EPiCell: Electro-Physical Co-Modeling for Standard Cell PPA Prediction

Wenbo An[1,2 †], Kairong Guo[1,†], Haoyi Zhang[1], Yibo Lin[1,3,4*]
[1]School of Integrated Circuits, Peking University, Beijing
[2]Southern University of Science and Technology, Shenzhen
[3]Institute of Electronic Design Automation, Peking University, Wuxi
[4]Beijing Advanced Innovation Center for Integrated Circuits
anwb2023@mail.sustech.edu.cn, {krguo25, hy.zhang}@stu.pku.edu.cn, yibolin@pku.edu.cn

ABSTRACT

In advanced process nodes, the pursuit of extreme PPA optimization has driven an explosion in the demand for customized standard cells. To satisfy this demand, automated layout synthesis has been increasingly adopted to explore vast design spaces. However, this paradigm shifts the bottleneck from design creation to verification, as characterizing the massive volume of generated variants via SPICE is computationally prohibitive. Meanwhile, conventional geometric heuristics fail to proxy PPA at advanced nodes due to dominant layout effects. Existing learning-based surrogates often lack the fidelity to capture these complex dependencies. To bridge this gap, we propose EPiCell, an electro-physical co-modeling framework for rapid PPA estimation. EPiCell features a Heterogeneous Graph Transformer (HGT) that explicitly models transistors, routing metals, and supply rails as distinct entities, unifying circuit topology with fine-grained layout geometry. By employing relation-aware attention, it effectively captures the nonlocal electro-physical interactions governing cell performance. Validated on a dataset of over 18,000 auto-generated layouts based on ASAP7, EPiCell achieves high fidelity against SPICE simulations, with low average prediction errors of 1.82% for leakage power, 4.01% for internal power, 3.06% for delay, and 3.29% for transition. Crucially, it demonstrates superior ranking consistency with SPICE, improving median Spearman's ρ from 0.25 to 0.9 for internal power, from 0.17 to 0.81 for delay, and from 0.14 to 0.70 for transition. This offers a high-fidelity and scalable surrogate model to enable efficient design space exploration.

KEYWORDS

Standard Cell Layout, PPA Prediction, Graph Transformer

1 INTRODUCTION

Standard cells are the cornerstone of modern very-large-scale integration (VLSI) design. In advanced process nodes, cell libraries have expanded dramatically, often containing thousands of unique logic functions to meet stringent power, performance, and area (PPA) targets. The manual design and optimization of these libraries is a critical bottleneck, characterized by long, iterative design cycles. To accelerate this process, automated standard cell layout synthesis has emerged as a powerful paradigm[15][24][14][12]. These tools can assist human designers by rapidly providing initial solutions or generating a vast number of layout variants for a single logic function[9][23]. This automation shifts the bottleneck from design creation to design evaluation. However, it is computationally infeasible to invoke the full SPICE characterization flow for every one of the potentially thousands of layout candidates. Concurrently, at advanced nodes, conventional geometry-driven heuristics (e.g., cell area, wire length) break down as layout effects dominate device behavior, decoupling geometric simplicity from actual electrical performance. This disconnect is explicitly demonstrated in Figure 1, which plots the performance versus wire length for numerous layout variants of the same cell. We observe not only a significant variation in PPA among these variants but also a clear decoupling from simple geometric metrics: the layout with the optimal wire length (blue dot) yields poor performance, while the layout with the optimal PPA (red dot) has a sub-optimal wire length. The breakdown of traditional heuristics has immediate consequences in real design flows—designers can no longer use geometric rules or coarse-grained proxies to reliably rank large sets of layout variants. This, in turn, makes it impossible to guarantee the quality of the initial candidates selected for manual refinement, ultimately degrading the overall efficiency of standard-cell library development.

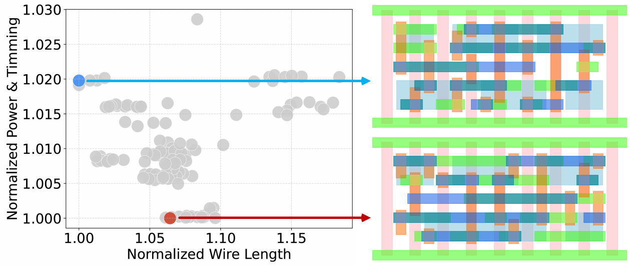

Figure 1: The decoupling of geometric heuristics from standard cell layout performance. The scatter plot shows 120 layout variants of A2O1A1O1Ixp33, plotting normalized wire length against normalized weighted sum of power and timing.

Therefore, a high-fidelity surrogate model must achieve both strong overall PPA prediction accuracy, and—more importantly—capability of capturing the subtle performance differences induced by layout effects. In this scenario, machine learning (ML)-based methods have emerged as a promising alternative for scalable PPA prediction. The landscape of performance prediction is vast, and most existing work addresses various problems at different design levels. Many prior efforts operate at the gate or block level rather than the standard cell design level[2][8][10][7][28][25], or they focus exclusively on isolated stages like transistor placement or routing quality[17][23][21][27][11][3][22], failing to capture the

[†] These authors contributed equally to this work.
[*] Corresponding author.

holistic impact of the final layout. Similarly, methods designed for Liberty generation typically rely on interpolation or migration strategies[20][4][26][16][18], which depend on pre-existing characterization data and cannot predict the performance of entirely new layout structures.

