International Symposium on Physical Design
March 26-29, 2023
Online with virtual participation

NVCell 2

Routability-Driven Standard Cell Layout in Advanced Nodes with Lattice Graph Routability Model

Chia-Tung (Mark) Ho, Alvin Ho, Matthew Fojtik, Minsoo Kim,
Shang Wei, Yaguang Li, Brucek Khailany, Haoxing Ren

NVIDIA

ACKNOWLEDGEMENT

NVIDIA colleagues: Alvin Ho, Matthew Fojtik, Minsoo Kim, Shang Wei, Yaguang Li, Brucek Khailany, **Haoxing Ren**

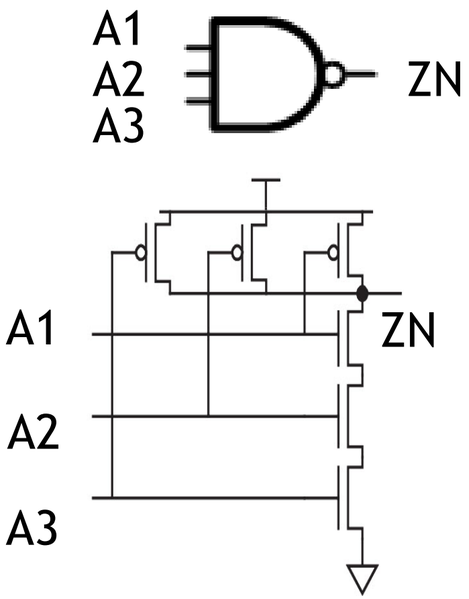
OUTLINES

- Background & Motivation
- Related Works & Contributions
- Preliminary: NVCell
- Routability-Driven Standard Cell Automation Framework
- Experimental results
- Conclusions & Future Works

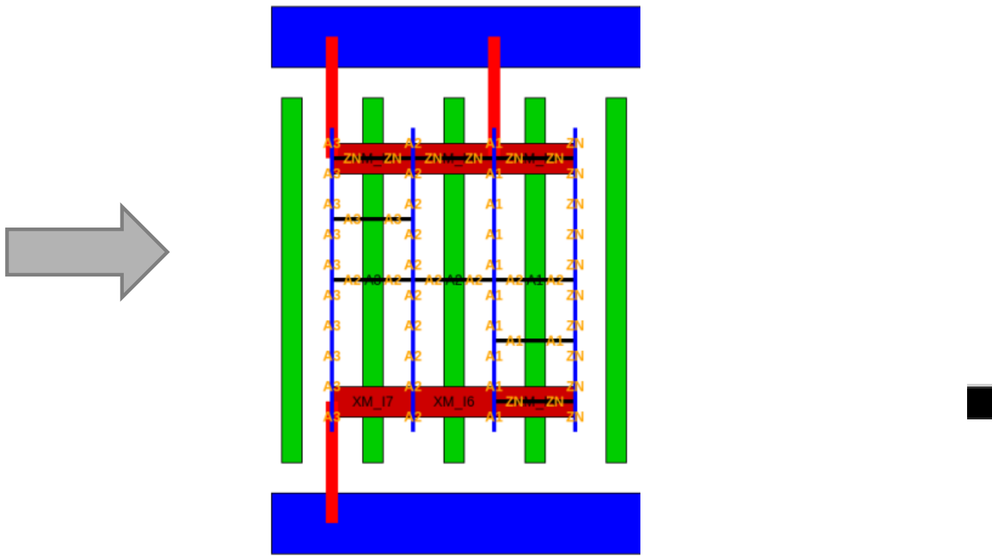
BACKGROUND & MOTIVATION

STANDARD CELL LAYOUT AUTOMATION

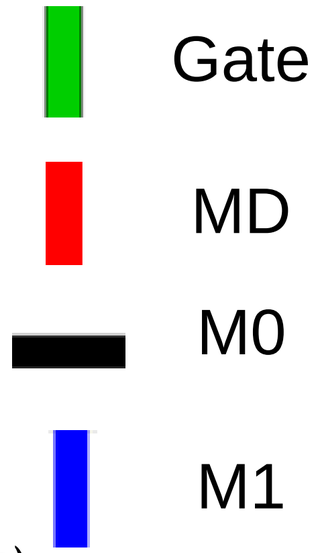
- Std cells are building blocks of digital design layout: AND, NOR, Flip-Flop, Adder, etc
- Layout mostly by hand today, long design turn around time for the library (a few months)
- Standard cell automatic layouts - Fast design turn around time, More custom cell design, Design Technology Co-Optimization



Schematic

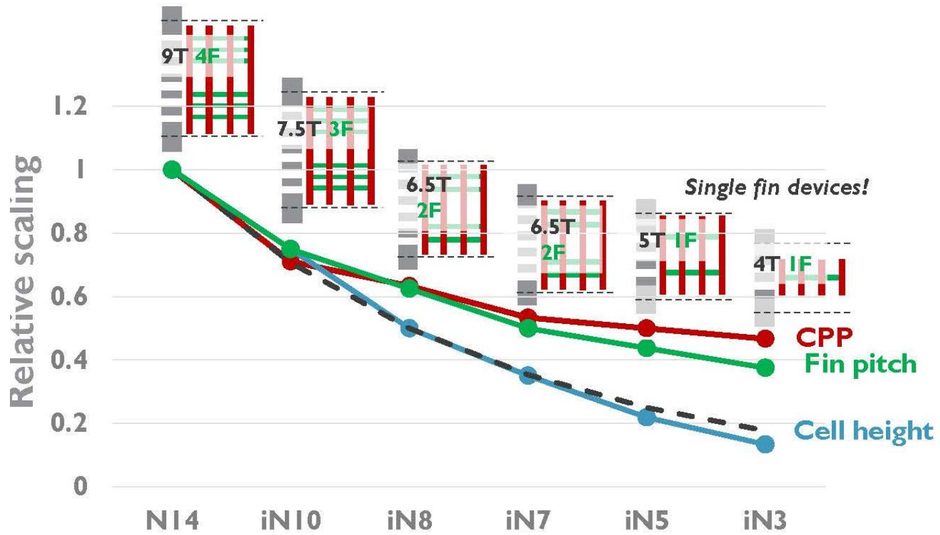


Simplified Grid-based layout diagram (Sticks)

