CAPCell: Standard Cell Layout Synthesis with Parasitic Capacitance Aware Parallel Sampling

Kairong Guo[1], Jiechen Huang[2], Wenjian Yu[2], Yibo Lin[1,3,4*]
[1]School of Integrated Circuits, Peking University, Beijing
[2]Department of Computer Science and Technology, BNRist, Tsinghua University, Beijing
[3]Institute of Electronic Design Automation, Peking University, Wuxi
[4]Beijing Advanced Innovation Center for Integrated Circuits
krguo25@stu.pku.edu.cn, hjc22@mails.tsinghua.edu.cn, yu-wj@tsinghua.edu.cn, yibolin@pku.edu.cn

***Abstract*—As technology nodes scale into the angstrom era, the correlation between geometric heuristics (e.g., total wirelength) and actual circuit performance has significantly degraded. Consequently, traditional standard cell synthesis frameworks, which rely heavily on geometric proxies, fail to capture intricate parasitic effects, resulting in sub-optimal Power, Performance, and Area (PPA). To bridge this gap, we propose CAPCell, a neuro-symbolic framework that directly targets parasitic capacitance optimization during layout synthesis. CAPCell integrates a pre-trained CNN as a high-fidelity surrogate for expensive field solvers and employs a hybrid solver-guided sampling strategy to navigate the routing search space. This approach allows for the rigorous optimization of electrical metrics while guaranteeing strict design rule compliance (DRC-clean). Validated on the ASAP7 7nm technology node, CAPCell demonstrates the capability to break the limitations of geometric heuristics. Experimental results show that our method achieves an average delay reduction of 0.36 ps (0.2%) per cell under strict iso-area conditions. Crucially, these cell-level improvements translate to the system level, yielding 2% improvement in maximum operating frequency and 7% reduction in TNS for block-level designs.**

***Index Terms*—standard cell, layout synthesis, transistor-level placement and routing**

I. Introduction

Standard cells constitute the atomic foundation of modern Very Large Scale Integration (VLSI) designs. As technology nodes scale into the angstrom era, the complexity of cell design has exploded due to intricate design rules. Consequently, the industry has largely transitioned from manual layout design to automated standard cell layout synthesis [1] [2] [3]. The core objective of this process is to generate standard cell layouts that are not only DRC-clean but also optimized for Power, Performance, and Area (PPA).

The increasing decoupling of geometric heuristics from standard cell layout performance renders existing synthesis methodologies ineffective. This discrepancy is explicitly demonstrated in Figure 1, where layout variants with identical wirelengths exhibit vastly different performance metrics. Currently, dominant synthesis frameworks—ranging from classic SAT/SMT approaches [4] [5] [6] [7] to recent heuristic search-based methods—rely heavily on geometric proxies to guide the search space. While these methods excel at enforcing complex design rules, they fundamentally struggle to optimize

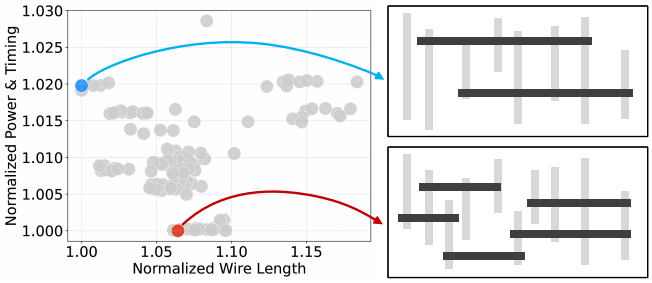

Fig. 1: The decoupling of geometric heuristics from standard cell layout performance. The scatter plot shows 120 layout variants of A2O1A1O1Ixp33, plotting normalized wire length against normalized weighted sum of power and timing.

electrical objectives directly. Even recent works explicitly targeting timing, such as [8], still operate within the geometric paradigm by prioritizing the wirelength reduction of critical paths. However, this specific approach incurs additional area penalties. Furthermore, relying on such geometric heuristics leaves the solver blind to the intricate parasitic effects that dominate in advanced nodes.

To bridge this gap, selecting an appropriate optimization objective is critical. While end-to-end performance metrics—such as propagation delay and dynamic power—are the ultimate targets, directly optimizing them within the search loop faces two major hurdles. First, rigorous evaluation necessitates expensive parasitic extraction (PEX) and SPICE simulations, which are computationally prohibitive for iterative search. Second, although data-driven approaches for direct layout-to-performance prediction have been explored [9], their generalization capability remains questionable due to the inherent difficulty of the task. In this context, parasitic capacitance serves as the ideal high-fidelity surrogate. The validity of this objective is corroborated by recent advancements such as [3], which implicitly targets parasitic reduction by tuning soft constraint weights for coupling nets in a MaxSAT framework. It acts as the physical root cause of dynamic power consumption and signal delay, yet unlike timing metrics, it is structurally determined by the layout geometry. This reduces the prediction task to a pure geometric pattern recognition problem—a domain where machine learning has demonstrated

*Corresponding author.

robust generalization and extensive success [10] [11]. This strategy also resonates with manual layout practices, where experienced designers intuitively minimize parasitic coupling to facilitate timing closure. Consequently, by targeting parasitic capacitance, we may effectively optimize the physical foundations of PPA without incurring the excessive cost of full-flow analysis or the generalization risks of black-box performance models.