Consequently, the specific challenge of predicting PPA directly from transistor-level layout geometry remains underexplored. While recent layout-aware frameworks such as ProtoCellLayout[19] have made strides by incorporating geometric information, their modeling scope remains limited. [19] relies on symbolic one-hot encoding for timing arcs and local message-passing GNNs. This approach struggles to capture continuous timing-path interactions—particularly the coupled behaviors of transition and delay—and fails to model the global timing correlations essential for accurate ranking. Furthermore, topology-focused approaches like GTN-Cell[18] simplify routing into net-level abstractions, severely constraining their ability to capture the fine-grained layout dependent effects that dominate advanced nodes. These limitations underscore the critical need for a framework that effectively bridges circuit topology, geometric layout, and device-level physics, redefining PPA estimation as the process of learning the intrinsic correspondence between circuit structure, layout geometry, and electrical behavior.

With these insights, this paper proposes EPiCell, an Electro-Physical Co-Modeling Framework for Standard Cell PPA Estimation, leveraging a Heterogeneous Graph Transformer (HGT) to achieve comprehensive modeling that integrates electrical characteristics and physical layout geometry for high-fidelity PPA prediction.

The contributions of this work are summarized as follows:

- We propose **EPiCell**, an electro-physical co-modeling framework. It leverages a **Heterogeneous Graph Transformer (HGT)** that operates on a transistor-level heterogeneous graph, effectively captures complex electro-physical interactions and layout-dependent effects.
- We propose a dissimilarity-aware sampling strategy to efficiently cover the design space with limited but representative cell samples, avoiding cell characterization for massive layout variants with costly SPICE simulations.
- Validated on the ASAP7[5] PDK, EPiCell achieves a low average prediction error of 1.82% for leakage power, 4.01% for internal power, 3.06% for delay, and 3.29% for transition; and in ranking consistency with SPICE simulation, improves median Spearman's ρ from 0.25 to 0.90 for internal power, from 0.17 to 0.81 for delay, and from 0.14 to 0.70 for transition.

The rest of the paper is organized as follows: Section 2 introduces the EPiCell framework for PPA prediction; Section 3 explains the details of the proposed method; Section 4 validates the method with experimental results; Section 5 concludes the paper.

2 PRELIMINARY

2.1 Problem formulation

The standard cell characterization flow, as shown in the top path (blue) of Figure 2, is a multi-stage process used to generate accurate performance data. First, the cell layout undergoes Parasitic Extraction (PEX), where a detailed netlist is extracted, annotated with complex parasitic resistances and capacitances (RC), including inter-net coupling effects. This netlist is then passed to a SPICE Simulator, which runs extensive simulations to measure the cell's precise electrical behavior under various conditions. The final results, including power, delay, and transition metrics, are compiled into a Liberty (.lib) file. While this PEX-plus-SPICE workflow provides high fidelity, it is computationally intensive, making it infeasible for evaluating the large volume of layout variants produced, and thus unsuitable for rapid design space exploration.

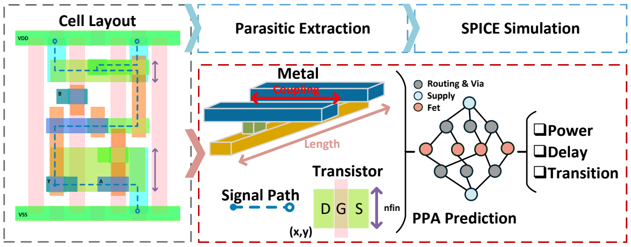

Figure 2: Comparison of standard cell layout evaluation flows. The traditional characterization flow (top, blue arrows) requires full Parasitic Extraction and SPICE Simulation. Our proposed flow (bottom, red arrows) uses a graph model based on extracted electro-physical details to directly predict PPA.

To address this challenge, we formulate the PPA prediction task as a graph-based learning problem as shown in the bottom path (red) of Figure 2. Our flow begins by extracting the critical electro-physical properties from the layout, serving as a lightweight proxy for the data computed during PEX. We capture transistor information, such as size (nfin) and position, and routing information including metal position, length, and features corresponding to parasitic coupling. These extracted details are then represented using a heterogeneous graph G, which explicitly distinguishes three entity types: FETs, Routing & Vias, and Voltage Supply nodes (VDD/VSS). This graph structure is featurized with the extracted physical properties and, critically, allows for the explicit modeling of charge/discharge paths from output nodes to the supply rails, which fundamentally govern timing behavior. This attributed graph G is then fed into our graph transformer-based prediction model, which is trained to mimic the SPICE simulation process by learning to propagate complex electro-physical interactions. The model's final output consists of the cell's key PPA metrics, including power, delay, and transition time, enabling rapid evaluation of layout candidates.