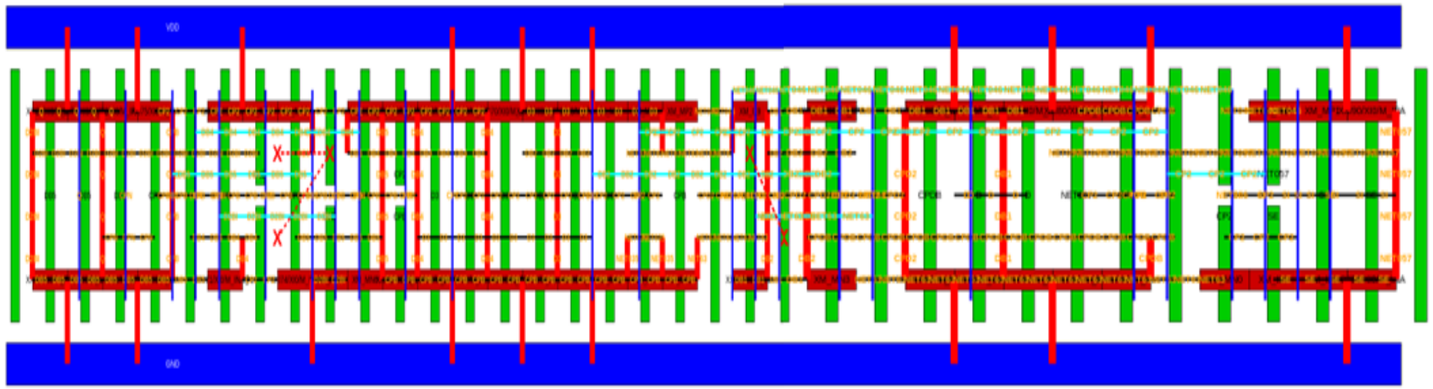


ROUTABILITY CHALLENGES

- Std cell height scaling is essential to advance the technology for Power, Performance, Area, and Cost (PPAC).
- Routability challenges
  - Limited in-cell routing resource: less horizontal routing tracks (i.e., < 5 RTs)
  - Increasing number and complexity of design rules + strict patterning rules



Standard Cell Scaling Roadmap from IMEC



An example of routability issue in the advanced node of a flip-flop layout design

Source: https://www.imec-int.com/en/imec-magazine/imec-magazine-november-2018/the-supervia-a-promising-scaling-booster-for-the-sub-3nm-technology-node

RELATED WORKS & CONTRIBUTIONS

RELATED WORKS

- Sequential standard cell synthesis approach [1], [2], [5], and [6]
  - Performs the placement step first and then the routing step
  - Placement: heuristic based methods, exhaustive search based methods, mathematical programming based methods, and simulated annealing technique.
  - Routing: channel routing, SAT, and Mixed-Integer Linear Programming(MILP) based routing methods.
  - Recently, Ren et al. [1] used the simulated annealing technique to generate optimal transistor placement, leveraged genetic algorithms for routing, and applied reinforcement learning to fix the design rule violation.

Still struggling to generate routable placements for routing in the advanced nodes.

- Simultaneously standard cell synthesis approach [7], [8], [9], and [10]
  - Encoding the design rules in the engine and generate routable standard cell layouts using Satisfiability-modulo theory (SMT).

Scalability is worse on large and complex standard cell designs (i.e., more than 50 devices).

CONTRIBUTIONS

- Propose a novel Pin Density Aware (PDA) congestion metric to capture the routability of local areas.

  -> Achieves correlations of 0.9543 and 0.8364 with the average routing congestion and the area of golden unrouted probability distribution, respectively.

- Develop a novel lattice graph routability modeling approach to capture the routability of local areas, routability impacts between local areas, and global net connections.

  -> Achieves correlations of 0.9608 and 0.8536 with the average routing congestion and the area of golden unrouted probability distribution, respectively.

- Propose a dynamic standard cell external pin allocation methodology.

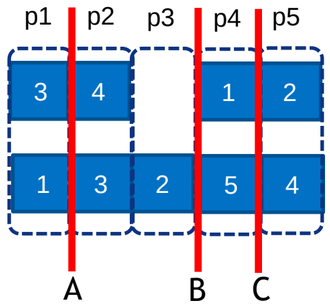
  -> Improve routability and design rule fixing in the routing phase.

- Achieves cell layouts with smaller area than the existing industrial standard cell library for 13.9% of over 1000 cells.

PRELIMINARY: NVCELL [1]

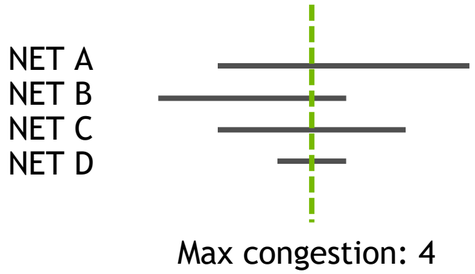
NVCELL PLACEMENT

- Simulated Annealing based algorithm for placement: Swap, Move, Flip
- Heuristic based congestion estimation
- Simple model-based routability model (1D conv model and max pool embeddings)
  - Input features: *[#poly nets, left pmos diffusion connected, left nmos diffusion connected, right pmos diffusion connected, right nmos diffusion connected, #M1 pins on the left, #M1 pins on the right, #of nets cross poly]*
  - Predict: [$routable$, $routablebutwithDRCs$, $notroutable$] for each placement




```
PMOS sequence: 3,  4,  NA, 1,  2
NMOS sequence: 1,  3,  2,  5,  4
PIN  sequence: A,  NA, B,  C,  NA
```

