International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Machine Learning-Driven Timing Closure Optimization in Physical Design of High-Performance Silicon Architectures for Hyperscale Cloud and Consumer SoCs

Lovish Masand

Abstract: Sub-7nm FinFET and gate-all-around (GAA) silicon processes have pushed multi-corner multi-mode (MCMM) timing sign-off to a scale at which conventional static timing analysis (STA) and engineering change order (ECO) iteration cycles no longer fit within competitive tapeout schedules. Foundry-specified corner sets at these nodes can span more than a dozen hold corners alone, and parametric on-chip variation (POCV) modelling adds complexity that compounds with each additional scenario. This article examines how machine learning (ML) techniques, when integrated selectively into physical design timing closure flows, may help practitioners manage this complexity without replacing the authoritative role of STA at sign-off. A structured review covers four application areas — chip placement, pre-route slack estimation, multi-corner timing inference, and ECO convergence guidance — drawing on peer-reviewed empirical evidence from recent literature. A four-checkpoint ML-STA co-signoff framework is proposed, with each gate validated against STA ground truth before the flow advances. Evidence from the reviewed literature suggests that selective ML integration may reduce MCMM iteration cycles, improve corner coverage efficiency, accelerate IR drop estimation, and reduce false ECO convergence in ways directly applicable to ARM Neoverse-class and Cortex-class SoC design contexts. Cross-foundry model transferability and standardisation of ML-augmented sign-off remain the principal barriers to broad production adoption.

Keywords: *machine learning, timing closure, physical design, static timing analysis, MCMM, ECO, FinFET, GAA, SoC, silicon architecture*

1. Introduction

Timing closure at sub-7nm has become one of the more persistent scheduling problems in physical design. The combination of larger corner sets, tighter process variation budgets, and denser interconnect has stretched MCMM sign-off into a phase that can consume a substantial portion of available tapeout time [1]. For teams working on ARM Neoverse-class server processors or Cortex-class mobile SoCs, this is not an abstract concern. Both product families demand sign-off across aggressive frequency and power targets, and the practical question is how to achieve that without running every ECO iteration across every corner in sequence.

ECO flows carry their own compounding risk. Each correction that closes timing in one scenario can open violations in another, a failure mode known as false convergence that grows more likely as the active constraint set expands [3]. At advanced nodes, the local sensitivity of interconnect delay to layout context means that a buffer insertion affecting one path can alter parasitic loading on adjacent paths in ways not visible until the next full STA run. The schedule cost of discovering this late is significant.

ML-based approaches to timing closure reframe parts of this problem rather than solving them in the conventional sense. Instead of accelerating individual STA computations, ML methods learn statistical relationships between design features and timing outcomes, making it possible to predict violations earlier, prioritise corners more intelligently, and guide ECO decisions with better information [4]. This article presents a structured review of where that approach has produced verifiable results, proposes a four-checkpoint ML-STA co-signoff framework integrating ML at specific, well-defined stages, and identifies what remains unresolved before these methods are suitable for routine production use [5].

Independent Researcher, USA
ORCID ID: 0009-0007-3615-4246

2. Background: Physical Design and Timing Closure at Advanced Nodes

2.1 Static Timing Analysis and MCMM Flows

STA computes arrival times and required arrival times at every endpoint across all relevant clock domains, using worst negative slack (WNS) and total negative slack (TNS) as the primary closure metrics [1]. The complication at advanced nodes is that the number of PVT corners required to adequately sample the process distribution has grown considerably, and the interactions among parametric variation, multi-input switching, and crosstalk delay make accurate per-corner analysis more expensive than at older nodes [2]. AOCV and POCV models replaced flat OCV margins because the pessimism of flat derating was incompatible with achievable timing budgets at 7nm and below, but POCV's spatial correlation model introduces its own computational overhead [3]. Empirical multi-corner timing studies note that foundry-specified hold corner sets at advanced nodes may reach 14 or more corners [18], making exhaustive evaluation impractical within production schedules.

Attribute	OCV	AOCV	POCV
Variation model	Flat derating	Stage/distance-based	Statistical (sigma-based)
Pessimism level	High	Moderate	Low
Spatial correlation	None	Partial	Full
Compute cost	Low	Moderate	High
Typical node	>=28nm	16–28nm	<=7nm

Table 1. Comparison of OCV, AOCV, and POCV variation-aware timing methodologies. Author's own analysis.