2.2 Rationale for Graph Representation and Transformer Modeling

A heterogeneous electro-physical graph is essential for explicitly representing these entities and their role-specific interactions. FETs, routing metals, and supply rails contribute to PPA through inherently different physical mechanisms, and affect timing and power in asymmetric ways. Collapsing these entities into a homogeneous representation forces the model to infer their differences implicitly, making it difficult to attribute layout effects to the correct physical causes, while modeling them as heterogeneous node types preserves their semantic distinctions and allows the network

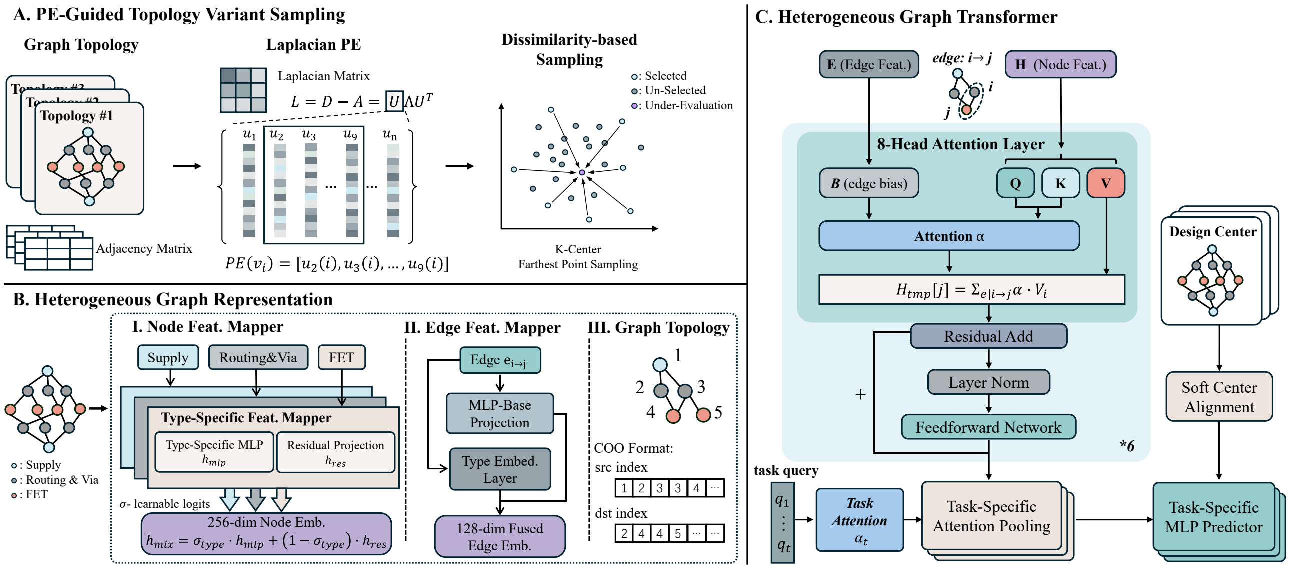

Figure 3: The Framework of EPiCell - A. Sampling Topology Variant for Dataset Construction B. Transforming Std-Cell to Graph Representation C. Leveraging a Heterogeneous Graph Transformer for PPA Prediction

to learn how each entity's variations propagate through the electrical chain.

Conventional GNNs, fall short in this scenario because fixed-hop message passing cannot capture the non-local, multi-stage behaviors that underlie layout effects. In contrast, transformer-based attention performs dynamic, content-aware dependency modeling. When combined with our circuit-structured DAG, attention effectively learns physically valid signal- and supply-propagation paths rather than arbitrary graph interactions, thus enabling the model to track how layout-dependent perturbations accumulate along real electrical chains and ultimately influence power and timing.

The combination of heterogeneous representation and edge-restricted transformer attention yields a surrogate capable of discriminating variations, enabling accurate fine-grained PPA ranking across large layout-variant sets.

3 METHODOLOGY

3.1 Layout Generation & Topology Extraction

3.1.1 Layout Variants Generation: We employ an **SMT-based (Satisfiability Modulo Theories) method for standard cell layout generation**, which is similar in principle to [15]. This approach models the transistor placement and routing problem as a constraint satisfaction problem, where each feasible solution corresponds to a unique layout design.

To obtain diverse layout variants, we utilize an **ALLSAT-like enumeration strategy**. Specifically, whenever a feasible solution x_{new} is found, a new blocking clause ($x \neq x_{new}$) is added to the constraint set to exclude this solution from subsequent searches. To ensure both generation efficiency and solution diversity, we adopt a two-stage, decoupled approach. First, we generate multiple distinct placement solutions. Then, for each of these placement solutions, we apply a similar enumeration method to obtain multiple corresponding routing solutions.

3.1.2 Layout Topology Extraction: Although topological relations and geometric information could be directly exported during our generation process, we established a streamlined extraction flow to ensure our framework can process layouts from arbitrary sources, including manually-designed cells. Our extraction flow is built upon commercial LVS (Layout Versus Schematic) tool. The process begins by utilizing the transistor placement data provided by the LVS report. Following this, a graph traversal is initiated from the I/O pins, navigating the circuit structure based on the layout's interlayer connectivity rules. This traversal sequentially identifies the specific geometric shapes that constitute each net. This extracted electro-physical information serves as the final input for our prediction model.

3.2 PE-Guided Topology Variant Sampling

Automated standard-cell generation tools can synthesize an extensive set of layout variants derived from the same logic design, making full SPICE characterization impractical. When constructing the training set for the surrogate model, a key trade-off arises: using as few cells as possible to mitigate characterization overhead while still ensuring that the selected samples span the design space with sufficient diversity to support strong model generalization.