Swap, move, and flip of placement sequence



Heuristic based congestion estimation

NVCELL ROUTING

- Leverage maze routing to generate routing candidates

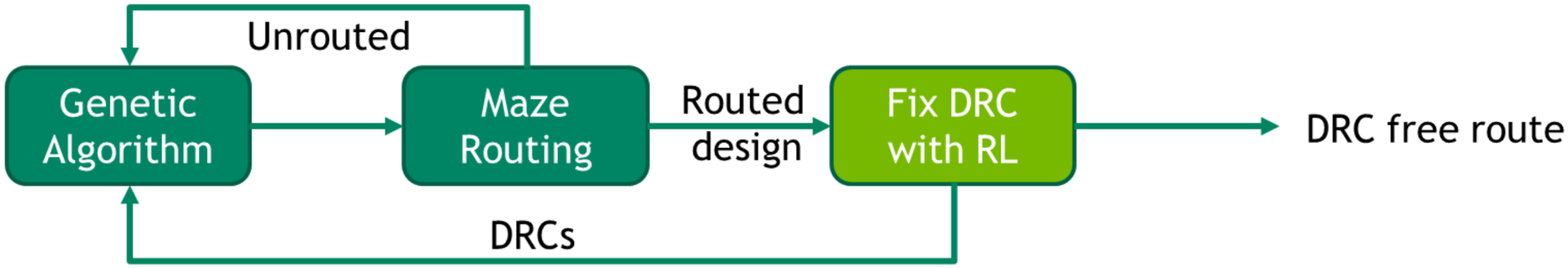
  → solve the connectivity problem

- Leverage RL to fix DRC of the routing candidates

  → solve the DRC problem

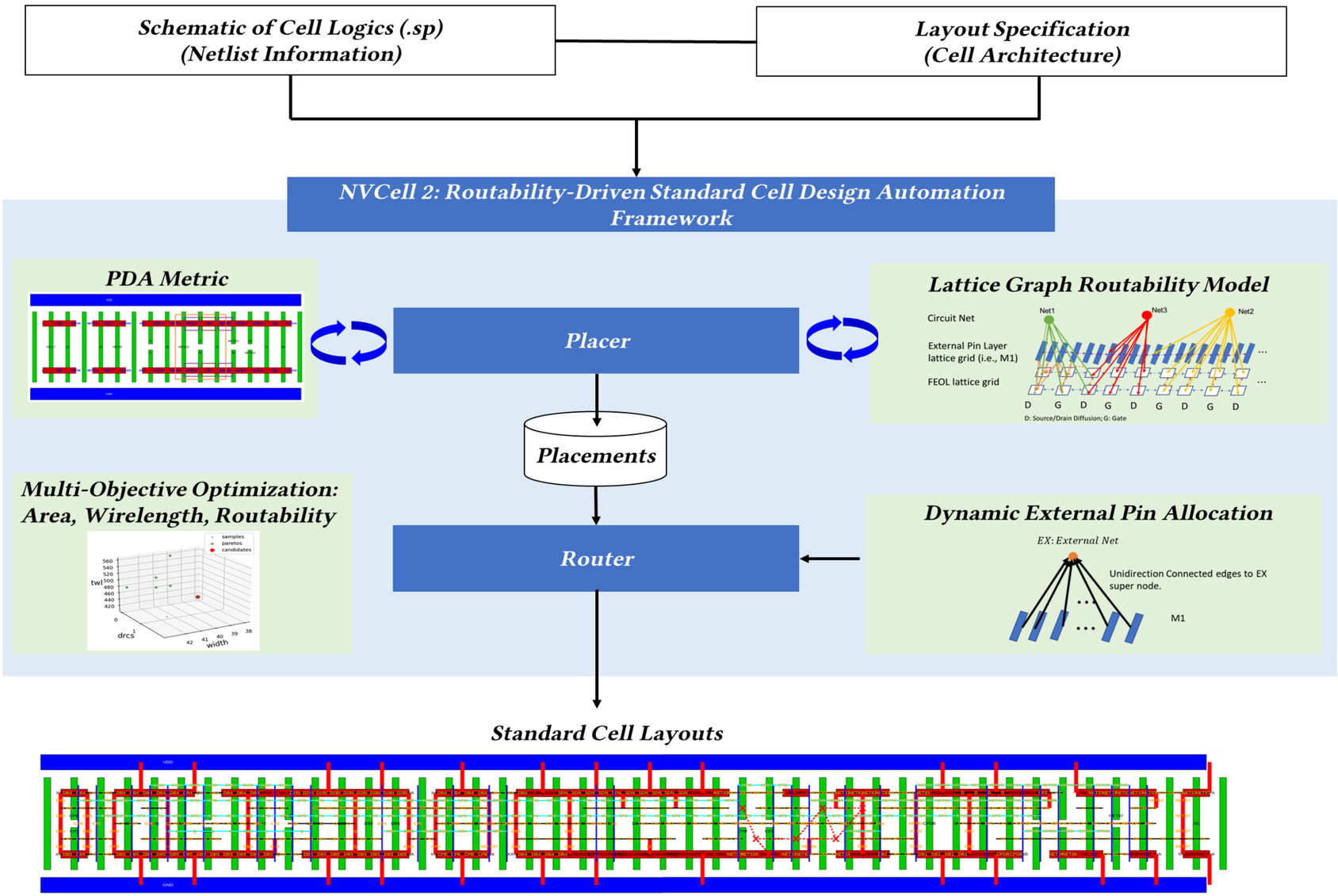
- Leverage genetic algorithm to minimize unroutable nets and DRC numbers

  → solve the optimization problem



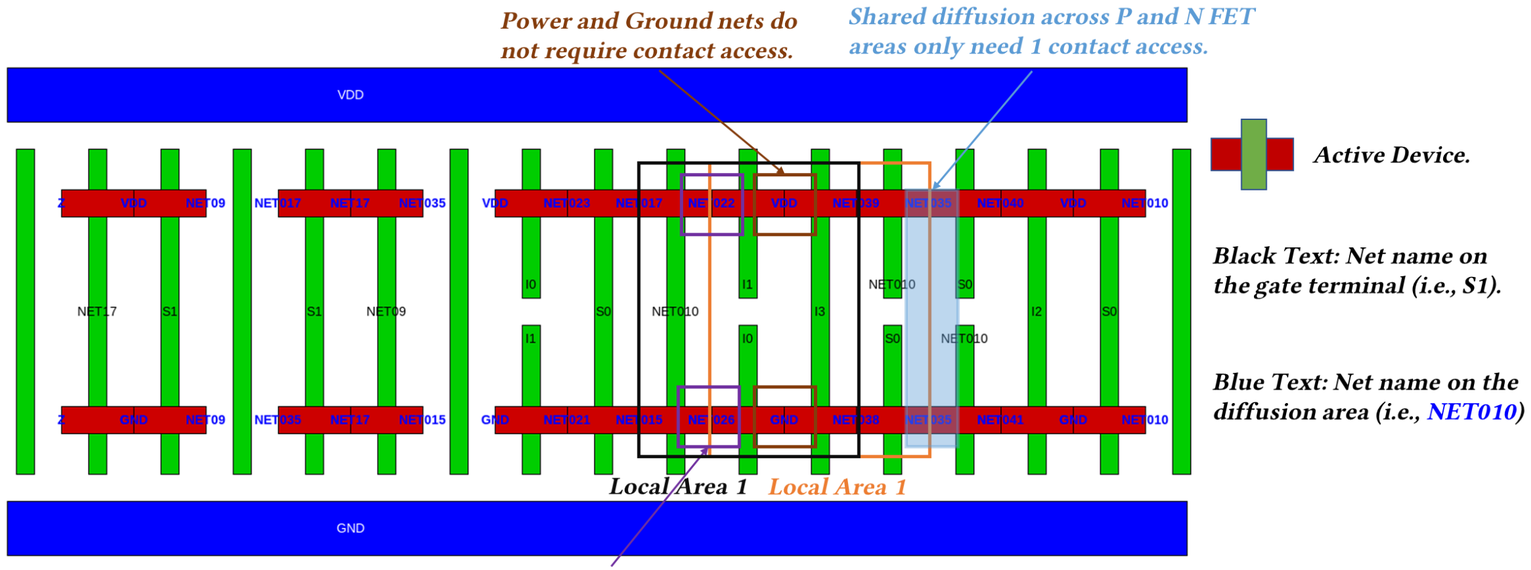
ROUTABILITY-DRIVEN STANDARD CELL AUTOMATION FRAMEWORK

FRAMEWORK OVERVIEW



PIN DENSITY AWARE METRIC

- Calculate the window-based pin density considering diffusion sharing/break, PN gate, source and drain connections.
- Cell-level metric: $P_{cell} = Topk(\frac{\sum_{i=1}^{N_w} PDA_i}{N_w})$

