerably more robust than distance-based kernels [35].

Branch outputs are combined by a learned convex combination fitted on a held-out validation split, separately for each target. To obtain an uncertainty estimate, K = 5 independent copies of the full model are trained with different initialisations and different bootstrap resamples, following the deep-ensemble construction [36]. The predictive mean and variance are

$$\mu(x) = \left(\frac{1}{K}\right)\Sigma_k \hat{f}_{k(x)}, \quad \sigma^2(x) = \left(\frac{1}{K}\right)\Sigma_k \left[\hat{f}_{k(x)} - \mu(x)\right]^2 \qquad (9)$$

The variance gates the active-learning loop and is reported alongside every prediction, so a designer can distinguish a confident estimate from an extrapolation.

4.6 Stage 5: Constrained Multi-objective Search

The optimiser is NSGA-II [10], chosen because it handles mixed discrete-continuous representations naturally, maintains a diverse approximation to the entire Pareto front in a single run rather than requiring a scalarisation sweep, and is insensitive to non-convexity. Real-valued genes use simulated binary crossover and polynomial mutation; categorical genes use uniform crossover and flip mutation. Constraints are handled by the constrained-domination rule, under which a feasible solution always dominates an infeasible one and two infeasible solutions are compared by total violation. Fitness is evaluated on the surrogate rather than the simulator, reducing the cost of one evaluation from seconds to microseconds. To avoid converging onto regions where the surrogate is optimistically wrong, fitness is penalised by the predictive standard deviation:

$$F_{j(x)} = \mu_{j(x)} + \lambda \bullet \sigma_{j(x)}, \quad j \in \{power, delay\} \qquad (10)$$

with λ = 1.0 in the reported experiments. This is a lower-confidence-bound formulation and the multi-objective analogue of the expected-improvement principle [27]: it discourages the optimiser from exploiting model error. At the end of each generation block the ρ = 5 % of the population with the highest σ is submitted to the simulator, the results are added to D, the surrogate is retrained and the search resumes.

4.7 Stage 6: Verification

No design point is reported on the basis of a surrogate prediction alone. The non-dominated set from Stage 5 is re-simulated at transistor level under the nominal, slow-slow and fast-fast corners and at the temperature extremes of the target specification. A Monte-Carlo analysis of 1000 trials with correlated global and independent local mismatch parameters is then run on the survivors, and any point whose 3σ delay violates timing or whose 3σ power exceeds the budget is discarded. Failing points are returned to the optimiser with their measured violation, which both removes them from the reported front and supplies a particularly informative training record.

4.8 Computational Complexity

Let N_s be the number of simulator calls, c_s the cost of one call, G the number of generations, p the population size and c_m the cost of one surrogate evaluation. Direct evaluation of NSGA-II on the simulator costs O(G · p · c_s). The proposed framework costs O(N_s · c_s + G · p · c_m + R · c_t), where R is the number of retraining rounds and c_t the cost of one retraining. Because c_m is smaller than c_s by roughly six orders of magnitude and N_s ≪ G · p by construction, the dominant term is N_s · c_s. The achievable speedup is therefore approximately G · p / N_s, bounded above by the ratio of search budget to simulation budget and limited in practice by the sample complexity of the surrogate — the point at which reducing N_s further degrades accuracy enough to compromise the returned front. Section 6.4 characterises this empirically.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

V. EXPERIMENTAL SETUP

5.1 Benchmark Circuits

Eight digital benchmarks were selected to span the structures encountered in practice: regular arithmetic with long carry chains, tree-structured arithmetic with high fan-out, irregular control logic and a general-purpose datapath. Table III lists them with their sizes and the dimensionality of the resulting design space.

TABLE III. BENCHMARK CIRCUITS AND DESIGN-SPACE DIMENSIONALITY

Benchmark	Description	Cells	Nets	Logic depth	Design variables
CLA-8	8-bit carry-look-ahead adder	168	191	9	34
KS-32	32-bit Kogge–Stone adder	742	826	11	58
WAL-16	16×16 Wallace-tree multiplier	2 914	3 208	24	71
C1908	ISCAS-85 error-correcting logic	880	913	22	49
C3540	ISCAS-85 ALU and control	1 669	1 719	31	62
C6288	ISCAS-85 16×16 multiplier	2 406	2 448	124	68
FIR-8	8-tap FIR filter datapath	3 187	3 501	27	74
ALU-32	32-bit RISC-V integer ALU	4 052	4 396	19	82

5.2 Technology, Tools and Compute Environment

All experiments used a 45 nm predictive technology model with three threshold-voltage flavours and two gate-oxide thicknesses. Transistor-level simulation used a SPICE-class simulator and timing used an open-source static timing analyser configured with the same library. The implementation flow was scripted so each candidate could be instantiated, simulated and characterised without manual intervention. Surrogates were implemented in PyTorch [38] with graph operations from PyTorch Geometric [39], and the tabular branch used the gradient-boosting implementation of [22]. Optimisation ran on a workstation with a 16-core processor, 128 GB of memory and a single GPU, with the simulator on the CPU and the surrogate on the GPU. The mean cost of one full characterisation, averaged over the eight benchmarks, was 4.1 s; this figure is the denominator of all runtime comparisons in Section 6.6.

5.3 Training Protocol

For each benchmark an initial Latin-hypercube sample of $N_0 = 400$ designs was characterised and split 70/15/15 into training, validation and test partitions, stratified by decile of measured power so the extremes were represented in all three. Models were trained with Adam at an initial learning rate of 10^{-3}, halved whenever validation loss failed to improve for ten epochs, with early stopping after thirty epochs without improvement. Targets were log-transformed before training, since power and delay are strictly positive and approximately log-normally distributed across the design space, and predictions were exponentiated back before evaluation. Table IV lists the principal hyperparameters.