The fundamental challenge of standard cell layout synthesis lies in its nature as a combinatorial optimization problem under strict design constraints, coupled with highly non-linear performance objectives. SAT/SMT solvers operate effectively in a discrete boolean space designed to satisfy hard feasibility constraints, but they lack the mathematical machinery to handle continuous, differentiable electrical costs. Given that strict adherence to design rules is non-negotiable, strictly abandoning these solvers is infeasible. Instead, a more rational strategy is to adopt a hierarchical framework: maintaining the solver as a low-level engine for valid constraint enforcement, while employing a high-level external policy to guide the discrete search towards electrically superior regions.

To address these challenges, we propose **CAPCell**, a capacitance-driven framework for standard cell layout synthesis and optimization.

The main contributions of this paper are summarized as follows:

- We propose CAPCell, a neuro-symbolic framework that directly targets parasitic capacitance optimization during standard cell layout synthesis, which moves beyond geometric proxies.
- We introduce a hybrid optimization strategy that combines a SAT solver for hard constraint enforcement with a neural policy for soft performance guidance, ensuring DRC-clean while enabling the optimization of complex, non-linear electrical metrics.
- We demonstrate the effectiveness of CAPCell on the ASAP7 7nm predictive PDK. Experimental results indicate that our framework achieves a consistent delay reduction of 0.36 ps (0.2%) per cell while strictly maintaining the minimum area. Furthermore, block-level evaluations confirm that these cumulative cell-level gains translate into 2% improvement in maximum operating frequency and 7% reduction in TNS, compared to standard wirelength-driven libraries.

The rest of the paper is organized as follows, Section II introduces the background and problem formulation; Section III explains the details of the proposed flow; Section IV validates the flow with experimental results; Section V concludes the paper.

II. Preliminaries

A. SAT Formulation for Multi-Commodity Flow based Routing

Standard cell layout synthesis proceeds from placement to routing. Due to limited routing tracks and complex design rules in advanced nodes, SAT-based routing is widely adopted to strictly ensure DRC-clean.

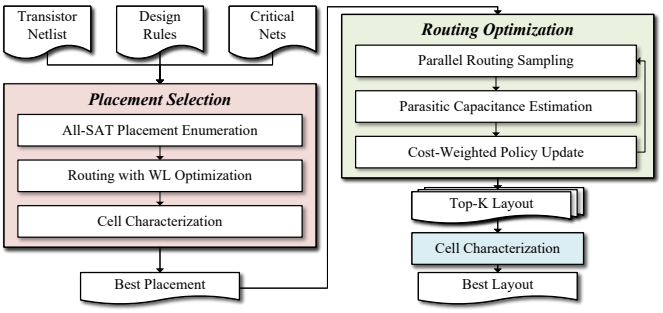

Fig. 2: Proposed parasitic-capacitance-driven standard cell layout synthesis flow.

The routing problem is typically formulated as a Multi-Commodity Flow (MCF) problem on a 3-D grid graph $G(V, E)$. This formulation relies on a hierarchical variable structure involving **flow**, **net**, and **metal** indicators. Connectivity for each net n is maintained by commodity flow variables $f_m^n(u, v)$. To ensure a continuous path from source to sink, the commodity flow conservation constraint is applied to every vertex:

$$\sum_{u \in adj(v)} f_{in}^n(u, v) = \sum_{w \in adj(v)} f_{out}^n(v, w) \tag{1}$$

Flow variables are mapped to net usage indicators $e_{u,v}^n$, and an exclusive net usage constraint is enforced via an At-Most-One (AMO) formulation:

$$\sum_{n \in N} e_{u,v}^n \leq 1 \tag{2}$$

This constraint implies that for any routing resource (each edge), the solver must select one state from $N+1$ possibilities: the edge is either assigned to one specific net n, or it remains unused (0). Finally, the physical metal presence is determined by the union of net usage variables:

$$m_{u,v} = \bigvee_{n \in N} e_{u,v}^n. \tag{3}$$

Complex design rules are formulated directly on these metal variables to ensure DRC-clean.

B. Problem Formulation

Given the transistor-level netlist of a standard cell, design rules, and a specified set of critical nets $\mathcal{N}_{crit}$ targeted for parasitic optimization, the goal is to generate a DRC-clean and parasitic capacitance optimized standard cell layout.