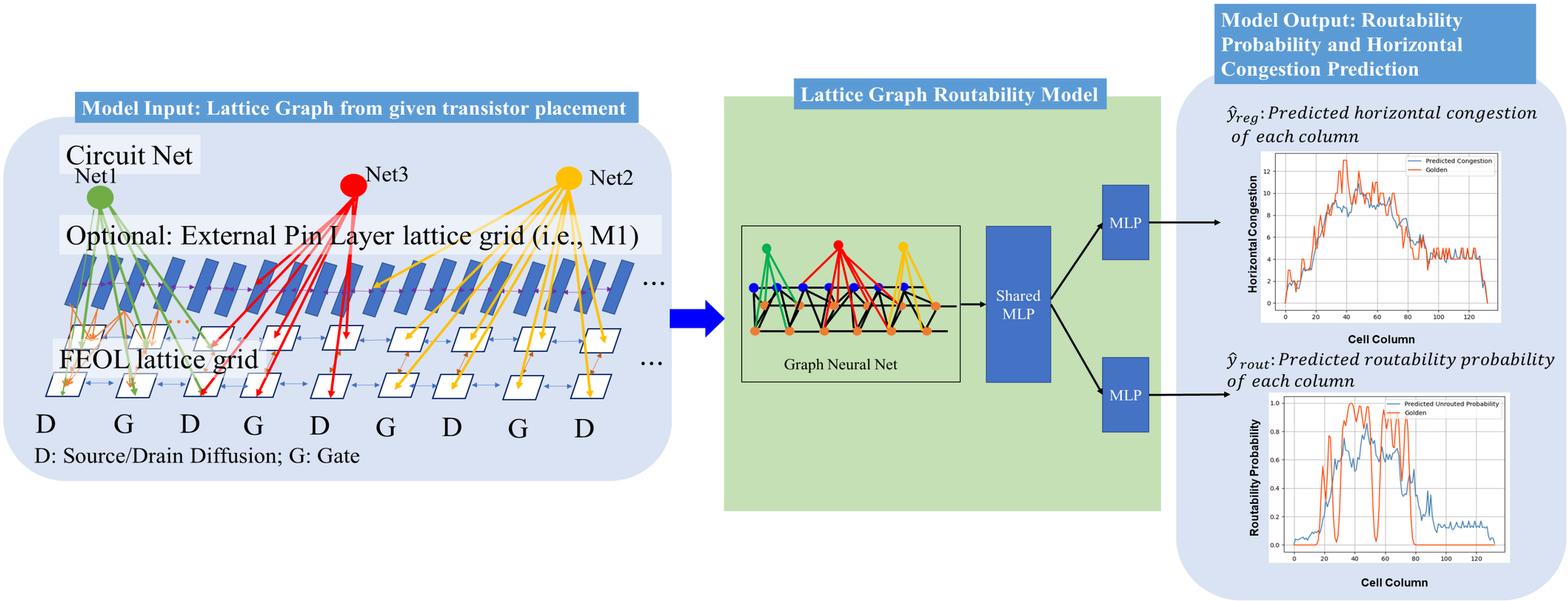


 # Required Contact Access in the Local Area 1 = 6 (i.e., I1, I0, I3, NET010, S0, NET035 nets)
 # Crossing Nets = 0
Pin Density Aware Congestion at Local Area 1 = 6 + 0 = 6

 # Required Contact Access in the Local Area 2 = 6 (i.e., I1, I0, I3, NET010, NET017, NET015 nets)
 # Crossing Nets = 2 (i.e., NET035, S0 nets)
Pin Density Aware Congestion at Local Area 2 = 6 + 2 = 8

LATTICE GRAPH ROUTABILITY MODEL OVERVIEW

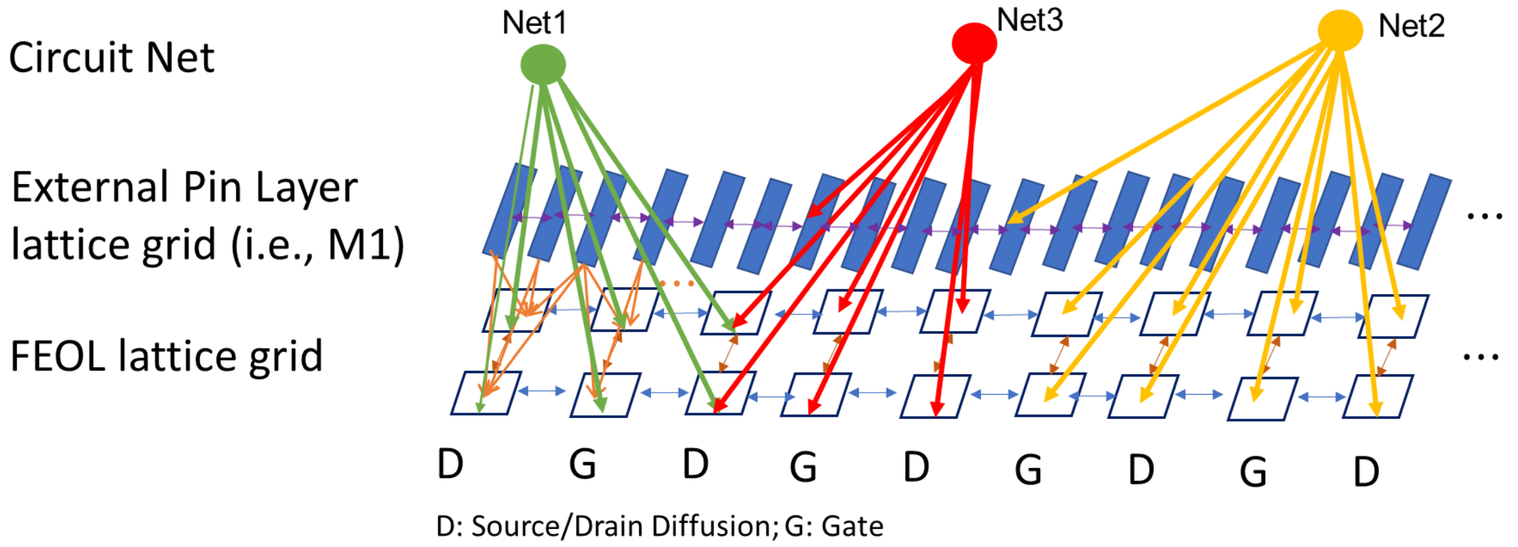
- Given: Circuit, transistor placement, and M1 Pin Placement Information
- Predict: Demanded routing resource and routability probability of each column
  - $\hat{y}_{reg}$: demanded routing resource (hori/vertical) at each column. dim = 1 x cell columns
  - $\hat{y}_{rout}$: routability probability at each column. dim = 1 x cell columns



INPUT FEATURES

Node Types	Node Feature	Edge Types
Circuit Nets	#pins, spanV, spanH	Direct pin assignment, grid link within net bbox
FEOL Lattice Grid	Type of the lattice (G/D), #M0 Access points, #Required contacts	Type of the neighbors (common/split G/D, Diff to Gate)
External Pin Layer Lattice Grid	Type (i.e., I/O Pin), #M0 via connection	Type of the connection (M1 grid to FEOL grid)

Note: #pins, # M0 Access points, and # Required contacts are dynamic based on diffusion sharing/break and PN nets.