TABLE IV. PRINCIPAL HYPERPARAMETERS

Component	Hyperparameter	Value
GCN branch	Propagation layers / hidden dimension	5 / 128
GCN branch	Dropout / batch size	0.2 / 32
GBT branch	Trees / max depth / learning rate	800 / 6 / 0.05
GBT branch	Row and column subsample	0.8 / 0.8
Ensemble	Number of members K	5
Optimiser	Population size / generations	120 / 300
Optimiser	Crossover / mutation probability	0.9 / 1/d
Optimiser	Uncertainty penalty λ	1.0
Active learning	Resimulation fraction ρ per round	5 %

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

Verification	Monte-Carlo trials	1 000

5.4 Baselines and Evaluation Metrics

Four baselines were implemented. The reference sizing is the output of a conventional synthesis flow at default effort with a single nominal supply, defining the 100 % point against which improvements are reported. Simulated annealing uses a geometric cooling schedule with a scalarised objective weighting power and delay equally [9]. Gaussian-process Bayesian optimisation with expected improvement operates directly on the simulator over the same space [26], [27]. NSGA-II on SPICE is the same evolutionary algorithm used in Stage 5 but evaluating every candidate with the simulator, and therefore isolates the contribution of the surrogate from that of the search algorithm [10]. All methods received identical design-space bounds and constraints.

Surrogate quality is reported as mean absolute percentage error and coefficient of determination on the held-out test partition. Optimisation quality is reported as percentage reduction in power, critical-path delay, power-delay product and energy-delay product relative to the reference sizing, and as normalised hypervolume of the returned front with respect to a common reference point [37]. Efficiency is reported as simulator calls and wall-clock time to convergence. Significance is assessed with the Wilcoxon signed-rank test over paired per-benchmark results [40], which makes no normality assumption and suits the small sample size.

VI. RESULTS AND DISCUSSION

6.1 Surrogate Accuracy

Table V compares five surrogate variants on the held-out test partition, pooled across benchmarks. Linear regression is included only as a reference point; its error confirms that the mapping from design vector to power and delay is strongly non-linear, as the physical models of Section 2 predict. Among the non-linear models the proposed hybrid ensemble is the most accurate on both targets, reducing power MAPE from 4.2 % for the gradient-boosted baseline to 2.7 % and delay MAPE from 3.8 % to 2.4 %.

TABLE V. SURROGATE MODEL ACCURACY ON THE HELD-OUT TEST PARTITION

Model	Power MAPE (%)	Power R^2	Delay MAPE (%)	Delay R^2	Train time (min)
Linear regression	12.8	0.812	11.4	0.794	0.1
Support-vector regression	7.1	0.918	6.6	0.910	3.4
Random forest	5.9	0.941	5.4	0.933	1.9
Gradient-boosted trees	4.2	0.965	3.8	0.961	2.6
Multilayer perceptron	4.6	0.958	4.1	0.954	6.8
Graph neural network only	3.3	0.976	2.9	0.972	14.2
****Proposed hybrid ensemble**	****2.7**	****0.984**	****2.4**	****0.981**	****21.5**

The gap between the graph-only model and the tabular models is instructive. It is largest on C6288, the benchmark with the greatest logic depth, where the critical path traverses 124 levels and aggregate statistics such as mean fan-out convey almost nothing about the delay-determining structure. It is smallest on CLA-8, where the circuit is regular enough that a handful of scalar descriptors nearly determine the answer. This supports the hypothesis that the graph branch contributes precisely the topological information the tabular branch cannot express.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

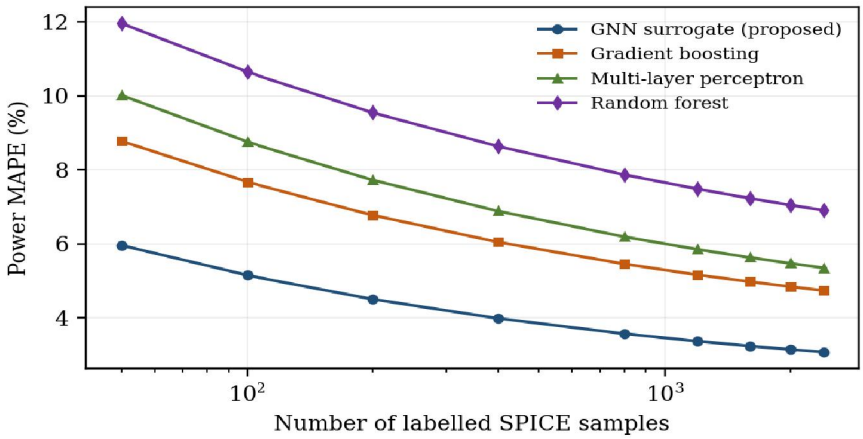

Fig. 3. Power prediction error against the number of labelled SPICE samples. The graph-based surrogate reaches a given accuracy with roughly half the training data required by the tabular baselines, and its advantage is largest in the low-data regime that matters most in practice.

Figure 3 shows the learning curves. The significance lies in the low-data regime: to reach 5 % power MAPE the gradient-boosted model needs about 600 labelled samples whereas the proposed surrogate needs about 280, which at 4.1 s per sample saves roughly twenty minutes of characterisation per benchmark before any optimisation begins.

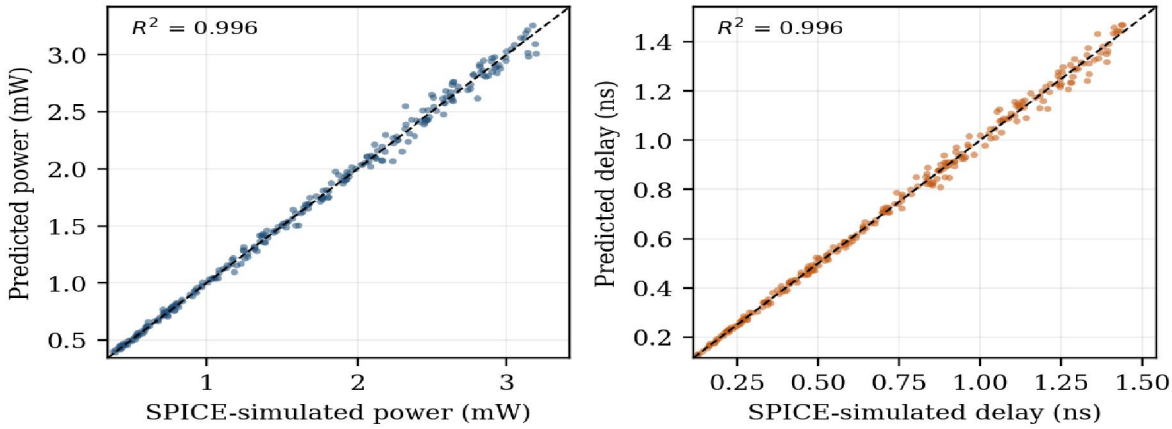

Fig. 4. Predicted against SPICE-simulated values for power (left) and delay (right) on the held-out test partition, pooled across benchmarks. The dashed line is the identity. Residuals are approximately homoscedastic in the log domain with no systematic bias at either extreme.