2.2 Physical Design Flow: Floorplan to Sign-off

Timing closure is embedded in a feedback loop that spans placement, clock tree synthesis (CTS), and routing. Placement determines wire length distributions that shape interconnect delay; CTS determines skew that affects setup and hold margins; routing finalises parasitics that govern sign-off accuracy [2]. The practical difficulty is that decisions made at placement have timing consequences not fully visible until post-route STA, creating a gap of several flow stages between action and consequence. ECO provides the corrective mechanism, but at advanced nodes the density of timing-critical paths and sensitivity of local layout context mean that ECO changes carry a higher probability of introducing new violations elsewhere [3]. This gap between early design decisions and late timing visibility is a structural problem that ML prediction is well-positioned to address.

2.3 Challenges Specific to Cloud and Consumer SoCs

Neoverse-class processors operate across performance, balanced, and power-saving modes, each with its own timing constraints and corner requirements, substantially increasing MCMM corner count and sign-off constraint complexity [8]. Cortex-class designs face different pressures: area and power efficiency take priority, and the buffers and cell upsizes used as ECO strategies in unconstrained designs are often unavailable within tight area budgets. Both product families illustrate why a single ECO or sign-off strategy rarely transfers cleanly from one design context to another.

3. Machine Learning Applications in Timing Closure

3.1 ML-Augmented Floorplanning and Placement

Placement quality is a primary determinant of timing closure difficulty in subsequent stages. Traditional analytical placement optimises wire length and congestion estimates as proxies for timing, but the relationship between those proxies and post-route slack is imperfect [2]. Reinforcement learning-based placement reframes the problem: a policy network learns to sequence macro and cell placements in

ways that improve predicted post-routing PPA outcomes. Mirhoseini et al. demonstrated that this approach could produce chip layouts competitive with those generated by human expert engineers on power, performance, and area metrics, with placements generated in under 6 hours [6]. A placement that achieves lower wire length and better congestion distribution tends to produce fewer post-route timing violations, reducing the number of ECO iterations needed to reach sign-off. For Neoverse-class designs where macro count and routing congestion are significant closure risks, the ability to evaluate more placement configurations in a fixed schedule window translates directly into improved post-route timing margin at tapeout.

3.2 Pre-Route Slack Prediction

Pre-route slack prediction aims to close the gap between placement decisions and their timing consequences by providing estimates before routing is performed. Graph neural network (GNN) approaches model the netlist as a directed graph and propagate timing information through it in a manner analogous to a timing engine, enabling timing-driven guidance for placement and CTS optimisation at stages where full STA has not traditionally been practical [4]. The GNN approach explored by Guo et al. demonstrates that timing engine-inspired graph propagation can achieve substantially faster slack estimation than a complete routing and STA flow [14]. The transferability of such models to novel designs across process nodes remains an active research question and a practical consideration for teams working across multiple design families.

Method	Approach	Key Metric	Note
GNN timing engine [14]	Graph propagation mimicking STA	R2 = 0.8957 (test)	130nm SkyWater; conceptual citation — no metrics used in text
ML corner inference [18]	Dominant corner selection + ML inference	LESS10 = 98.2%; 7x STA	ITC'99 benchmarks + industrial validation
DRC hotspot (J-Net) [9]	Custom CNN on layout features	TPR 93.0%; AUC 0.958	Sub-10nm; inference <1 min/design

Table 2. Summary of selected ML approaches for pre-route timing prediction and sign-off support. Author's analysis of reviewed literature.

3.3 Multi-Corner Timing Inference

Dominant corner selection and corner inference address the computational cost of full MCMM evaluation by identifying a minimal corner subset from which the remaining corners can be predicted with acceptable accuracy. Zhao et al. proposed an iterative dominant corner selection strategy that achieves a LESS10 prediction accuracy of 98.2% with a mean absolute error of 2.18 ps, compared to 83.2% LESS10 for the greedy deletion baseline — an improvement of 15% [18]. At two dominant corners, the approach achieves a 7x acceleration of STA evaluation. In an industrial application using two dominant corners to predict twelve non-dominant corners from a foundry-specified set of fourteen hold corners, the method accelerated timing closure process runtime by more than 2x [18]. A seven-fold STA acceleration at the corner evaluation stage, combined with a greater than two-fold speedup in industrial closure runtime, is directly applicable to production sign-off flows for both Neoverse-class and Cortex-class design timelines.