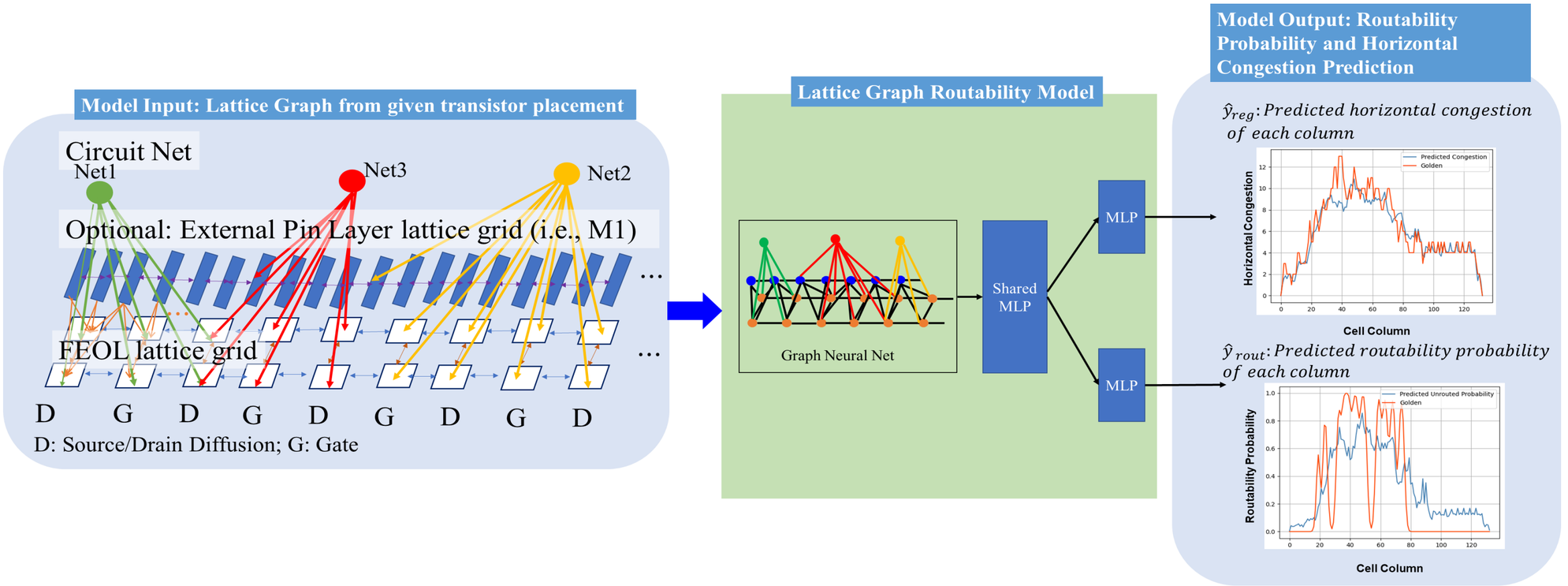
TRAINING LATTICE GRAPH ROUTABILITY MODEL

- Regression Loss Function:

$$L_{reg} = -\frac{1}{N}\sum\left(y_{reg} - \hat{y}_{reg}\right)^2$$

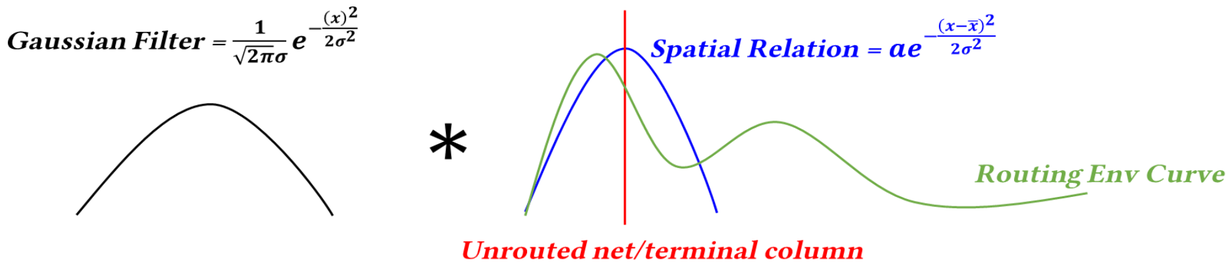
- Routability Probability Loss Function:

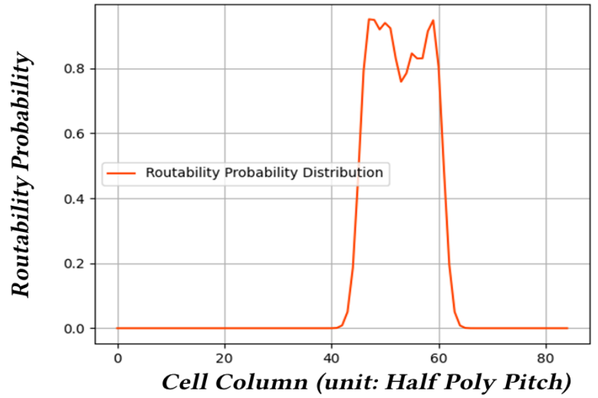
$$L_{rout} = D_{KL}(Y_{rout}||\hat{Y}_{rout}) = Y_{rout} log\frac{Y_{rout}}{\hat{Y}_{rout}}\ ,\ Y_{rout} = Softmax(y_{rout}),\ \hat{Y}_{rout} = Softmax(\hat{y}_{rout})$$