III. Algorithm

A. Overview of CAPCell Flow

The overall architecture of the proposed CAPCell framework is illustrated in Figure 2. The workflow proceeds in two sequential stages: placement selection and routing optimization. First, we generate a set of valid transistor placements using an All-SAT solver. For each placement candidate, we generate a corresponding layout using a wirelength-driven

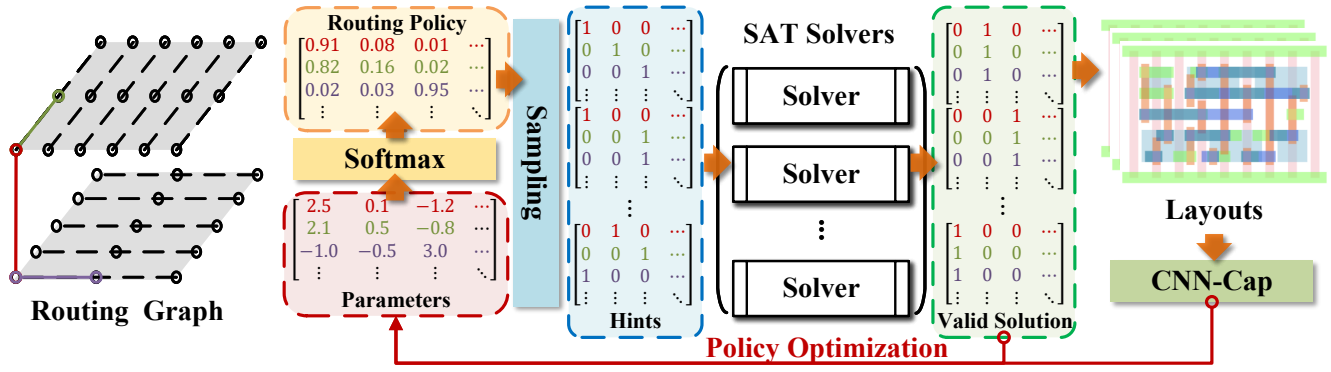

Fig. 3: Detailed workflow of the parasitic-capacitance-driven routing optimization loop.

router and perform standard cell characterization to select the best placement. Second, with the placement fixed, we employ our proposed neuro-symbolic routing engine to iteratively optimize the routing topology for parasitic capacitance.

1) All-SAT based Placement Enumeration and Selection: In the first phase, the goal is to determine a fixed transistor placement. Starting with the netlist and design rules, we use an All-SAT solver to enumerate a subset of valid placement solutions. To enhance solution quality, we incorporate the routability-driven pruning and symmetry breaking constraints proposed in [12], which effectively eliminate redundant isomorphic solutions. Furthermore, the solver is constrained to a fixed boundary, ensuring all enumerated placements achieve the minimum possible area. To evaluate these candidates, we perform routing with a wirelength minimization objective and run the complete cell characterization flow (PEX + SPICE) on these layouts. The placement with the best PPA metrics is selected for the subsequent optimization phase.

2) Phase 2: Capacitance-Driven Routing Optimization: The optimization loop consists of three synchronized modules:

- Parallel Routing Sampling: A neural policy π_θ generates probabilistic routing hints, which are subsequently legalized by a SAT solver into DRC-clean routing solutions.
- Parasitic Capacitance Estimation: A pre-trained CNN predicts parasitic capacitance as a high-fidelity surrogate for expensive field solvers (PEX), enabling rapid evaluation.
- Cost-Weighted Policy Update: The policy is iteratively updated via self-imitation, selecting "elite" samples with lowest capacitance to guide the policy gradient.

Finally, the Top-K optimized layouts undergo rigorous cell characterization to identify the final best layout. The notations are provided in Table I.

B. Solver-Guided Parallel Sampling via User-Propagator

This section details how CAPCell bridges the continuous probability space of the neural network and the discrete boolean space of the SAT solver using the **User-Propagator** mechanism. We formulate the routing problem on a grid graph $G = (V, E)$. As shown in the left panel of Figure 3, for each edge $e \in E$, the neural policy π_θ predicts a categorical

TABLE I: Notations used in CAPCell Formulation

Category	Description
Problem Setup	
$G(V, E)$	Grid graph with vertices V and edges E
$\mathcal{N}, \mathcal{C}$	Set of nets and routing candidates
$\mathcal{N}_{crit}$	Subset of critical nets targeted for optimization
Neural Policy	
π_θ	Neural routing policy parameterized by θ
$P_\theta(e)$	Probability distribution vector for edge e
$\mathcal{H}(\pi_\theta)$	Entropy regularization term
Solver & Sampling	
$\mathcal{H}$	Set of probabilistic routing hints
f_{SAT}	The SAT solver function (Repair: $H \to \tau_{sat}$)
τ_{sat}	The **solver-repaired** valid routing topology
Optimization	
$C(\tau)$	Normalized parasitic capacitance cost
C_{best}	Global minimum capacitance observed (for Phase I)
T	Temperature parameter for Boltzmann weighting
$A(\tau)$	Advantage function (for Phase II)

probability distribution over the set of possible candidates $\mathcal{C} = \mathcal{N} \cup \{\emptyset\}$ (where $\emptyset$ denotes no net usage). Formally, let $P_\theta(e) \in [0, 1]^{|\mathcal{C}|}$ be the output probability vector computed via a **softmax** function, satisfying $\sum_{c \in \mathcal{C}} P_\theta(e, c) = 1$.