Laplacian positional embeddings (PE) are highly sensitive to topological variations in the transistor-level connectivity graph, aligning with observations from [1] which demonstrate the ability of Laplacian eigenfunctions to capture structural differences across graph topologies. Since all variants within the same design share the same number of FET instances, the resulting PE vectors become naturally comparable, enabling a meaningful measurement of topological dissimilarity. Based on these observations, we

Table 1: Details of Heterogeneous Node Encoding

Shared Feat.	Feature	Dim
Laplacian-PE	8 smallest non-trivial Laplacian eigenvectors as node coordinates	8
Type Encoding	[FET, Routing/Via, Supply]	3
Supply Path	[if_SupplyPath,PullUp,PullDown]	3
Signal Path	[if_SignalPath, Input, Output]	3
Geometry	Bounding-box $[(x_1, y_1), (x_2, y_2)]$ + Centroid-Coor $[(x_c, y_c)]$+ Area	7
Type-Specific	**Feature**	**Tot. Dim**
FET	w + l + nfin + if_flip + n/pmos	29
Routing[-1]	layer + cpl_{vertical} + $cpl_{\text{horizontal}}$	27
Supply	[if__Power-Rail, VDD, VSS]	27

[1] cpl stands for coupling, capturing metal-to-metal coupling. Horizontal coupling is the w/l ratio of same-layer metals, vertical coupling is normalized overlap area across layers.

employ a k-center farthest-point sampling procedure as demonstrated in Figure 3-A that iteratively selects the cell whose PE lies farthest from the current sampled set. This PE-guided strategy effectively identifies variants with maximal topological diversity, ensuring uniform coverage of the design space and enhancing the robustness and generalization capability of the PPA surrogate model.

3.2.1 Node Encoding: Circuit instances such as FETs, routing segments, and supply nodes exhibit fundamentally different electrical behaviors and interaction patterns. Conventional graph learning approaches often treat these entities as homogeneous nodes, inevitably compressing their semantic distinctions and distorting their functional characteristics. Even methods that incorporate one-hot node types still rely on a shared embedding function, a representation-level compromise that obscures the intrinsic heterogeneity of circuit element.

To address this limitation, we design fully type-specific encoding schemes that allow each node category to retain its electro-physical semantics within the representation, as shown in Tab.1. To further equip each node with global topology awareness, we leverage the eight smallest non-trivial Laplacian eigenvectors of the connectivity topology and incorporate these Laplacian positional embeddings into node feature vector. These heterogeneous encodings are further processed by type-specific feature mappers, each implemented as a gated fusion of a residual projection path and an MLP transformation path. The gating mechanism adaptively balances linear and nonlinear feature components, producing a 256-dim unified embedding for all node types. This design preserves behavioral diversity while enabling coherent interaction modeling within the heterogeneous graph transformer.

3.2.2 Edge Encoding: Existing circuit-graph representations simplify edges as mere indicators of physical connectivity, failing to convey functional roles in signal propagation or their geometric significance in supply conduction. Therefore, we design a relation-aware edge encoding that integrates a 9-dim type one-hot vector, functional-path descriptors for signal and supply behavior, and the centroid distance between connected instances. This composite encoding is further passed through an MLP-based projection and gated fusion with edge-type embedding, producing a 128-dim fused edge representation. This modeling method yields relational semantics, functional dependencies, and spatial context essential for accurate electro-physical reasoning.

3.2.3 Graph Topology: We construct circuit topology in COO format, encoding directed source–destination relations to form a directed acyclic graph (DAG) that reflects signal flow. This COO topology then serves as the structural backbone for attention propagation in the graph transformer.

3.3 Heterogeneous Graph Transformer

3.3.1 Edge-Restricted Multi-Head Attention Encoder: The overall encoder adopts a 6-layer, 8-head graph transformer architecture, where each layer consists of a connectivity-aware multi-head attention block followed by a Residual–LayerNorm–FFN update sequence.

At the core of each layer is the edge-restricted graph attention mechanism, as shown in Fig.3. Inspired by [6], rather than adopting fully connected self-attention — which would compute interactions among all N nodes — our encoder restricts message passing strictly to edges defined by the circuit graph, to leverage the graph connectivity inductive bias. For each directed edge (i⟶j), the layer computes attention scores using source-node queries and destination-node keys, refined by edge-conditioned biases derived from edge embeddings, as demonstrated in Equation 1.

$$\begin{aligned} s_{ij} &= \frac{\left((W_Q \mathrm{h}_i)\right)^{\top}(W_K \mathrm{h}_j)}{\sqrt{d_h}} + \left[g(\mathrm{e}_{ij})\right] \qquad edge: i \longrightarrow j \in \mathcal{E}, \\ \alpha_{ij} &= \mathrm{softmax}_{i:(i\rightarrow j)\in\mathcal{E}}\left(s_{ij}\right). \end{aligned} \tag{1}$$

where h_i and h_j represents node embedding of source-node and destination-node respectively, e_{ij} is edge embedding of edge (i⟶j), s_{ij} is the attention score and α_{ij} is the corresponding attention.

We adopt a source-as-query and destination-as-key formulation so that upstream circuit elements determine their influence on downstream nodes, naturally aligning the attention mechanism with the intrinsic signal-flow or supply-net behavior of circuit DAGs, prevents semantically invalid interactions, and significantly improves computational efficiency: its complexity scales as $O(E \cdot h \cdot d)$, where E is the number of edges, h the number of heads, and d the head dimension, in contrast to the $O(N^2 \cdot h \cdot d)$ cost of fully connected transformers.