3.4 ECO Convergence and False Positive Reduction

False ECO convergence — where a correction that closes timing in one scenario introduces violations in another — is among the more costly failure modes in advanced node physical design. ML-based ECO guidance targets this problem by improving the accuracy of timing prediction used to evaluate candidate ECO actions before they are committed. When ECO decisions are based on predictions that more accurately reflect post-route parasitics and multi-corner behaviour, the probability of introducing undetected violations in unanalysed scenarios decreases. Barboza et al. demonstrated that an ML-based pre-route timing estimation approach may reduce false ECO convergence by

approximately two-thirds compared to conventional ECO flow baselines [10]. A reduction of that magnitude, applied consistently across ECO iterations, could substantially shorten the path to sign-off for high-complexity SoC designs where ECO cycle count is a primary schedule driver.

Method	False Positive Reduction	Source
ML pre-route timing estimation	~2/3 reduction in false ECO convergence	Barboza et al. [10], DAC 2019
GR-to-DR ML consistency	Improved post-DR WS (conceptual)	Chhabria et al. [12], TODAES 2023

Table 3. ML techniques for ECO closure and false convergence reduction. Sources as cited.

4. ML-STA Co-Signoff Framework

4.1 Framework Architecture

The framework proposed here organises ML-augmented timing analysis into four sequential checkpoints. Checkpoint 1 applies pre-route slack prediction and DRC hotspot estimation after placement, flagging high-risk timing and layout regions before routing begins. Checkpoint 2 applies multi-corner inference during sign-off, reducing the number of full STA evaluations needed for MCMM coverage. Checkpoint 3 applies GR-to-DR timing consistency prediction at the transition from global to detailed routing. Checkpoint 4 applies IR drop and power integrity estimation during sign-off, identifying power grid interactions with timing margin before the final STA run. Each checkpoint cross-validates ML predictions against spot-check STA results before the flow advances. ML predictions at each checkpoint prioritise and focus STA evaluations — they do not replace STA at sign-off. The co-signoff concept keeps ML and STA synchronised throughout the flow, with ML providing early warning and prioritisation and STA providing the authoritative closure verdict.

4.2 Timing Consistency Between Global and Detailed Route

One of the more reliable sources of late-stage timing surprises in advanced SoC design is the degradation of slack estimates between global routing and detailed routing. Global routing produces wire length and congestion estimates used by the timing engine, but detailed routing introduces detour paths, via stacking, and local congestion effects that can substantially change actual interconnect parasitics from those predicted at the GR stage [12]. When GR-stage estimates are optimistic relative to post-DR actuals, the design may reach sign-off with undetected violations requiring late ECO that is expensive in both schedule and area. An ML model trained on GR-stage design features to predict post-DR timing consistency, as explored by Chhabria et al. [12], enables earlier detection of GR-to-DR degradation, allowing corrective action before detailed routing is committed. Checkpoint 3 of the co-signoff framework implements this gate as a mandatory validation step before sign-off STA is initiated.

4.3 IR Drop and Power Integrity

Dynamic IR drop reduces effective supply voltage at logic cells, increasing gate delay in ways that standard STA may not capture. At sub-10nm nodes, the sensitivity of gate delay to supply variation is high enough that power integrity is a first-order timing concern. PowerNet, proposed by Xie et al., applies a maximum convolutional neural network to transferable dynamic IR drop estimation and achieves a 30-times speedup over a commercial IR drop analysis tool on a design with approximately two million cells [17]. Across all evaluated designs, the model surpasses the previous best ML method for vectorless IR drop by 9% in prediction accuracy, with ROC AUC values above 0.9 and 0.95 at 1x1 and 5x5 tile granularities respectively [17]. A power grid mitigation tool guided by PowerNet predictions reduced IR drop hotspots by 26% and 31% on two industrial designs with limited modification to the power delivery network [17]. Integrating IR drop prediction at checkpoint 4 enables concurrent power grid correction and ECO rather than sequential treatment, reducing full sign-off cycles — particularly relevant for Neoverse-class designs where power delivery network complexity is a known closure risk.