Figure 4 confirms that the residuals are well behaved. The absence of a systematic trend at the extremes matters for the intended use: an optimiser searches for extreme points, so a surrogate accurate on average but biased in the tails would be actively misleading. Scatter is slightly wider at the low-power end, since leakage-dominated designs are hardest to predict.

6.2 Power, Delay and Composite Figures of Merit

Table VI reports the principal results, averaged over five independent runs per benchmark with different random seeds. The framework reduces power by 27.4 % and delay by 18.6 % on average against the reference sizing, a 40.8 % reduction in power-delay product and 51.6 % in energy-delay product. NSGA-II applied directly to SPICE reaches a similar but slightly inferior front while consuming roughly twenty times more wall-clock time.

DOI: 10.48175/IJARSCT-38215

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

TABLE VI. OPTIMISATION RESULTS RELATIVE TO THE REFERENCE SIZING (MEAN OF FIVE RUNS)

Method	Power red. (%)	Delay red. (%)	PDP red. (%)	EDP red. (%)	SPICE calls	Time (h)
Reference sizing	0.0	0.0	0.0	0.0	—	—
Simulated annealing	18.3	11.2	27.4	35.5	27 800	31.6
GP-BO on SPICE	21.7	14.1	32.7	42.2	16 200	18.4
NSGA-II on SPICE	25.1	16.9	37.8	48.3	20 000	22.8
**Proposed framework	**27.4	**18.6	**40.8	**51.6	**982	**1.12

The comparison against NSGA-II on SPICE is the most informative single result, because the two methods share a search algorithm, a design space and a constraint handler, differing only in the evaluation oracle. The framework matches and slightly exceeds direct simulation-driven search using roughly one twentieth of the evaluations. The 2.3-point quality advantage is not because the surrogate is more accurate than the simulator, which is impossible; it arises because cheap evaluation permits far more generations within the same time budget, so the search explores more thoroughly.

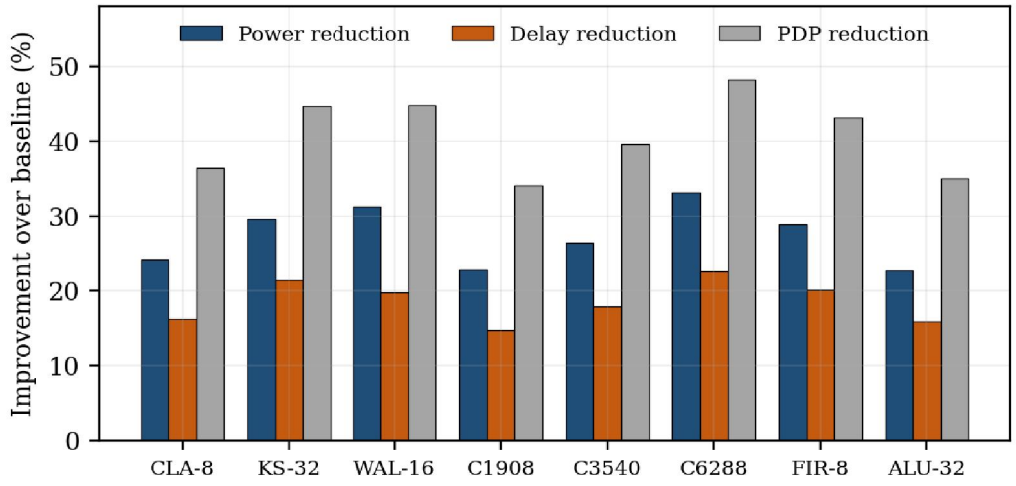

Fig. 5. Per-benchmark power, delay and power-delay-product reduction relative to the reference sizing. Improvements are largest on the deep, regular arithmetic structures where the reference flow leaves the most slack unexploited on non-critical paths.

Figure 5 disaggregates the average, and the variation is systematic rather than random. The largest improvements occur on C6288 (33.1 % power, 22.6 % delay) and WAL-16 (31.2 %, 19.8 %), deep multiplier structures with many non-critical paths on which the reference flow leaves timing slack unexploited; the framework converts that slack into leakage savings by assigning high-threshold cells and reduced widths off the critical path. The smallest improvements occur on CLA-8 (24.1 %, 16.2 %) and ALU-32 (22.7 %, 15.9 %) — in the first case the circuit is small enough that the reference flow already approaches the achievable optimum, and in the second the logic depth is shallow and the critical path passes through a large fraction of the cells, leaving little slack to exploit.

6.3 Pareto Front Quality

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

A single-point comparison understates the value of a multi-objective method, since the practical benefit is the ability to choose an operating point after the fact. Figure 6 shows the fronts returned for the ALU-32 benchmark.