3.3.2 Task-Query Attention Decoder for PPA Prediction: Beyond the encoder, EPiCell employs a task-adaptive predictor to transform node-level representations into cell-level PPA outputs. Instead of relying on global pooling, we introduce a task-query attention as demonstrated in Equation 2, readout that enables each PPA metric to selectively aggregate information from structurally relevant circuit regions.

$$\begin{aligned} \alpha_{i,t} &= \mathrm{softmax}_{\,i\in\mathcal{V}_c}\left(\mathrm{q}_t^{\top}\mathrm{h}_i\right), \\ \mathrm{z}_t &= \sum_{i\in\mathcal{V}_c} \alpha_{i,t}\,\mathrm{h}_i, \\ \hat{y}_t &= \mathrm{MLP}_t(\mathrm{z}_t) \end{aligned} \tag{2}$$

where h_i is the representation of node i produced by encoder, q_t is a learnable task query associated with prediction task t, z_t is the attention-pooled graph representation for task t, and $\hat{y}_t$ denotes the final PPA prediction obtained through the task-specific MLP head.

For each prediction task, a learnable query vector interacts with all encoded node embeddings via a dot-product scoring function, which is normalized using a graph-aware segment-wise softmax over each cell. This produces an attention distribution highlighting nodes that are most informative for the corresponding PPA component—e.g., pull-up paths for cell rise, pull-down networks for cell fall, or critical transistors for leakage power.

The attention-weighted node embeddings are then summed to yield a compact task-specific graph representation. This pooled feature is passed through a two-layer MLP head with GELU activation and dropout to generate the final numerical outputs for each PPA task. By assigning independent queries and MLP heads per task, the predictor naturally accommodates heterogeneous PPA behaviors while sharing a unified encoder backbone.

3.3.3 Soft Center Alignment: We further adopt a log-ratio–based contrastive regularizer to implement the alignment dynamics between each embedding and its associated design centers. The pull term encourages the embedding z to move toward the 8 nearest design centers by minimizing a soft distance-weighted attraction. In contrast, the push term penalizes configurations in which z is relatively closer to distant, irrelevant centers than to the selected top-8 ones, effectively enlarging the separation margin between design families. The overall center-alignment objective combines these two components:

$$L_{\text{pull}} = \sum_{d \in C_8(\mathrm{z})} w_d \, d(\mathrm{z}, \mathrm{c}_d), \tag{3}$$

$$L_{\text{push}} = -\lambda_{\text{push}} \log \frac{\sum_{d \in C_8(\mathrm{z})} \exp(-d(\mathrm{z}, \mathrm{c}_d))}{\sum_{d \notin C_8(\mathrm{z})} \exp(-d(\mathrm{z}, \mathrm{c}_d))}. \tag{4}$$

Where $C_8(\mathrm{z})$ denotes the 8 nearest design centers, $d(\mathrm{z}, \mathrm{c}_d)$ is the Euclidean distance, w_d is the distance-based softmax weight within $C_8(\mathrm{z})$, and λ_{push} controls the repulsion strength.

$$L_{\text{center-align}} = L_{\text{pull}} + L_{\text{push}}. \tag{5}$$

The resulting regularizer, as shown in Equation 5, integrates both the attractive and repulsive forces to regularize the embedding space, thereby encouraging a well-shaped, topology-aware embedding space that preserves intra-design consistency and enhances inter-design separation.

3.3.4 Overall Training Objective: To jointly optimize task-specific PPA prediction and encourage a more structured embedding geometry, we augment the uncertainty–weighted multi-task regression objective with our proposed contrastive center-alignment loss. The regression component follows a homoscedastic uncertainty formulation inspired by [13], allowing the model to adaptively balance heterogeneous PPA tasks, while the alignment term regularizes the representation space by encouraging embeddings to remain close to their nearest design centers. The overall training loss is defined as:

$$L_{\text{Total}} = \sum_t \left(e^{-\log \sigma_t^2} \, \|\hat{y}_t - y_t\|^2 + \log \sigma_t^2 \right) + \lambda L_{\text{center-align}} \tag{6}$$

Table 2: Generated standard cell types and the number of layout variants after sampling.

Cell Type	#Layout Variants	Example Cells
AND	1360	AND2x6,AND3x4,AND5x2
AO/OA	12739	A2O1A1Ixp33,AOI333xp17,OA32x2
MAJ	518	MAJx1,MAJx2,MAJx3
NAND	1441	NAND2x2,NAND4xp75,NAND5xp2
NOR	1302	NOR3xp33,NOR4xp25,NOR5xp2
OR	1336	OR2x4,OR3x4,OR5x2

4 EXPERIMENT

4.1 Experimental Setup & Dataset

Our experiments are based on the open-source ASAP7 PDK. We use Calibre (v2023.2) for LVS and PEX, and LIBERATE (v23.2.3) for characterization. All model training and inference are conducted on a platform equipped with an AMD EPYC 7543 CPU and an NVIDIA GeForce RTX 5090 GPU.