To efficiently guide the SAT solver, we employ a **Pre-sampled Hint Injection** strategy. The process operates as follows:

1) Parallel Hint Generation and Injection: To facilitate high-throughput exploration, we instantiate K independent SAT solver threads running in parallel. We first perform parallel sampling from the neural distribution to generate a batch of discrete hints $\mathcal{H} = \{H^{(1)}, \ldots, H^{(K)}\}$, assigning each $H^{(k)}$ to a specific solver instance. Since the candidates for each edge are mutually exclusive, we apply **Roulette Wheel Selection** (Categorical sampling) for each edge e to select a single specific candidate $c_e^{(k)}$:

$$c_e^{(k)} \sim \text{Categorical}(P_\theta(e)) \tag{4}$$

The hint set $H^{(k)}$ is thus constructed as a collection of these selected candidates, providing a unique geometric skeleton that guides the routing search of the k-th parallel solver.

2) Intervention via Decide Callback: We register a custom `Decide` callback within the Z3 solver. When the solver requests a branching decision for a boolean variable $x_{e,n}$

(indicating if net n uses edge e), the callback prioritizes the pre-sampled choice:

$$\text{Decide}(x_{e,n}) := \begin{cases} \text{True} & \text{if } n = c_e^{(k)} \\ \text{False} & \text{otherwise} \end{cases} \tag{5}$$

This mechanism overrides the solver's native heuristics, effectively steering the boolean search towards the global topology predicted by the neural network. Importantly, the hints are not added as hard constraints; if a hinted branch leads to conflict, Z3's CDCL backtracking and learned clauses can override the hinted assignment, enabling conflict-driven repair.

3) Conflict-Driven Repair: Critically, these hints are treated as soft preferences. If a neural hint violates strict design rules (e.g., causing a short circuit or spacing violation), the solver's native propagation logic automatically overrides the hint to resolve the conflict. This guarantees that every generated sample $\tau^{(k)}$ is a valid, **DRC-clean** layout.

The detailed mechanism for updating the policy parameters θ based on the quality of these sampled layouts will be elaborated in Section III-D.

C. CNN-Based Capacitance Estimation

Accurate evaluation of parasitic capacitance is pivotal for performance-driven routing. While field solvers provide golden-standard accuracy, their computational cost is prohibitive for the inner loop of a routing engine. To bridge this gap, we employ a high-fidelity surrogate model inspired by **CNN-Cap** [10].

1) Surrogate Model Architecture: We adopt a deep convolutional neural network (CNN) based on the ResNet architecture to capture the complex spatial coupling effects in high-density routing. Consistent with [10] [13], we utilize a **grid-based data representation**, where the routing cross-sections are encoded as 2D multi-channel density maps. Distinct from prior metal-only inputs, each metal and via layer is encoded as an individual feature-map channel. Given the compact nature of standard cells, vias contribute significantly to the coupling capacitance; this inclusive representation better captures inter-layer coupling effects in dense layouts. This allows the model to handle a variable number of conductors and effectively learn the non-linear relationship between geometric topology and electrostatic field distribution. Unlike the original component-wise extraction, our framework aggregates the predictions to estimate the total parasitic capacitance for each critical net directly. This end-to-end prediction enables rapid feedback for the routing policy.

2) Normalized Cost Formulation: To ensure the optimization objective is balanced across nets with varying lengths and varying absolute capacitance magnitudes, we define the cost metric $C(\tau)$ as a normalized ratio against the baseline (best placement layout in Section III-A). Let τ_{base} be the initial routing topology generated by a wirelength-driven router, and let $C_{base}(n)$ be the extracted capacitance of net n in this baseline layout. For a generated layout τ, the global cost $C(\tau)$ is calculated as the mean improvement ratio over the set of critical nets $\mathcal{N}_{crit}$:

$$C(\tau) = \frac{1}{|\mathcal{N}_{crit}|} \sum_{n \in \mathcal{N}_{crit}} \frac{\hat{C}_n(\tau)}{C_{base}(n)} \tag{6}$$

where $\hat{C}_n(\tau)$ is the CNN-predicted total capacitance of net n in layout τ. This normalization ensures that the policy optimization is driven by relative performance gain rather than raw geometric metrics.

D. Cost-Weighted Policy Optimization

To effectively navigate the combinatorial search space, we propose a two-phase **Cost-Weighted Policy Optimization** algorithm. This approach addresses the "cold start" problem while ensuring continuous performance improvement through a transition from weighted imitation to advantage-based reinforcement.