4.4 DRC Hotspot Prediction

DRC violations discovered during or after routing require layout modifications that can disturb nearby timing-critical paths, reopening violations that were previously closed. Predicting DRC hotspots before routing begins allows the physical design flow to avoid high-risk layout configurations before they are committed. J-Net, the customised convolutional network proposed by Liang et al. for DRC hotspot prediction at sub-10nm nodes, achieved a true positive rate of 93.0% and an area under the ROC curve of 0.958 on seen designs, with a false positive rate of 9.8% [9]. TPR improvements over three prior methods ranged from 14% to 40% at comparable false positive rates. Inference time was under one minute per design, compared to several hours for global routing-based approaches [9] — a runtime compatible with integration into early-stage placement and routing decision loops. In the co-signoff framework, DRC hotspot prediction contributes to checkpoint 1, enabling placement to respond to both timing risk and physical realisation risk simultaneously.

Gate	ML Technique	Key Metric	Source
1 — Pre-route	GNN slack prediction; DRC hotspot CNN	TPR 93.0%, AUC 0.958 [9]	Liang et al. [9]; Guo et al. [14]
2 — MCMM	Dominant corner selection + ML inference	LESS10 98.2%; 7x STA; >2x closure [18]	Zhao et al. [18]
3 — GR-to-DR	XGBoost parasitic correction model	Improved WS (conceptual) [12]	Chhabria et al. [12]
4 — IR drop	Max-CNN IR drop estimation (PowerNet)	30x speedup; 26–31% hotspot reduction [17]	Xie et al. [17]

Table 4. ML-STA co-signoff framework: checkpoint summary with ML technique, key metric, and source.

5. Discussion

5.1 Industrial Applicability and Limitations

Among the techniques reviewed here, reinforcement learning-based placement is closest to production deployment, having been demonstrated at scale in several EDA toolchain contexts. Multi-corner timing inference and pre-route slack prediction have produced encouraging results on benchmark circuits and selected industrial designs, but their behaviour on novel process nodes and previously unseen design families warrants further systematic evaluation before they are treated as production-ready. ECO guidance ML has a promising empirical foundation in the false convergence reduction results of Barboza et al. [10], but those results have not yet been independently replicated across a broad range of design and process conditions. Training data dependency is the most consistent limitation across all reviewed approaches. A model trained on one process node's design kit and library characterisation may not transfer reliably to a different node or foundry — a practical adoption barrier for engineers working across multiple foundry partnerships and process generations.

5.2 Open Research Challenges

Three challenges stand out as priorities for the field. Multi-foundry transferability of timing ML models requires transfer learning and domain adaptation methods capable of bridging differences in device models, parasitic extraction rules, and library characterisation across foundry partners — prerequisites for deployment in multi-foundry design environments. ML-STA co-signoff standardisation requires agreed interfaces and validation protocols independent of specific EDA toolchain implementations, reducing the integration burden and improving result reproducibility across organisations. Commercial EDA tool integration — specifically the incorporation of ML timing prediction within commercial sign-off tools — remains the highest-impact near-term direction for industry investment, and is identified by recent surveys as a key gap between research prototype and production infrastructure [5].

5.3 Future Directions

Federated learning for cross-foundry timing model sharing may allow multiple design teams to benefit from shared model training without exposing proprietary design data. GNN architectures with explicit timing engine priors, as explored by Guo et al. [14], offer a direction for improving pre-route timing prediction accuracy while maintaining physical interpretability important for sign-off trust. Extending reinforcement learning-based optimisation beyond placement to cover CTS and routing decisions could further reduce physical design iteration counts on advanced SoC designs, compounding benefits already observed at the placement stage. Progress across all three directions would move ML-augmented timing closure from a collection of promising research results toward a coherent, production-deployable methodology.

6. Conclusion

At sub-7nm FinFET and GAA process nodes, MCMM timing sign-off has reached a scale at which the computational cost of conventional STA iteration cycles is a genuine design schedule constraint. The evidence reviewed here suggests that ML integration at specific, well-defined points in the physical design flow may help practitioners manage that constraint. ML-driven dominant corner selection may achieve up to 98.2% timing prediction accuracy with a 7x STA acceleration and more than 2x industrial closure speedup [18]. ML-guided IR drop estimation may achieve a 30-times speedup over commercial analysis tools with IR drop hotspot reductions of 26–31% [17]. ML-based ECO guidance may reduce false convergence by approximately two-thirds, shortening sign-off iteration cycles for high-complexity SoC designs [10]. The four-checkpoint ML-STA co-signoff framework proposed here provides a structured approach to incorporating these capabilities while preserving the authority of STA sign-off. For practitioners working on ARM Neoverse-class and Cortex-class SoC designs, the most immediately applicable results involve multi-corner timing acceleration and IR drop estimation speedup — both of which address constraints directly schedule-critical in advanced node tapeout programmes. Wider adoption depends on progress in cross-foundry model transferability, co-signoff standardisation, and EDA tool integration: challenges that are technically tractable and that the research community is actively pursuing.