Following the generation methodology outlined in Section 3.1, we initially synthesized a large-scale pool of approximately 76,000 standard cell layout variants. We applied the sampling strategy described in Section 3.2 to select a subset of variants that maximizes design space diversity. Consequently, we obtained a final dataset of approximately 18,000 samples. For each sample in this curated set, we performed the complete characterization flow to acquire the ground-truth layout PPA metrics. The detailed breakdown of cell types and their corresponding variant counts in this final dataset is summarized in Table 2.

Our prediction framework targets PPA metrics including leakage power, internal power, cell rise delay, cell fall delay, rise transition, and fall transition. For the timing-related metrics, standard cells typically exhibit multiple timing arcs, each characterized by a Non-Linear Delay Model (NLDM) look-up table (LUT). To derive a scalar ground-truth value for training and evaluation, we compute the average value of the entries in the NLDM LUTs for all valid timing arcs of the cell.

4.2 Accuracy Validation

In this section, we conduct a comparative analysis between EPiCell and ProtoCellLayout [19], a state-of-the-art layout-aware PPA estimation framework to validate EPiCell's accuracy.

To systematically evaluate the prediction performance of different PPA surrogates, we consider three complementary experimental settings:

(a) Full: All cell instances in the dataset are randomly partitioned into 80% training and 20% testing. This configuration evaluates the model's upper-bound prediction accuracy when sufficient training data are available
(b) Variant-limited: For each design, only 30% of its standard-cell variants are uniformly sampled as training data, while the remaining 70% are reserved for testing. This setting emulates the model's few-shot generalization capability within a design.
(c) Cell-limited: Only 30% of the cell types are used for training (with all their associated cells), and the remaining 70% designs are held out for testing. This configuration focuses on the model's ability to generalize across designs.

Table 3: Accuracy comparison between [19] and EPiCell under three evaluation settings

Setting	Method	Leakage Power		Internal Power		Cell Rise		Cell Fall		Rise Transition		Fall Transition	
		MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2
Full	[19]	1.775	0.997	9.049	0.945	5.490	0.978	4.134	0.991	5.516	0.988	5.876	0.983
	EPiCell	1.822	0.998	**4.013**	0.983	**2.798**	0.966	**3.318**	0.989	**3.153**	0.980	**3.423**	0.987
Variant-limited	[19]	9.331	0.942	12.261	0.910	6.428	0.970	8.329	0.979	6.163	0.981	7.324	0.972
	EPiCell	**8.354**	0.961	**7.270**	0.977	6.723	0.953	**4.840**	0.935	**4.136**	0.973	**4.714**	0.957
Cell-limited	[19]	12.426	0.931	21.628	0.899	16.623	0.914	13.368	0.932	9.784	0.901	14.577	0.963
	EPiCell	**11.371**	0.922	**14.572**	0.912	**11.672**	0.931	**12.257**	0.928	**8.363**	0.913	**9.792**	0.948

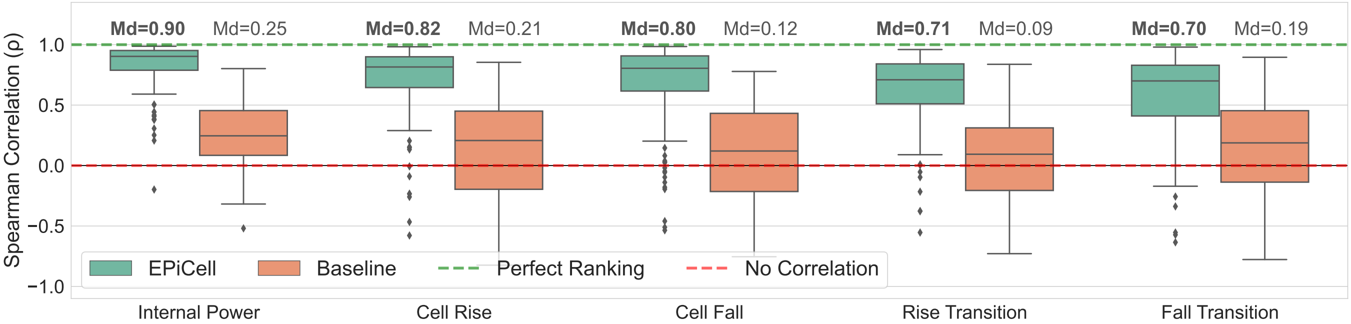

Figure 4: Ranking consistency with SPICE ground truth across key PPA metrics. Each point corresponds to one standard cell, and Md denotes the median Spearman's ρ.

As summarized in Table 3, EPiCell consistently demonstrates superior performance across all three generalization settings. Specifically, in the ***Full*** **setting**, our model achieves high-fidelity predictions with MAPEs of 1.82% for leakage power and 4.01% for internal power. For timing metrics, EPiCell maintains exceptional accuracy, achieving prediction errors of 3.06% for delay and 3.29% for transition, corresponding to 2.80% for cell rise, 3.32% for cell fall, 3.15% for rise transition and 3.42% for fall transition. This advantage persists even in the challenging *Cell-limited* setting, confirming that our heterogeneous graph formulation captures transferable electro-physical principles rather than merely memorizing cell-specific attributes.

While [19] achieves high global R^2 values comparable to EPiCell in certain metrics, this does not necessarily imply effective layout awareness. In standard cell libraries, PPA values are dominated by inter-class variations (i.e., differences between logic functions and drive strengths), whereas intra-class variations (differences between layout variants of the same cell) are relatively small. Consequently, a high global R^2 primarily reflects a model's ability to distinguish between different cell types and drive strengths. As we will demonstrate in the following section, despite similar global statistics, EPiCell significantly outperforms [19] in ranking the quality of the layout, validating its sensitivity to layout effects.