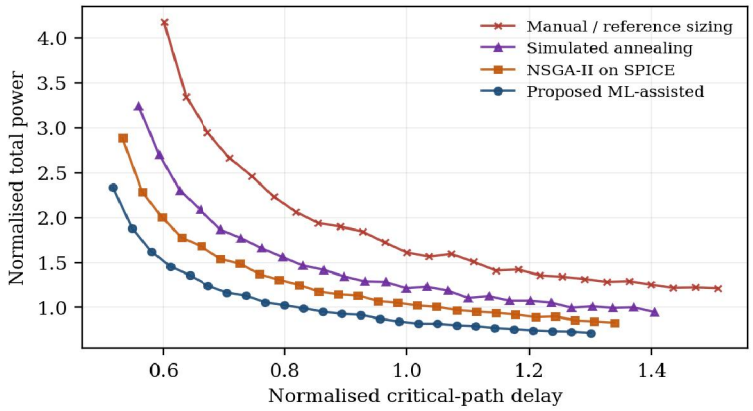

Fig. 6. Pareto fronts in the normalised power-delay plane for ALU-32. The proposed framework dominates the baselines across the whole trade-off range, and its front is both closer to the origin and more densely and evenly populated.

Two properties matter. The first is proximity to the origin, quantified by the hypervolumes in Table VII. The second is density and evenness along the front, since a designer usually needs a solution at a specific delay target rather than at whichever delay the optimiser happened to return. The framework produces the denser front because surrogate evaluation is cheap enough to maintain a large population for many generations, whereas simulator-driven search must keep the population small to remain tractable.

TABLE VII. NORMALISED HYPERVOLUME OF THE RETURNED PARETO FRONTS

Benchmark	Simulated annealing	GP-BO	NSGA-II on SPICE	Proposed
CLA-8	0.804	0.851	0.869	**0.897
KS-32	0.791	0.844	0.863	**0.906
WAL-16	0.776	0.831	0.858	**0.914
C1908	0.812	0.849	0.861	**0.892
C3540	0.798	0.840	0.857	**0.903
C6288	0.769	0.827	0.854	**0.921
FIR-8	0.783	0.836	0.859	**0.910
ALU-32	0.807	0.847	0.864	**0.895
**Mean	**0.793	**0.841	**0.861	**0.905

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

6.4 Convergence Behaviour

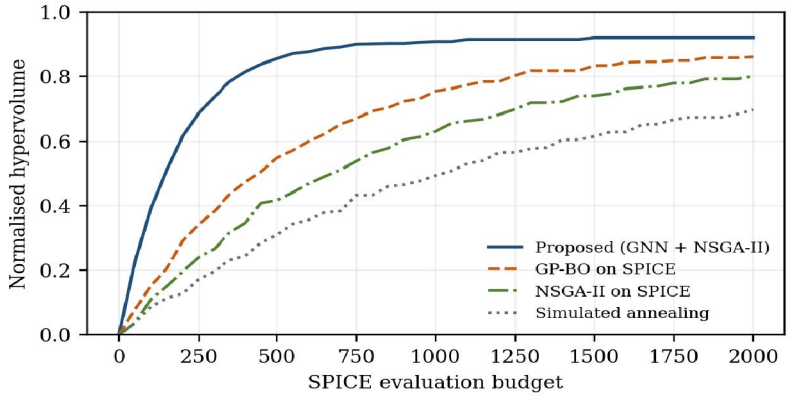

Fig. 7. Normalised hypervolume against SPICE evaluation budget. The framework reaches the final quality of NSGA-II on SPICE after roughly 300 simulator calls, a budget at which the simulator-driven baselines have achieved less than three quarters of their eventual hypervolume.

Figure 7 plots normalised hypervolume against simulator calls consumed. The three simulator-driven baselines improve slowly because every unit of progress costs a full characterisation. The proposed framework rises steeply during the first two hundred calls — the period in which the surrogate is established — then continues improving at low marginal cost, since further progress is driven by surrogate evaluations with only occasional confirmatory simulation. The crossover point matters for practice: the framework attains the terminal hypervolume of NSGA-II on SPICE after roughly 300 calls, a budget at which NSGA-II itself has realised about 72 % of its eventual value. Below about 120 calls the ordering reverses, because with too few labelled samples the surrogate is inaccurate enough that the uncertainty penalty in (10) dominates the fitness and the search stalls. This defines the regime of applicability: the method needs a minimum viable training set, below which direct search is preferable.

6.5 Statistical Significance

Because eight benchmarks constitute a small sample and the per-benchmark improvements are not normally distributed, significance was assessed with the Wilcoxon signed-rank test on paired results [40]. Table VIII reports the outcome against each baseline on the power-delay product.

TABLE VIII. WILCOXON SIGNED-RANK TEST, PROPOSED FRAMEWORK VERSUS BASELINES (n = 8)

Comparison	Median PDP difference (pp)	W statistic	p-value	Effect size r
vs. reference sizing	+40.6	0	0.0078	0.88
vs. simulated annealing	+13.2	0	0.0078	0.88
vs. GP-BO on SPICE	+8.1	1	0.0156	0.83
vs. NSGA-II on SPICE	+3.0	3	0.0391	0.73

All four comparisons are significant at the 5 % level. The comparison against NSGA-II on SPICE is the weakest, at p = 0.039, which is expected: the two methods share a search algorithm and differ only in evaluation, so the quality difference is genuinely modest. The substantive claim of this paper is not that the surrogate improves solution quality dramatically, but that it delivers comparable or slightly better quality at roughly one twentieth of the cost.

6.6 Runtime and Computational Cost

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

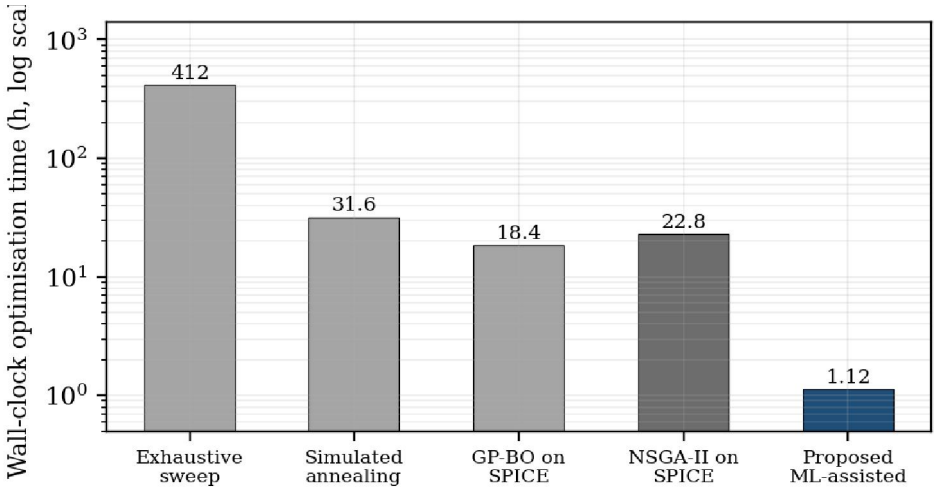

Fig. 8. Wall-clock optimisation time on a logarithmic scale, averaged over the eight benchmarks. The exhaustive-sweep figure is an extrapolation from a coarse grid over a reduced subspace, included to convey the scale of the underlying problem rather than as a practical baseline.