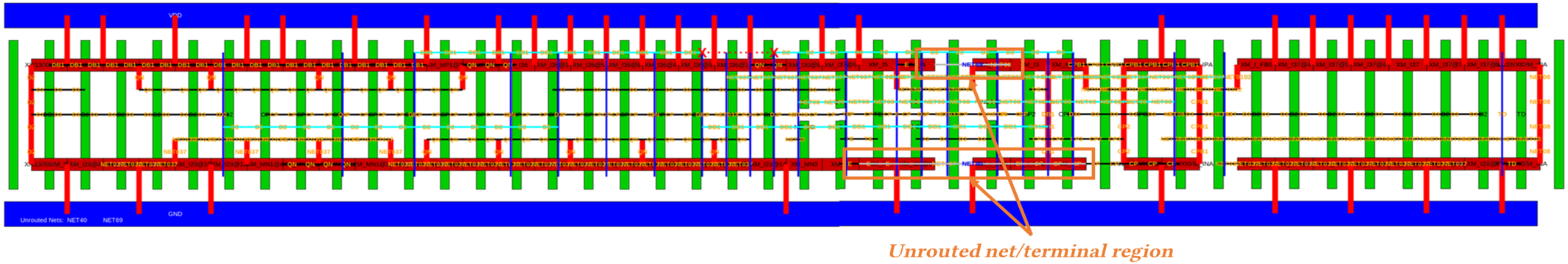


ROUTABILITY PROBABILITY

- Identify the cell columns which caused unrouted net (i.e., pin access and routing resource)
- Consider the surrounding congestion level and routing behavior of the columns with unrouted net
- Gaussian function (Spatial) x congestion ratio curve (Routing env)
- Convolve with gaussian filter with half poly pitch sigma to smooth







CELL-LEVEL ROUTABILITY METRIC

$\hat{y}_{reg}$: *Predicted column based congestion*

$\hat{y}_{rout}$: *Predicted column based unrouted net probability*

x_{pin}: *pin density vector from given placement*

k: *number of columns for routability metric calculation*

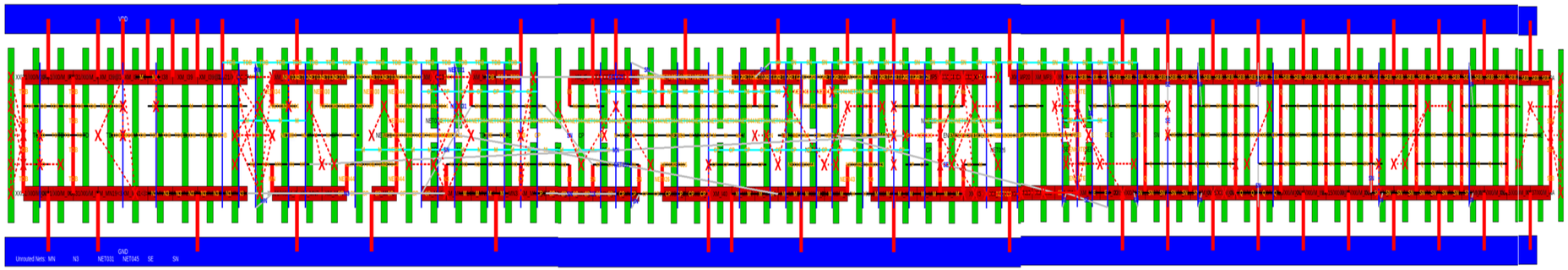
$$PinAccess\ Score = Topk(x_{pin} * \hat{y}_{rout})/k$$

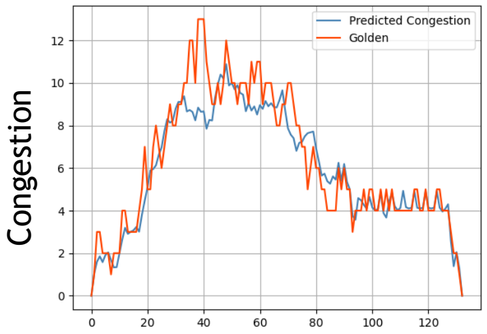
$$\text{Congestion Score} = Topk(\hat{y}_{reg} * \hat{y}_{rout})/k$$

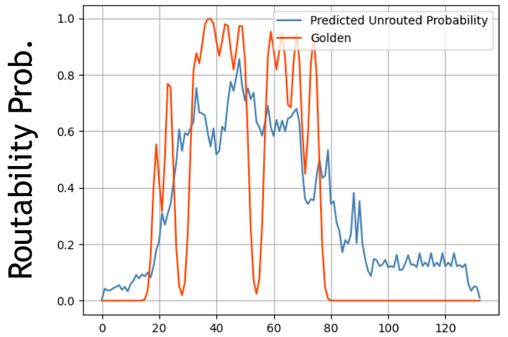
$$\mathbf{R_{cell}} = PinAccess\ Score + \text{Congestion Score}$$

How hard is the transistor pin be accessed?

How hard is the net crossing the region?







ROUTABILITY-DRIVEN PLACEMENT WITH MULTI-OBJECTIVE OPTIMIZATION

- Routability-Driven Placement Objectives

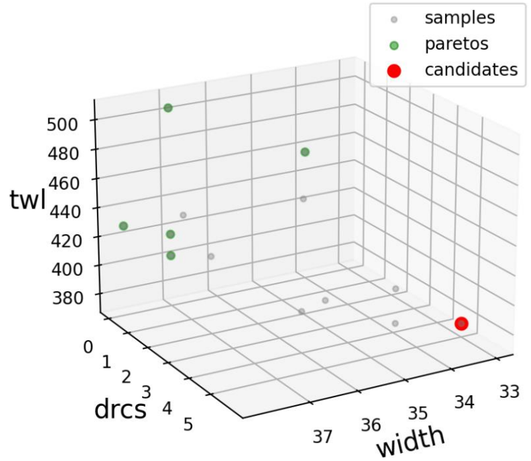
$$\textbf{Minimize } w_a CW + w_m \overline{WL} + w_{pda} P_{cell} + w_{pred} R_{cell}$$

- Multi-objective BOHB [3, 4] - Multiobjective Bayesian optimization algorithm (MOTPE) + Hyperband

Algorithm 1 Optimal Weight Configuration Candidate Selection Algorithm

Input: The data of Multi-Objective BOHB [9], [10] runs, $\mathcal{D}$. The metric axes for pareto extraction (i.e., CW, $drcs$, TWL, etc.) with priority.
Output: The optimal weight configuration candidates, $\mathcal{D}^\star$.