1) Optimization Objective: Our ultimate goal is to minimize the expected parasitic capacitance $C(\tau)$ over the distribution of valid layouts generated by the solver-guided policy. Crucially, the policy update is **not** based on the initial probabilistic hints H generated by the network, but on the valid solutions repaired by the solver. We treat the **solver-repaired solution** $\tau_{sat} = f_{SAT}(H)$ as the ground-truth label (pseudo-label) for the policy to imitate. We employ a gradient-based strategy by minimizing the following surrogate loss function:

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{B}|} \sum_{(\tau,c) \in \mathcal{B}} w_k \sum_{e \in E} \log P_\theta(e, \tau_e) - \lambda \mathcal{H}(\pi_\theta) \tag{7}$$

where $\mathcal{B}$ is the batch of valid samples collected from parallel solvers (specifically the subset of repaired solutions τ_{sat}), τ_e denotes the net chosen for edge e in the repaired layout, w_k is a phase-dependent weight, and $\mathcal{H}(\pi_\theta)$ is the entropy regularization term.

2) Phased Training Strategy: We dynamically adjust the weighting mechanism w_k across two phases:

Phase I: Cost-Weighted Imitation (Warmup). In the initial epochs ($t < T_{warm}$), the policy lacks the knowledge to generate valid routing patterns consistently. To bootstrap the learning, we employ a **Self-Imitation** strategy. Instead of treating all valid solver outputs equally, we prioritize those with lower capacitance. We filter the batch for the top-K "elite" samples and assign weights using a Boltzmann distribution based on their cost gap relative to the global best solution C_{best}:

$$w_k^{(I)} = \exp\left(-\frac{C(\tau_k) - C_{best}}{T}\right) \tag{8}$$

This ensures that even during the warmup phase, the policy is biased towards imitating electrically superior topologies rather than merely valid ones.

Phase II: Advantage-Weighted Optimization. Once the policy stabilizes ($t \geq T_{warm}$), we switch to a more aggressive **Advantage-based** update to escape local optima. We utilize a standardized advantage function:

$$w_k^{(II)} = A(\tau_k) = \frac{\bar{C} - C(\tau_k)}{\sigma_C + \epsilon} \tag{9}$$

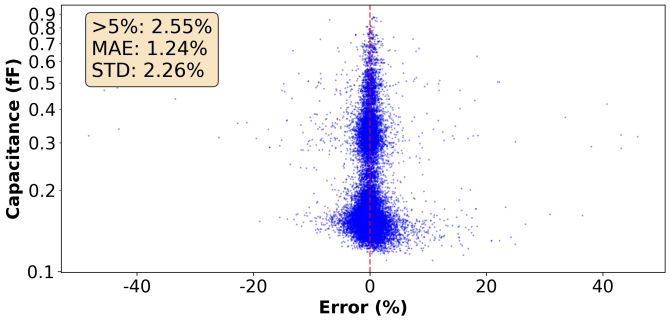

Fig. 4: Error of the capacitance estimator on the testing set. The x-axis represents the relative error percentage, and the y-axis represents the ground truth capacitance (in fF, log scale). The model achieves an MAE of 1.24%, with only 2.55% of samples exceeding 5% error.

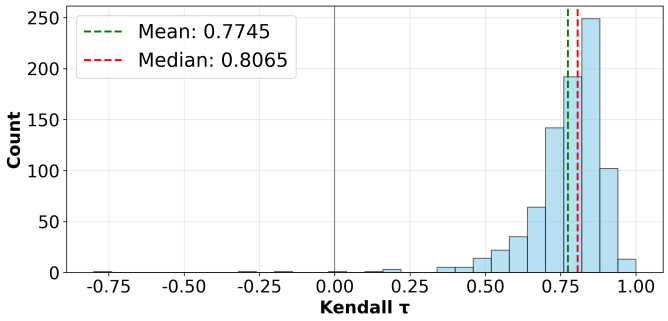

Fig. 5: Histogram of Kendall's Rank Correlation Coefficient (τ) calculated for each cell-net pair across layout variants. The distribution shows a high correlation with a mean of 0.77 (green dashed line) and a median of 0.81 (red dashed line), demonstrating the model's effectiveness in ranking topological quality.

- **Reward** ($w > 0$)**:** Samples better than the batch average ($\bar{C}$) are reinforced.
- **Penalty** ($w < 0$)**:** Crucially, samples worse than average are assigned negative weights. This explicitly suppresses the probability of generating inferior structures, a capability absent in the imitation phase.

This two-phase mechanism creates a robust curriculum: Phase I rapidly establishes a high-quality baseline distribution via weighted imitation of solver-repaired solutions, while Phase II refines this distribution by explicitly penalizing sub-optimal explorations.