The framework completes in 1.12 h on average against 22.8 h for NSGA-II on SPICE, a speedup of 20.4×. The breakdown is informative: 982 simulator calls at 4.1 s each account for essentially the whole 1.12 h, surrogate training for 21.5 min accumulated across all retraining rounds, and the 36 000 surrogate evaluations performed by the optimiser for under 40 s. The framework is therefore simulation-bound by design, and its cost is controlled almost entirely by the sampling policy.

6.7 Ablation Study

TABLE IX. ABLATION STUDY (MEAN OVER EIGHT BENCHMARKS)

Configuration	PDP red. (%)	Δ (pp)	SPICE calls	Corner-valid points (%)
**Full framework	**40.8	**—	**982	**100
without graph branch	37.0	−3.8	1 581	100
without tabular branch	39.1	−1.7	1 104	100
without deep ensemble	35.7	−5.1	1 744	100
without active learning	37.5	−3.3	1 400	100
without uncertainty penalty λ	38.2	−2.6	1 020	100
without verification stage	41.6	+0.8	704	86

Table IX isolates each component by removing it and re-running the pipeline. Removing the graph branch costs 3.8 percentage points of PDP reduction and raises the simulation budget by 61 %. Removing the ensemble — and therefore the uncertainty estimate — costs 5.1 points, because active learning degenerates to random resampling and the penalty in (10) can no longer be applied; removing active learning alone costs 3.3 points.

The last row deserves comment. Removing verification appears to improve reported PDP by 0.8 points while cutting the simulation budget by 28 %. This is an artefact: the apparent improvement consists entirely of design points that do not survive corner analysis, 14 % of the returned front. It is a concrete illustration of a failure mode to which surrogate-assisted optimisation is particularly prone — the optimiser will exploit any region where the model is optimistic, and without a ground-truth check those regions are indistinguishable from genuine optima.

DOI: 10.48175/IJARSCT-38215

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

6.8 Cross-node and Cross-benchmark Transfer

A surrogate that must be retrained from scratch for every design captures much less of the potential value than one that transfers. Two experiments were performed: a model trained on seven benchmarks and fine-tuned on the eighth with a reduced budget, and a model trained at 45 nm and fine-tuned on a 32 nm library.

TABLE X. TRANSFER LEARNING RESULTS

Transfer setting	Samples from scratch	Samples after transfer	Reduction	Final power MAPE (%)
Held-out benchmark (same node)	400	140	65 %	3.1
45 nm → 32 nm, same benchmarks	400	185	54 %	3.4
45 nm → 32 nm, new benchmark	400	255	36 %	3.9

Transfer is effective but incomplete. Fine-tuning on a held-out benchmark at the same node needs 65 % fewer labelled samples than training from scratch; transfer across technology nodes needs 36–54 % fewer with higher residual error, because the graph structure and the qualitative relationships between variables carry over but the quantitative device characteristics do not. This is consistent with the finding in [16] that a graph encoder is necessary for cross-node transfer in analog sizing.

6.9 Interpretability

A model that cannot be interrogated is difficult to trust in a flow where sign-off responsibility rests with a human engineer. SHAP values were computed on the learned power model to attribute predictions to input features [41].

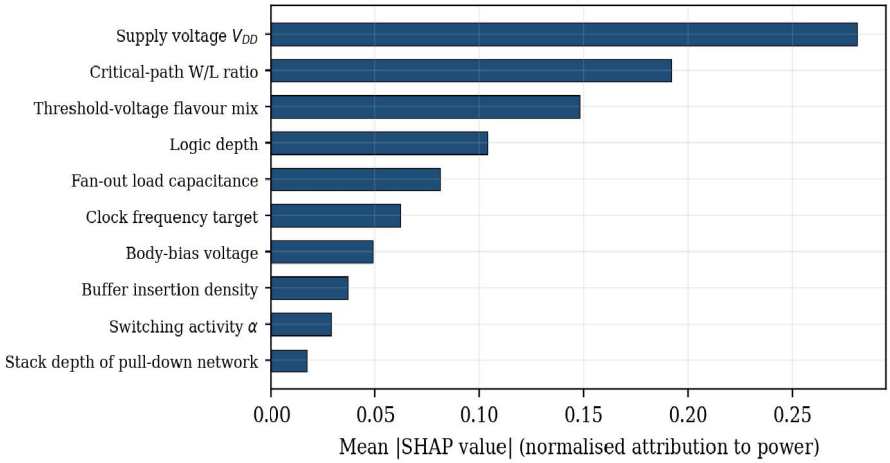

Fig. 9. Mean absolute SHAP attribution of the learned power model to each input feature. The ordering recovers the structure of the analytical power model of Section 2.1 without having been given it.

The result in Fig. 9 is reassuring rather than surprising, which is the point. Supply voltage dominates at 0.281, consistent with the quadratic dependence in (2); critical-path sizing follows at 0.192 and threshold-flavour mix at 0.148, consistent with the exponential leakage dependence in (4); logic depth, load capacitance and target frequency follow in an order matching the analytical model. The surrogate was never given the equations of Section 2 and recovered their qualitative structure from data. Had the attribution instead been dominated by an incidental feature such as cell count, that would have been evidence of a spurious correlation in the training set.