References

[1] J. Bhasker and R. Chadha, Static Timing Analysis for Nanometer Designs: A Practical Approach. New York: Springer, 2009.

[2] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, VLSI Physical Design: From Graph Partitioning to Timing Closure. New York: Springer, 2011.

[3] L. Scheffer, L. Lavagno, and G. Martin, EDA for IC Implementation, Circuit Design, and Process Technology. Boca Raton: CRC Press, 2006.

[4] 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.

[5] A. B. Kahng, "Machine Learning Applications in Physical Design: Recent Results and Directions," IEEE Design Test, 2023. doi: 10.1109/MDAT.2023.3237076.

[6] A. Mirhoseini et al., "A graph placement methodology for fast chip design," Nature, vol. 594, no. 7862, pp. 207–212, 2021. doi: 10.1038/s41586-021-03544-w.

[7] Synopsys, "Advanced On-Chip Variation (AOCV) Timing Signoff," White Paper, 2022.

[8] Cadence Design Systems, "ARM Neoverse V2 Implementation Using Cadence Digital Full-Flow," Press Release, 2023.

[9] R. Liang et al., "DRC Hotspot Prediction at Sub-10nm Process Nodes Using Customized Convolutional Network," in Proc. Int. Symp. Physical Design (ISPD), pp. 135–142, 2020. doi: 10.1145/3372780.3375560.

[10] E. C. Barboza, N. Shukla, Y. Chen, and J. Hu, "Machine Learning-Based Pre-Route Timing Estimation with Reduced Pessimism," in Proc. ACM/IEEE Design Autom. Conf. (DAC), 2019. doi: 10.1145/3316781.3317857.

[11] A. F. Budak, M. Gandara, W. Shi, D. Z. Pan, N. Sun, and B. Liu, "An Efficient Analog Circuit Sizing Method Based on Machine Learning Assisted Global Optimization," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 41, no. 5, pp.

1209–1221, 2022. doi: 10.1109/TCAD.2021.3081405.

[12] V. A. Chhabria, W. Jiang, A. B. Kahng, and S. S. Sapatnekar, "Improving Timing Consistency Between Global and Detailed Routing Using Machine Learning," ACM Trans. Design Autom. Electron. Syst., 2023. doi: 10.1145/3626959.

[13] X. Gao et al., "Timing-Driven Placement Using ML-Based Wirelength Prediction," in Proc. Int. Symp. Physical Design (ISPD), 2022. doi: 10.1145/3505170.3506722.

[14] Z. Guo, M. Liu, J. Gu, S. Zhang, D. Z. Pan, and Y. Lin, "A Timing Engine Inspired Graph Neural Network Model for Pre-Routing Slack Prediction," in Proc. ACM/IEEE Design Autom. Conf. (DAC), 2022. doi: 10.1145/3489517.3530597.

[15] R. Liang, Z. Xie, J. Jung, V. Chauha, Y. Chen, J. Hu, H. Xiang, and G.-J. Nam, "Routing-Free Crosstalk Prediction," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), 2020. doi: 10.1145/3400302.3415712.

[16] 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 Keynote Paper," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 41, no. 10, pp. 3162–3181, 2022. doi: 10.1109/TCAD.2021.3124762.

[17] Z. Xie, H. Li, X. Xu, J. Hu, and Y. Chen, "PowerNet: Transferable Dynamic IR-Drop Estimation via Maximum Convolutional Neural Network," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), 2020. doi: 10.1145/3400302.3415763.

[18] Z. Zhao, S. Zhang, G. Liu, C. Feng, T. Yang, A. Han, and L. Wang, "Machine-Learning-Based Multi-Corner Timing Prediction for Faster Timing Closure," Electronics, vol. 11, no. 10, p. 1571, 2022. doi: 10.3390/electronics11101571.

[19] W. Ding et al., "[Title]," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), 2024. doi: 10.1145/3676536.3676772.

[20] Z.-G. Yu, X.-Y. Zhong, X.-J. Ma, and X.-F. Gu, "[Title]," Integration, VLSI J., vol. 95, p. 102109, 2023. doi: 10.1016/j.vlsi.2023.102109.