IV. Experimental Results

Our proposed CAPCell framework is implemented in C++ and Python. The layout synthesis and solver interactions are handled in C++, while the neural policy and capacitance estimation networks are implemented using PyTorch. The system is executed on a workstation equipped with an AMD EPYC 9654 CPU and NVIDIA RTX 5090 GPUs. We employ the Z3 Solver [14] (v4.8.5) for both SAT-based placement and routing. We evaluate our framework using the **ASAP7** [15] 7nm predictive PDK. Parasitic extraction is performed using *Calibre xACT*, followed by cell characterization using *Cadence Liberate*.

A. Capacitance Estimation Precision

The estimator was trained on a dataset covering **151** ASAP7 standard cells. We generated **22,075** layout variants, resulting in **122,450** net-level data points. For each cell, its layout variants were split 8:2 into training and testing sets to evaluate generalization to unseen routing topologies.

Figure 4 validates the model's absolute precision. The Mean Absolute Percentage Error (MAE) is restricted to **1.24%**, with a mere **2.55%** of outliers exceeding 5% error. This tight error distribution confirms that the model robustly captures parasitic effects across different scales.

More critically for optimization guidance, Figure 5 demonstrates the model's ranking fidelity. The Kendall's τ achieves a mean of **0.77** and a median of **0.81**, indicating a strong correlation in quality ordering. This ensures the surrogate model can effectively distinguish superior topologies from inferior ones to drive the policy gradient.

TABLE II: Comparison of post-layout performance metrics between the Baseline (wirelength-driven) and CAPCell (capacitance-driven) implementations. All values are normalized to the Baseline. Delay and Power metrics are computed by averaging the values across all valid timing arcs and slews/load conditions in the characterized Liberty (.lib) files. Note that Area remains identical for both methods as the placement is fixed.

Cell Type	Baseline (WL opt.)			CAPCell			
	WL	Power	Delay(ps)	WL	Power	Delay(ps)	Delay_impr.(ps)
AND2x2	1.00	1.00	58.304	1.08	1.01	58.272	0.032
AND3x1	1.00	1.00	495.692	1.02	1.00	495.456	0.236
AND3x2	1.00	1.00	58.59	1.07	0.99	58.257	0.333
AOI21xp5	1.00	1.00	111.735	1.01	0.98	111.608	0.127
AOI22xp5	1.00	1.00	226.782	1.05	1.00	226.671	0.111
BUFx2	1.00	1.00	267.526	1.08	1.00	267.342	0.184
BUFx3	1.00	1.00	183.589	1.01	0.99	183.408	0.181
BUFx4	1.00	1.00	88.173	1.04	0.99	87.722	0.451
BUFx8	1.00	1.00	34.276	1.01	0.99	33.930	0.346
FAxp33	1.00	1.00	1327.36	1.04	1.00	1323.638	3.722
INVx1	1.00	1.00	78.374	1.03	1.02	78.365	0.009
INVx2	1.00	1.00	258.749	1.00	1.01	258.687	0.062
INVx4	1.00	1.00	78.352	1.04	1.01	78.238	0.114
INVx8	1.00	1.00	19.704	1.12	1.00	19.594	0.110
NAND2x1	1.00	1.00	461.953	1.00	0.99	461.785	0.168
NAND2x2	1.00	1.00	44.964	1.04	1.00	44.839	0.125
NAND3x1	1.00	1.00	454.685	1.01	0.99	454.556	0.129
NAND3x2	1.00	1.00	47.804	1.07	0.99	47.199	0.605
NOR2x1	1.00	1.00	481.331	1.01	0.97	481.144	0.187
NOR2x2	1.00	1.00	46.646	1.00	1.00	46.591	0.055
NOR3x1	1.00	1.00	484.803	1.01	1.00	484.010	0.793
NOR3x2	1.00	1.00	49.279	1.00	0.98	49.195	0.084
OAI21xp5	1.00	1.00	105.588	1.02	0.99	105.454	0.134
OAI22xp5	1.00	1.00	124.986	1.08	0.98	124.777	0.209
OR2x2	1.00	1.00	58.816	1.00	1.00	58.701	0.115
OR3x1	1.00	1.00	497.847	1.00	0.97	496.694	1.153
OR3x2	1.00	1.00	60.457	1.04	0.98	59.937	0.520
XNOR2xp5	1.00	1.00	390.753	1.00	1.00	390.739	0.014
XOR2xp5	1.00	1.00	409.163	1.00	0.99	408.922	0.241
Norm./Avg.	1.00	1.00	1.00	1.03	0.99	**0.998**	0.364

B. Cell-level Performance Improvement

We applied CAPCell to optimize a suite of fundamental standard cells. For the optimization configuration, we set the parallel sampling width $K = 64$ per iteration, with a warmup phase of 10 epochs ($T_{warm} = 10$). The set of critical nets $\mathcal{N}_{crit}$ was defined to include all I/O pins and internal cascaded nets for multi-stage logic. More generally, $\mathcal{N}_{crit}$ can be specified by users (e.g., from chip-level timing analysis) to focus optimization on selected nets.