6.10 Threats to Validity

Several limitations qualify these results. The benchmarks, though structurally diverse, are blocks of a few thousand cells — two orders of magnitude smaller than a modern processor core — and it is not established that the surrogate scales to that regime unmodified, since the graph branch's memory footprint grows linearly with cell count. The technology is a predictive model rather than a foundry PDK, so absolute improvement magnitudes should not be read

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

as production figures, though the relative comparisons between methods, which share the same technology, are more robust. The reference sizing used default synthesis effort, and a more aggressively tuned reference would narrow the reported gap. Finally, the framework optimises average power at a fixed activity factor; a workload-dependent activity profile would shift the balance between dynamic and static terms and hence the location of the optimum.

VII. CHALLENGES AND OPEN PROBLEMS

7.1 Data Scarcity and the Cost of Labels

The central economic problem of this field is that labels are expensive: every training sample requires a simulation whose cost is precisely what the method exists to avoid. Public datasets such as CircuitNet have relieved this for the tasks they cover [18], but no comparable resource exists for gate-level power-delay optimisation, and the transfer results of Section 6.8 suggest a dataset built at one node and cell library will only partially serve another. Promising directions include synthetic netlist generation reproducing the structural statistics of real designs, multi-fidelity schemes combining many cheap approximate evaluations with few accurate ones, and physics-informed regularisation that constrains the model to respect known analytical relationships and so reduces the data needed to identify it.

7.2 Generalisation and Distribution Shift

Surrogates trained on one distribution degrade on another, and the degradation is not always accompanied by an obvious signal. A model trained predominantly on arithmetic datapaths will be confidently wrong on control-dominated logic with very different fan-out statistics. The ensemble uncertainty helps, but ensembles are known to be over-confident under genuine distribution shift, since all members share a training distribution and therefore make correlated errors outside it. Reliable out-of-distribution detection for circuit representations remains open.

7.3 Interpretability and Designer Trust

Sign-off decisions carry engineering and commercial liability, and a designer cannot delegate that responsibility to a model whose reasoning is inaccessible. Post-hoc attribution such as SHAP [41] provides a partial answer, as Section 6.9 illustrates, but attribution is not explanation: knowing that supply voltage accounts for 28 % of a prediction does not establish that the model behaves correctly outside the training range. Intrinsically interpretable surrogates — models constrained to the functional form of the analytical power equations with learned coefficients — trade accuracy for a much stronger guarantee, and the right point on that trade-off is not established.

7.4 Process, Voltage and Temperature Variation

The framework treats variation through a verification stage that filters the returned front. This is sound but conservative: it discards designs rather than steering the search away from variation-sensitive regions in the first place. A more capable formulation would have the surrogate predict the full distribution of power and delay rather than the nominal value, so the optimiser could directly minimise 3σ delay rather than the mean. This requires learning a conditional density rather than a conditional mean — a harder problem requiring more data — and it interacts awkwardly with the Pelgrom scaling of mismatch with device area [19], since the variance itself depends on the variables being optimised.

7.5 Integration with Commercial Sign-off Flows

Academic frameworks are typically evaluated on open-source tools and predictive models, whereas production designs are closed with commercial tools against foundry PDKs, and the gap is not merely one of accuracy. Commercial flows impose constraints — licence-limited parallelism, restricted scripting interfaces, incremental-mode timing engines with their own convergence behaviour, and non-disclosure obligations preventing training data from leaving the organisation — that materially affect what an automated search can do. The federated formulations of Section 8.3 address the last of these; the others remain practical obstacles.

7.6 Reproducibility and Benchmarking

Published results in this area are hard to compare because authors use different technologies, benchmarks, reference flows and definitions of improvement. A 30 % power reduction against a weak reference is not comparable to a 10 % reduction against a strong one. The field would benefit from a common benchmark suite with a fixed reference

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

implementation and a standard reporting format, analogous to the role the ISPD and CAD contest benchmarks played in placement research [8].

VIII. FUTURE RESEARCH DIRECTIONS

8.1 Foundation Models and Language-model Agents

The transformer architecture [45] has enabled models pretrained on broad corpora and adapted to specific tasks with comparatively little labelled data; applied to hardware, models trained on large collections of register-transfer-level code generate syntactically valid and functionally plausible Verilog [46]. The natural extension here is a pretrained circuit encoder trained on a large netlist corpus with a self-supervised objective and then fine-tuned as a PPA regressor. If such an encoder learned transferable structural representations, the per-design labelling cost identified in Section 7.1 would fall substantially. The open question is what self-supervised objective suits netlists: masked-token prediction, which works well for text, has no obvious analogue for a structure whose semantics are functional rather than sequential.

8.2 Physics-informed and Hybrid Models

Physics-informed neural networks incorporate known governing equations into the training loss so the learned function is constrained to satisfy them [43]. The analytical models of Section 2 provide exactly such constraints: the quadratic dependence of dynamic power on supply voltage and the exponential dependence of subthreshold current on threshold voltage are not approximations to be learned but relationships known a priori. A surrogate predicting only the residual between the analytical model and the simulated value would learn a much simpler function, require less data, and extrapolate more safely outside the training range, since extrapolation would be governed by the analytical term.

8.3 Federated and Privacy-preserving Learning

The most valuable training data for this problem sits inside semiconductor companies and cannot be shared. Federated learning trains a shared model by exchanging parameter updates rather than data [44], which in principle allows several organisations to build a common surrogate without disclosing netlists or PDK-specific characterisation results. The obstacles are substantial: participants' data distributions differ systematically, and parameter updates can leak information about training data, so differential-privacy guarantees would be required. Nonetheless this is the only obvious route to training-set sizes comparable to those that made progress possible in vision and language.

8.4 Beyond-CMOS and Emerging Devices

As CMOS scaling approaches its physical limits, attention is turning to alternative devices [42]. Negative-capacitance field-effect transistors, tunnel FETs, carbon-nanotube transistors and spintronic devices each present a different power-delay trade-off and — critically here — each lacks the decades of accumulated design experience and mature compact models that CMOS enjoys. This makes learned surrogates comparatively more attractive for these technologies, not less, since the alternative is not a well-tuned heuristic but no heuristic at all. The methodology developed here is agnostic to device physics: it requires only a simulator that can evaluate a design vector, and that vector is structurally analogous to (6) for an emerging-device circuit.