4.3 Consistency with SPICE Simulation-based Layout Ranking

We evaluate the consistency with SPICE simulation-based layout ranking using the *Full* Setting dataset described in Section 4.2. which consists of 133 unique standard cells (differentiated by function and drive strength), and each cell contains 20 to 30 layout variants. Here, we employ **Spearman's Rank Correlation Coefficient (ρ)** to quantify the consistency between the predicted ranking and the ground-truth SPICE simulation ranking. Note that leakage power is excluded from this analysis as it is primarily determined by the circuit netlist and shows minimal sensitivity to layout variations compared to dynamic power and timing.

As shown in Figure 4, [19] exhibits marginal correlation ($\rho \leq 0.25$) across all metrics, indicating a failure to distinguish between layout variants. In contrast, our model demonstrates robust layout variation awareness. Specifically, our method improves a median ρ of from 0.25 to **0.90** for Internal Power, from 0.17 to **0.81** for Cell Delay, and from 0.14 to **0.70** for Transition metrics compared with our baseline. These results confirm that our approach effectively captures layout-dependent parasitic effects, providing accurate and reliable ranking guidance for layout selection.

5 CONCLUSION

In this work, we presented EPiCell, a heterogeneous electro-physical modeling framework for accurate and scalable standard-cell PPA prediction. By integrating device geometry and routing context with an edge-restricted graph transformer, EPiCell captures the electro-physical interactions that govern layout effects. Evaluated on 18,000 ASAP7 layout variants, EPiCell achieves prediction errors of 1.82% for leakage power, 4.01% for internal power, 3.06% for delay, and 3.29% for transition, while improving median Spearman's ρ from 0.25 to **0.90** for internal power, from 0.17 to **0.81** for delay, and from 0.14 to **0.70** for transition. EPiCell provides a high-fidelity surrogate for large-scale layout exploration and more efficient standard-cell library development.

ACKNOWLEDGEMENT

This project is supported in part by National Science and Technology Major Project (2021ZD0114702).

REFERENCES

[1] Mikhail Belkin and Partha Niyogi. 2001. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani (Eds.), Vol. 14. MIT Press. https://proceedings.neurips.cc/paper_files/paper/2001/file/f106b7f99d2cb30c3db1c3cc0fde9ccb-Paper.pdf

[2] Yifan Chen, Jing Mai, Xiaohan Gao, Muhan Zhang, and Yibo Lin. 2023. MacroRank: Ranking Macro Placement Solutions Leveraging Translation Equivariancy. In *Proceedings of the 28th Asia and South Pacific Design Automation Conference (ASPDAC '23)*. Association for Computing Machinery, New York, NY, USA, 258–263. https://doi.org/10.1145/3566097.3567899

[3] Yuyang Chen, Yiwen Wu, Jingya Wang, Tao Wu, Xuming He, Jingyi Yu, and Hao Geng. 2024. LLM-HD: Layout Language Model for Hotspot Detection with GDS Semantic Encoding. 1–6. https://doi.org/10.1145/3649329.3658479

[4] Xu Cheng, Yuyang Ye, Guoqing He, Qianqian Song, and Peng Cao. 2024. Heterogeneous Graph Attention Network Based Statistical Timing Library Characterization with Parasitic RC Reduction. 171–176. https://doi.org/10.1109/ASP-DAC58780.2024.10473881

[5] Lawrence T Clark, Vinay Vashishtha, Lucian Shifren, Aditya Gujja, Saurabh Sinha, Brian Cline, Chandarasekaran Ramamurthy, and Greg Yeric. 2016. ASAP7: A 7-nm finFET predictive process design kit. *Microelectronics Journal* 53 (2016), 105–115.

[6] Vijay Prakash Dwivedi and Xavier Bresson. 2021. A Generalization of Transformer Networks to Graphs. arXiv:cs.LG/2012.09699 https://arxiv.org/abs/2012.09699

[7] Wenji Fang, Yao Lu, Shang Liu, Qijun Zhang, Ceyu Xu, Lisa Wu Wills, Hongce Zhang, and Zhiyao Xie. 2023. MasterRTL: A Pre-Synthesis PPA Estimation Framework for Any RTL Design. arXiv:cs.AR/2311.08441 https://arxiv.org/abs/2311.08441

[8] Zizheng Guo, Mingjie Liu, Jiaqi Gu, Shuhan Zhang, David Z. Pan, and Yibo Lin. 2022. A timing engine inspired graph neural network model for pre-routing slack prediction. In *Proceedings of the 59th ACM/IEEE Design Automation Conference (DAC '22)*. Association for Computing Machinery, New York, NY, USA, 1207–1212. https://doi.org/10.1145/3489517.3530597

[9] Chia-Tung Ho, Alvin Ho, Matthew Fojtik, Minsoo Kim, Shang Wei, Yaguang Li, Brucek Khailany, and Haoxing Ren. 2023. 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*. ACM, Virtual Event USA, 44–52.