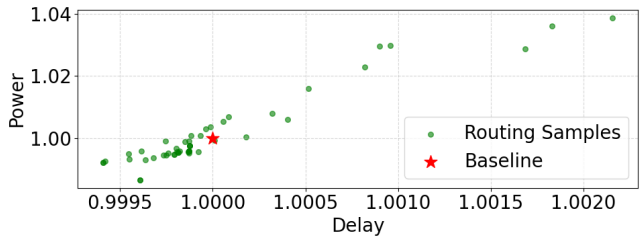

Fig. 6: Power-Delay distribution of the top-50 layout candidates for the XOR2xp5 cell. The red star represents the baseline (wirelength-driven) performance. The green dots indicate CAPCell outputs, with the cluster in the bottom-left quadrant highlighting Pareto-superior solutions that achieve both lower delay and reduced power.

We benchmark the performance against the best layout selected during the placement enumeration phase. Upon completion of the optimization loop, the top 50 candidate layouts are selected for final characterization (PEX + SPICE), and the best result is reported.

Table II summarizes the comparison results. With the transistor placement fixed, CAPCell achieves consistent timing improvements within the **identical cell area**. Although the total wirelength increases slightly (average **1.03**×) as the router detours nets to avoid coupling, this strategy effectively translates to reduced signal delay. Notably, complex cells like FAxp33 and OR3x1 show significant delay reductions of **3.72 ps** and **1.15 ps**, respectively.

Regarding power, while reduced coupling capacitance can lower dynamic switching power, detoured routes may increase total wire capacitance and degrade slew, which can offset these benefits. However, the average power remains comparable to the baseline (**0.99**×). While the average delay improvement per cell (∼**0.36 ps**, representing a **0.2%** reduction) might seem minor individually, these gains accumulate along critical paths. The impact of this cumulative optimization on full-chip timing is evaluated in the following block-level analysis.

To visualize the optimization, Figure 6 plots the top-50 candidates for XOR2xp5. While minor outliers exist due to the inherent gap between the capacitance proxy and final PPA metrics, the distribution predominantly favors the lower-left quadrant. This confirms that CAPCell successfully identifies Pareto-superior layouts that outperform the baseline.

C. Block-level Improvement

To validate the propagation of cell-level gains to the block-level, we performed physical implementation using *Cadence Innovus* on identical gate-level netlists.

We constructed two distinct library configurations for comparative analysis:

- **Baseline Library:** Composed of combinational cells generated by the wirelength-driven baseline router.
- **Optimized Library:** Composed of combinational cells optimized by CAPCell.

Note that for sequential elements, both configurations utilize the original flip-flops (DFFs) from the ASAP7 PDK to strictly isolate the gains derived from combinational logic optimization.

TABLE III: Comparison of block-level performance metrics implemented with Baseline and CAPCell libraries. **Freq.** (Maximum Frequency) is reported in GHz, calculated as $1/(T_{target} - \text{WNS})$, where $T_{target} = 0.2ns$. **WNS** and **TNS** are reported in ns.

Design	#Cells	Baseline			CAPCell		
		freq.	WNS	TNS	freq.	WNS	TNS
gcd	601	3.11	-0.122	-3.343	3.16	-0.116	-3.552
uart	943	4.69	-0.013	-0.388	4.74	-0.011	-0.224
riscv	13964	1.12	-0.689	-698.834	1.15	-0.673	-696.462
aes	20047	2.46	-0.206	-51.183	2.51	-0.198	-50.462
jpeg	135536	2.21	-0.251	-428.960	2.23	-0.249	-433.343
Norm./Avg.		1.00	1.00	1.00	**1.02**	**0.95**	**0.93**

To rigorously stress-test the timing closure capabilities and magnify the performance differences between the two libraries, we imposed an aggressive clock period constraint of **0.2 ns** during the placement and routing stages. The target placement utilization was set to **0.6**. Table III summarizes the block-level timing metrics. While the individual cell-level delay reduction (∼0.2%) appears marginal, these gains successfully accumulate along critical paths to yield tangible block-level improvements. On average, the CAPCell-optimized designs achieve a 2% increase in maximum operating frequency, and 7% reduction in TNS, confirming that our capacitance-driven strategy effectively alleviates timing bottlenecks across the entire design.