8.5 Toward Closed-loop Autonomous Design

The longer-term direction suggested by these threads is a flow in which surrogate, optimiser and verification engine operate without per-stage human intervention, with the designer specifying intent and constraints and adjudicating the returned trade-off rather than driving the search. The results of Section 6 indicate this is technically feasible for blocks of the size studied. Extending it to full designs requires progress on all the problems of Section 7 and, equally, on the question of what verification evidence suffices for a human engineer to sign off on a design that no human sized.

IX. CONCLUSION

This paper has presented a machine learning-assisted framework for the joint optimisation of power and delay in CMOS VLSI circuits. It addresses the central difficulty of the problem — a combinatorially large design space with an expensive evaluation oracle — by learning a surrogate for the oracle and spending the simulation budget selectively

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

where the surrogate is least reliable. Its components are a hybrid surrogate combining a graph neural network over the netlist with a gradient-boosted regressor over scalar descriptors, a deep ensemble supplying calibrated predictive uncertainty, an uncertainty-guided active-learning loop, a constrained NSGA-II search with a lower-confidence-bound fitness, and a verification stage that re-simulates every returned point under process corners and Monte-Carlo variation.

On eight digital benchmarks the framework reduced power by 27.4 % and critical-path delay by 18.6 % on average relative to a reference sizing, a 40.8 % reduction in power-delay product, while completing in 1.12 h against 22.8 h for the same evolutionary search driven directly by SPICE — a 20.4× speedup at slightly better solution quality. Surrogate accuracy reached 2.7 % mean absolute percentage error for power and 2.4 % for delay. Ablation confirmed that each component contributes: the graph branch is worth 3.8 percentage points of PDP reduction, the ensemble 5.1 points and the active-learning loop 3.3 points. Transfer experiments showed that a trained surrogate reduces the labelling requirement for a new benchmark at the same node by 65 %, and across technology nodes by 36–54 %.

Three findings have implications beyond the specific framework. The ablation on the verification stage demonstrates concretely that surrogate-assisted optimisation without a ground-truth check produces design points that look better and are worse, so verification is a structural requirement rather than optional overhead. The SHAP analysis showed that the learned model recovers the qualitative structure of the analytical power equations without having been given them, which is the strongest available evidence that it has learned physics rather than dataset artefacts. The convergence analysis identified a minimum viable simulation budget below which the method underperforms direct search, which delimits its regime of applicability honestly.

The limitations set out in Sections 6.10 and 7 are real, and the largest of them — scaling to designs two orders of magnitude larger, closing the gap between predictive models and foundry PDKs, and reliably detecting when a surrogate is extrapolating — will determine whether methods of this kind enter production flows. The direction of travel across the field, from prediction toward closed-loop optimisation and from per-design models toward transferable ones, suggests the question is one of engineering effort rather than feasibility.

REFERENCES

1. R. H. Dennard, F. H. Gaensslen, H.-N. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, “Design of ion-implanted MOSFET's with very small physical dimensions,” IEEE J. Solid-State Circuits, vol. 9, no. 5, pp. 256–268, Oct. 1974, doi: https://doi.org/10.1109/JSSC.1974.1050511
2. M. Bohr, “A 30 year retrospective on Dennard's MOSFET scaling paper,” IEEE Solid-State Circuits Soc. Newsl., vol. 12, no. 1, pp. 11–13, Winter 2007, doi: https://doi.org/10.1109/N-SSC.2007.4785534
3. M. Horowitz, “1.1 Computing's energy problem (and what we can do about it),” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, San Francisco, CA, USA, 2014, pp. 10–14, doi: https://doi.org/10.1109/ISSCC.2014.6757323
4. H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, “Dark silicon and the end of multicore scaling,” in Proc. 38th Annu. Int. Symp. Computer Architecture (ISCA), San Jose, CA, USA, 2011, pp. 365–376, doi: https://doi.org/10.1145/2000064.2000108
5. A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, “Low-power CMOS digital design,” IEEE J. Solid-State Circuits, vol. 27, no. 4, pp. 473–484, Apr. 1992, doi: https://doi.org/10.1109/4.126534
6. K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, “Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits,” Proc. IEEE, vol. 91, no. 2, pp. 305–327, Feb. 2003, doi: https://doi.org/10.1109/JPROC.2002.808156
7. S. Borkar, “Design challenges of technology scaling,” IEEE Micro, vol. 19, no. 4, pp. 23–29, Jul./Aug. 1999, doi: https://doi.org/10.1109/40.782564
8. I. L. Markov, J. Hu, and M.-C. Kim, “Progress and challenges in VLSI placement research,” Proc. IEEE, vol. 103, no. 11, pp. 1985–2003, Nov. 2015, doi: https://doi.org/10.1109/JPROC.2015.2478963