[10] Chia-Tung Ho and Andrew B. Kahng. 2019. IncPIRD: Fast Learning-Based Prediction of Incremental IR Drop. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–8. https://doi.org/10.1109/ICCAD45719.2019.8942110

[11] Yiyang Jiang, Fan Yang, Bei Yu, Dian Zhou, and Xuan Zeng. 2021. Efficient Layout Hotspot Detection via Binarized Residual Neural Network Ensemble. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 40, 7 (2021), 1476–1488. https://doi.org/10.1109/TCAD.2020.3015918

[12] Yibo Lin Kairong Guo. 2025. Multi-Row Standard Cell Layout Synthesis with Enhanced Scalability. In *2025 3nd International Symposium of Electronics Design Automation (ISEDA)*. 1–6.

[13] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2017. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. *CoRR* abs/1705.07115 (2017). arXiv:1705.07115 http://arxiv.org/abs/1705.07115

[14] Suwan Kim and Taewhan Kim. 2024. Optimal Transistor Folding and Placement for Synthesizing Standard Cells of Complementary FET Technology. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*. 1–6.

[15] Daeyeal Lee, Dongwon Park, Chia-Tung Ho, Ilgweon Kang, Hayoung Kim, Sicun Gao, Bill Lin, and Chung-Kuan Cheng. 2021. 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* 40, 10 (2021), 2142–2155.

[16] Jaeseung Lee, Sunggyu Jang, Jakang Lee, and Seokhyeong Kang. 2025. *LIBMixer: An all-MLP Architecture for Cell Library Characterization towards Design Space Optimization*. Association for Computing Machinery, New York, NY, USA, 814–820. https://doi.org/10.1145/3658617.3697602

[17] Tai-Cheng Lee, Cheng-Yen Yang, and Yih-Lang Li. 2020. /TPlace: Machine Learning-Based Delay-Aware Transistor Placement for Standard Cell Synthesis. In *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. 1–8.

[18] Lihao Liu, Yunhui Li, Beisi Lu, Li Shang, and Fan Yang. 2025. GTN-Cell: Efficient Standard Cell Characterization Using Graph Transformer Network. In *2025 Design, Automation & Test in Europe Conference (DATE)*. IEEE, 1–7.

[19] Zhiyuan Luo, Le Zhou, Zenghui Zhang, Jie Zhou, Zhien Li, Huiqing You, Feng Yao, and Zhenyu Zhao. 2025. 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)*. 1–9.

[20] Tianliang Ma, Zhihui Deng, Sun Xuguang, and Leilai Shao. 2024. Fast Cell Library Characterization for Design Technology Co-Optimization Based on Graph Neural Networks. 472–477. https://doi.org/10.1109/ASP-DAC58780.2024.10473933

[21] Haoxing Ren, George F. Kokai, Walker J. Turner, and Ting-Sheng Ku. 2020. ParaGraph: Layout Parasitics and Device Parameter Prediction using Graph Neural Networks. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*. 1–6. https://doi.org/10.1109/DAC18072.2020.9218515

[22] Shuyuan Sun, Yiyang Jiang, Fan Yang, Bei Yu, and Xuan Zeng. 2022. Efficient Hotspot Detection via Graph Neural Network. In *2022 Design, Automation Test in Europe Conference Exhibition (DATE)*. 1233–1238. https://doi.org/10.23919/DATE54114.2022.9774579

[23] Jiun-Cheng Tsai, Wei-Min Hsu, Yun-Ting Hsieh, Yu-Ju Li, Wei Huang, CN Ho, Hsuan-Ming Huang, Jen-Hang Yang, Heng-Liang Huang, Aaron C-W Liang, et al. 2024. MAXCell: PPA-Directed Multi-Height Cell Layout Routing Optimization using Anytime MaXSAT with Constraint Learning. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*. 1–9.

[24] Pascal Van Cleeff, Stefan Hougardy, Jannik Silvanus, and Tobias Werner. 2020. BonnCell: Automatic Cell Layout in the 7-nm Era. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 10 (2020), 2872–2885.

[25] Ziyi Wang, Siting Liu, Yuan Pu, Song Chen, Tsung-Yi Ho, and Bei Yu. 2023. Restructure-Tolerant Timing Prediction via Multimodal Fusion. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*. 1–6. https://doi.org/10.1109/DAC56929.2023.10247802

[26] Hao-Yu Wu, Hsin-Tzu Chang, Shiuan-Yun Ding, Iris Hui-Ru Jiang, Benson Tsao, Vinson Wu, and Wei-Kai Shih. 2025. Generative Model Based Standard Cell Timing Library Characterization. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*. 1–7. https://doi.org/10.1109/DAC63849.2025.11133303

[27] H.Y. Yang, Jing Su, Yi Zou, Yuzhe Ma, Bei Yu, and Evangeline Young. 2018. Layout Hotspot Detection With Feature Tensor Generation and Deep Biased Learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* PP (05 2018), 1–1. https://doi.org/10.1109/TCAD.2018.2837078

[28] Ruizhe Zhong, Junjie Ye, Zhentao Tang, Shixiong Kai, Mingxuan Yuan, Jianye Hao, and Junchi Yan. 2024. PreRoutGNN for Timing Prediction with Order Preserving Partition: Global Circuit Pre-training, Local Delay Learning and Attentional Cell Modeling. arXiv:cs.LG/2403.00012 https://arxiv.org/abs/2403.00012