V. Conclusion

In this paper, we addressed the critical decoupling between geometric heuristics and electrical performance in advanced standard cell synthesis. We introduced CAPCell, a capacitance-driven routing framework that leverages a neuro-symbolic architecture to optimize parasitic effects directly. By employing a CNN-based surrogate model within a SAT-guided search loop, our approach effectively identifies routing topologies that minimize coupling capacitance—a dominant performance detractor that traditional wirelength-driven solvers often overlook. Experimental results on the ASAP7 7nm PDK demonstrate that CAPCell successfully uncovers Pareto-superior solutions. We achieved a consistent delay reduction of 0.36 ps (0.2%) across benchmark cells without incurring any area penalty. Furthermore, the block-level validation confirms the scalability of these gains, resulting in 2% improvement in maximum operating frequency and 7% reduction in TNS. These findings validate that shifting the optimization objective from pure geometry to physical parasitics is essential for unlocking the full PPA potential of standard cells in the post-FinFET era.

Acknowledgment

This project is supported in part by the MIND project (MINDXZ202406) and the Natural Science Foundation of Beijing, China (Grant No. Z230002).

REFERENCES

[1] C.-T. Ho, A. Ho, M. Fojtik, M. Kim, S. Wei, Y. Li, B. Khailany, and H. Ren, "NVCell 2: Routability-Driven Standard Cell Layout in Advanced Nodes with Lattice Graph Routability Model," in *Proceedings of the 2023 International Symposium on Physical Design*. Virtual Event USA: ACM, 2023, pp. 44–52.

[2] C.-T. Ho, A. Chandna, D. Guan, A. Ho, M. Kim, Y. Li, and H. Ren, "Novel transformer model based clustering method for standard cell design automation," in *Proceedings of the 2024 International Symposium on Physical Design*, 2024, pp. 195–203.

[3] J.-C. Tsai, W.-M. Hsu, Y.-T. Hsieh, Y.-J. Li, W. Huang, C. Ho, J.-H. Yang, and H.-L. Huang, "MAXCell: PPA-Directed Multi-Height Cell Layout Routing Optimization using Anytime MaXSAT with Constraint Learning," *New York*, 2024.

[4] D. Park, I. Kang, Y. Kim, S. Gao, B. Lin, and C.-K. Cheng, "ROAD: Routability Analysis and Diagnosis Framework Based on SAT Techniques," in *Proceedings of the 2019 International Symposium on Physical Design*. San Francisco CA USA: ACM, 2019, pp. 65–72.

[5] D. Lee, D. Park, C.-T. Ho, I. Kang, H. Kim, S. Gao, B. Lin, and C.-K. Cheng, "SP&R: SMT-Based Simultaneous Place-and-Route for Standard Cell Synthesis of Advanced Nodes," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 10, pp. 2142–2155, 2021.

[6] C.-K. Cheng, C.-T. Ho, D. Lee, B. Lin, and D. Park, "Complementary-fet (cfet) standard cell synthesis framework for design and system technology co-optimization using smt," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 6, pp. 1178–1191, 2021.

[7] K. Guo, H. Lu, R. Guo, J. Wang, C. Zhao, H. Wu, R. Wang, and Y. Lin, "Standard cell layout synthesis for dual-sided 3d-stacked transistors," in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2026.

[8] S. Chung, H. Seo, and T. Kim, "Synthesis of standard cells of minimum delay," in *2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 2025, pp. 1–8.

[9] Z. Luo, L. Zhou, Z. Zhang, J. Zhou, Z. Li, H. You, F. Yao, and Z. Zhao, "Protocelllayout: Prototype-guided graph learning for accurate and generalizable standard cell layout ppa estimation," in *Proceedings of the 2025 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2025, pp. 1–9.

[10] D. Yang, W. Yu, Y. Guo, and W. Liang, "Cnn-cap: Effective convolutional neural network based capacitance models for full-chip parasitic extraction," in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 2021, pp. 1–9.

[11] J.-C. Tsai, H.-M. Huang, W.-M. Hsu, P.-T. Lee, J.-H. Yang, H.-L. Huang, Y.-J. Su, and C. H.-P. Wen, "Rescap: Fast-yet-accurate capacitance extraction for standard cell design by physics-guided machine learning," in *Proceedings of the 30th Asia and South Pacific Design Automation Conference*, 2025, pp. 1243–1250.

[12] Y.-J. Su, J.-C. Tsai, H.-M. Huang, A. C.-W. Liang, H.-Y. Tsai, W.-M. Hsu, J.-H. Yang, and C. H.-P. Wen, "Cop&r: Co-optimizing place-and-route for standard cell layout via mcts and allsat," in *2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 2025, pp. 1–9.

[13] D. Yang, H. Li, W. Yu, Y. Guo, and W. Liang, "Cnn-cap: Effective convolutional neural network-based capacitance models for interconnect capacitance extraction," *ACM Transactions on Design Automation of Electronic Systems*, vol. 28, no. 4, pp. 1–22, 2023.

[14] L. De Moura and N. Bjørner, "Z3: An efficient smt solver," in *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2008, pp. 337–340.

[15] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, "Asap7: A 7-nm finfet predictive process design kit," *Microelectronics Journal*, vol. 53, pp. 105–115, 2016.