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

9. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, no. 4598, pp. 671–680, May 1983, doi: https://doi.org/10.1126/science.220.4598.671
10. I. 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, doi: https://doi.org/10.1109/4235.996017B. Kahng, "Machine learning applications in physical design: Recent results and directions," in Proc. Int. Symp. Physical Design (ISPD), Monterey, CA, USA, 2018, pp. 68–73, doi: https://doi.org/10.1145/3177540.3177554
11. G. Huang et al., "Machine learning for electronic design automation: A survey," ACM Trans. Des. Autom. Electron. Syst., vol. 26, no. 5, Art. no. 40, Jun. 2021, doi: https://doi.org/10.1145/3451179
12. M. Rapp, H. Amrouch, Y. Lin, B. Yu, D. Z. Pan, M. Wolf, and J. Henkel, "MLCAD: A survey of research in machine learning for CAD," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 41, no. 10, pp. 3162–3181, Oct. 2022, doi: https://doi.org/10.1109/TCAD.2021.3124762
13. A. Mirhoseini et al., "A graph placement methodology for fast chip design," Nature, vol. 594, no. 7862, pp. 207–212, Jun. 2021, doi: https://doi.org/10.1038/s41586-021-03544-w
14. 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 Des. Integr. Circuits Syst., vol. 40, no. 4, pp. 748–761, Apr. 2021, doi: https://doi.org/10.1109/TCAD.2020.3003843
15. H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H.-S. Lee, and S. Han, "GCN-RL circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning," in Proc. 57th ACM/IEEE Design Automation Conf. (DAC), 2020, pp. 1–6, doi: https://doi.org/10.1109/DAC18072.2020.9218757
16. Z. Xie, H. Ren, B. Khailany, Y. Sheng, S. Santosh, J. Hu, and Y. Chen, "PowerNet: Transferable dynamic IR drop estimation via maximum convolutional neural network," in Proc. 25th Asia and South Pacific Design Automation Conf. (ASP-DAC), 2020, pp. 13–18, doi: https://doi.org/10.1109/ASP-DAC47756.2020.9045574
17. Z. Chai, Y. Zhao, W. Liu, Y. Lin, R. Wang, and R. Huang, "CircuitNet: An open-source dataset for machine learning in VLSI CAD applications with improved domain-specific evaluation metric and learning strategies," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 42, no. 12, pp. 5034–5047, Dec. 2023, doi: https://doi.org/10.1109/TCAD.2023.3287970
18. M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," IEEE J. Solid-State Circuits, vol. 24, no. 5, pp. 1433–1439, Oct. 1989, doi: https://doi.org/10.1109/JSSC.1989.572629
19. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015, doi: https://doi.org/10.1038/nature14539

I. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: https://doi.org/10.1023/A:1010933404324

20. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794, doi: https://doi.org/10.1145/2939672.2939785
21. C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: https://doi.org/10.1007/BF00994018
22. F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," IEEE Trans. Neural Netw., vol. 20, no. 1, pp. 61–80, Jan. 2009, doi: https://doi.org/10.1109/TNN.2008.2005605
23. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proc. Int. Conf. Learning Representations (ICLR), Toulon, France, 2017. [Online]. Available: https://arxiv.org/abs/1609.02907

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, September 2026 Impact Factor: 8.2

24. B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," Proc. IEEE, vol. 104, no. 1, pp. 148–175, Jan. 2016, doi: https://doi.org/10.1109/JPROC.2015.2494218
25. D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," J. Global Optimization, vol. 13, no. 4, pp. 455–492, Dec. 1998, doi: https://doi.org/10.1023/A:1008306431147
26. 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, pp. 1–8, doi: https://doi.org/10.1145/3240765.3240843
27. R. Kirby, S. Godil, R. Roy, and B. Catanzaro, "CongestionNet: Routing congestion prediction using deep graph neural networks," in Proc. IFIP/IEEE 27th Int. Conf. Very Large Scale Integration (VLSI-SoC), Cuzco, Peru, 2019, pp. 217–222, doi: https://doi.org/10.1109/VLSI-SoC.2019.8920342
28. H. Yang, J. Su, Y. Zou, Y. Ma, B. Yu, and E. F. Y. Young, "Layout hotspot detection with feature tensor generation and deep biased learning," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 38, no. 6, pp. 1175–1187, Jun. 2019, doi: https://doi.org/10.1109/TCAD.2018.2837078
29. M.Settaluri, A. Haj-Ali, Q. Huang, K. Hakhamaneshi, and B. Nikolić, "AutoCkt: Deep reinforcement learning of analog circuit designs," in Proc. Design, Automation and Test in Europe Conf. (DATE), Grenoble, France, 2020, pp. 490–495, doi: https://doi.org/10.23919/DATE48585.2020.9116200
30. 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, pp. 1–4, doi: https://doi.org/10.1145/3316781.3326334
31. H. Ren and J. Hu, Eds., Machine Learning Applications in Electronic Design Automation. Cham, Switzerland: Springer, 2022, doi: https://doi.org/10.1007/978-3-031-13074-8
32. [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. 21, no. 2, pp. 239–245, May 1979, doi: https://doi.org/10.1080/00401706.1979.10489755
33. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009, doi: https://doi.org/10.1007/978-0-387-84858-7
34. B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 2017, pp. 6402–6413. [Online]. Available: https://arxiv.org/abs/1612.01474
35. E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," IEEE Trans. Evol. Comput., vol. 3, no. 4, pp. 257–271, Nov. 1999, doi: https://doi.org/10.1109/4235.797969
36. A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 2019, pp. 8024–8035. [Online]. Available: https://arxiv.org/abs/1912.01703
J. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in ICLR Workshop on Representation Learning on Graphs and Manifolds, New Orleans, LA, USA, 2019. [Online]. Available: https://arxiv.org/abs/1903.02428
37. F. Wilcoxon, "Individual comparisons by ranking methods," Biometrics Bulletin, vol. 1, no. 6, pp. 80–83, Dec. 1945, doi: https://doi.org/10.2307/3001968
38. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 2017, pp. 4765–4774. [Online]. Available: https://arxiv.org/abs/1705.07874
39. J. Shalf, "The future of computing beyond Moore's Law," Philosophical Transactions of the Royal Society A, vol. 378, no. 2166, Art. no. 20190061, Mar. 2020, doi: https://doi.org/10.1098/rsta.2019.0061

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, September 2026
Impact Factor: 8.2

K. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Computational Physics, vol. 378, pp. 686–707, Feb. 2019, doi: https://doi.org/10.1016/j.jcp.2018.10.045

40. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 2017, pp. 1273–1282. [Online]. Available: https://arxiv.org/abs/1602.05629
41. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 2017, pp. 5998–6008. [Online]. Available: https://arxiv.org/abs/1706.03762
42. S. Thakur, B. Ahmad, H. Pearce, B. Tan, B. Dolan-Gavitt, R. Karri, and S. Garg, "VeriGen: A large language model for Verilog code generation," ACM Trans. Des. Autom. Electron. Syst., 2024. [Online]. Available: https://arxiv.org/abs/2308.00708