$\mathcal{D}^\star = ExtractPareto(\mathcal{D}, axes)$;
if $\mathcal{D}^\star \neq \emptyset$ **then** ▷ Find the optimal candidates.
  return $\mathcal{D}^\star$;
end if
$\hat{\mathcal{D}} = \emptyset$;
for $axe \in axes$ **do** ▷ Extract pareto of each axe in the $axes$.
  $\hat{\mathcal{D}} = \hat{\mathcal{D}} + ExtractPareto(\mathcal{D}, axe)$;
end for
$cur\_axes = []$;
$\mathcal{D}^\star = \hat{\mathcal{D}}$;
for $axe \in axes$ **do** ▷ Find optimal candidate based on the priority of each axe.
  $cur\_axes.append(axe)$;
  $\mathcal{D}^\star = ExtractPareto(\mathcal{D}^\star, cur\_axes)$;
end for
return $\mathcal{D}^\star$;



An example of a Flip-Flop Design
Baseline width=42
MOBOHB final metrics: width=33 (reduced 21.4%), twl=376

Note: Only show the routable designs in order to display the figure in scale

DYNAMIC EXTERNAL PIN ALLOATION

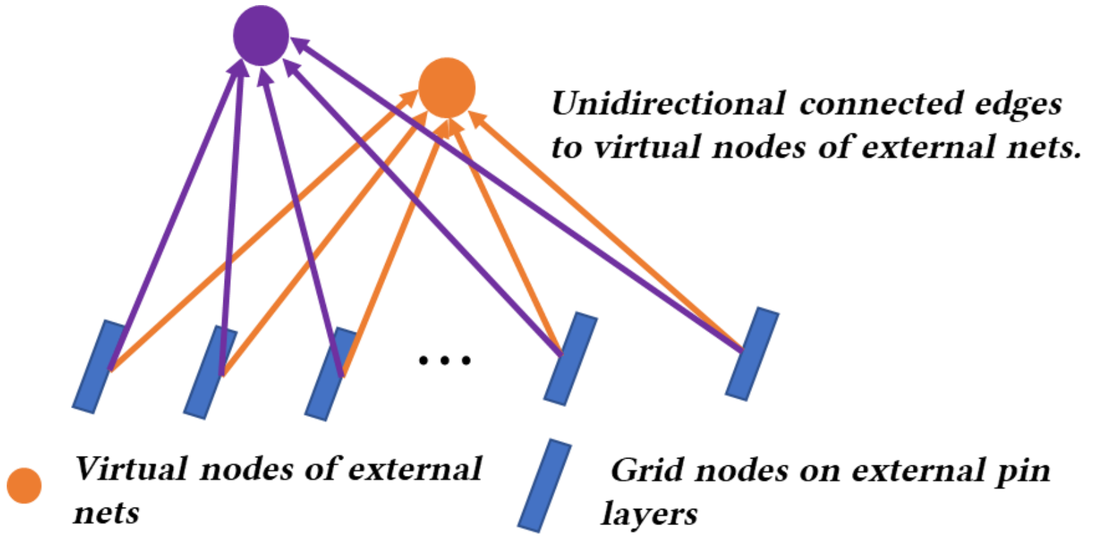
- Router decides the external pin location instead of placer -> Improve the routability and DRC fixing

1. Construct an artificial virtual point, v_n^{vir}, of external nets. (Line 2)
2. Establish the connection from grids on pin metal layers (i.e., M1) to v_n^{vir}. (Line 3 - 5)
3. Add the v_n^{vir} to the routing terminal set. (Line 7 - 10)
4. Perform standard routing algorithms (i.e., Maze routing). (Line 11)

Algorithm 1 Dynamic External Pin Allocation

▷ Input: EX: external nets; $G(V, E)$: 3-dimensional routing graph.

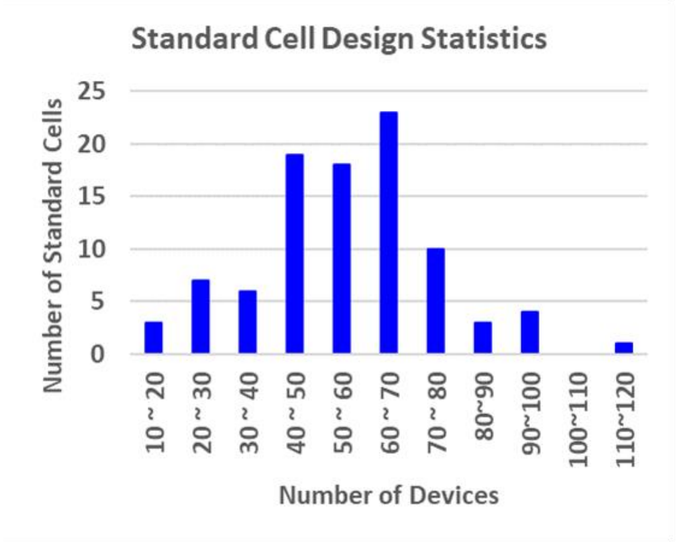
1: **for** $n \in EX$ **do** ▷ Create virtual nodes for external nets
2: Add v_n^{vir};
3: **for** $v \in V_{pinlayers}$ **do** ▷ Create edges between the vertices on the pin layers (i.e., M1)
4: Create a virtual edge from v to v_n^{vir};
5: **end for**
6: **end for**
7: **for** $n \in EX$ **do** ▷ Add virtual nodes to routing terminals
8: Get the routing terminal set of n, T_n;
9: $\overline{T}_n = v_n^{vir} + T_n$;
10: **end for**
11: Perform standard routing algorithms (i.e., Maze router) to route terminals of each net;

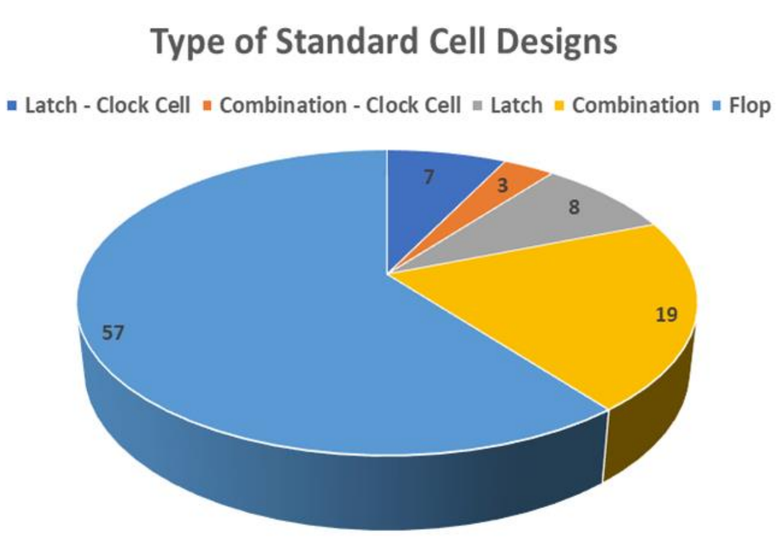


Experimental Results

DESIGNS FOR EXPERIMENTS

- **Experiment I (Routability Metric Accuracy Study):** Validate the proposed routability metrics.
- **Experiment II (Routability Experiment)**: Study the routability improvement of PDA metric, lattice graph routability model, and dynamic pin allocation.
- **Experiment III (Multi-Objective Optimization):** Perform MOBOHB on cell area, total wirelength, and routability to generate optimized cell layouts

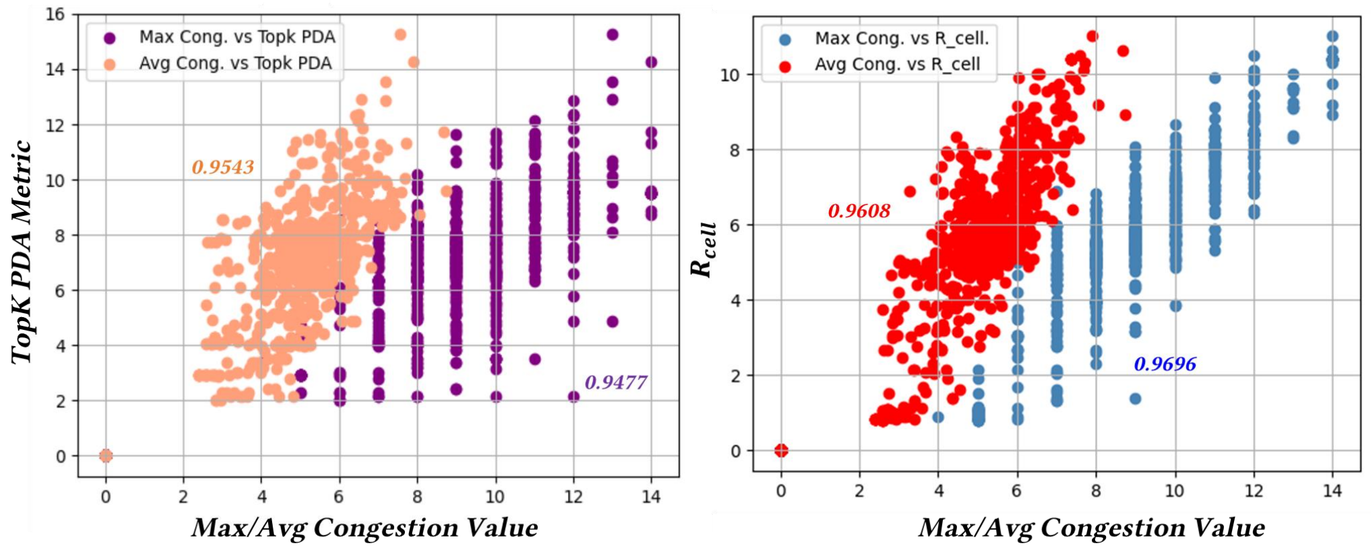




Statistics of 94 complex hard-to-route cells benchmark

EXP I: STRONG CORRELATION WITH GOLDEN CONGESTION

- Extract the golden congestion from the 2240 routed cell layouts
- Cell-level PDA metric (i.e., P_{cell}) achieves 0.9477/0.9543 correlations to Max/Avg congestion
- R_{cell} achieves 0.9696/0.9608 correlations to Max/Avg congestion



EXP I: HIGH CORRELATION WITH AREA OF UNROUTED PROBABILITY DISTRIBUTION

$$PinAccess\ Score = Topk\left(x_{pin} * \hat{y}_{rout}\right)/k$$

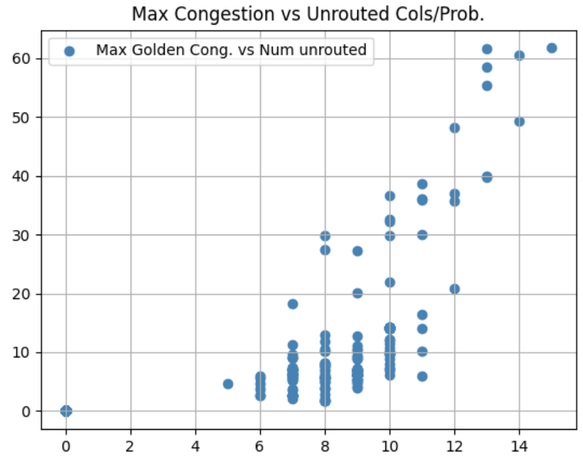
$$\text{Congestion Score} = Topk(\hat{y}_{reg} * \hat{y}_{rout})/k$$

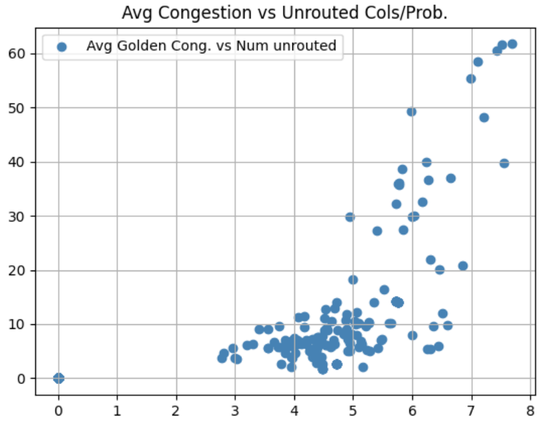
$$\mathbf{R_{cell}} = PinAccess\ Score + \text{Congestion Score}$$

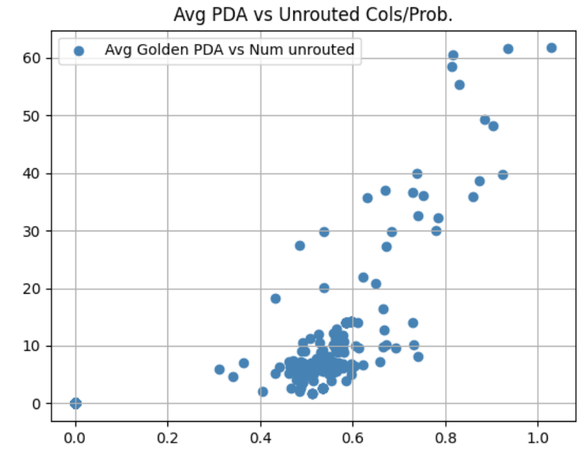
How hard is the transistor pin be accessed?

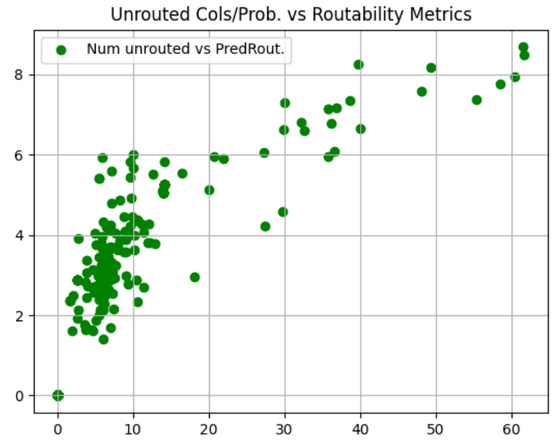
How hard is the net crossing the region?

Metrics	Num Unrouted Prob. Area (Exclude routed designs)
Max Cong.	0.7922
Avg Cong.	0.8030
P_{cell}	0.8364
R_{cell}	**0.8536**

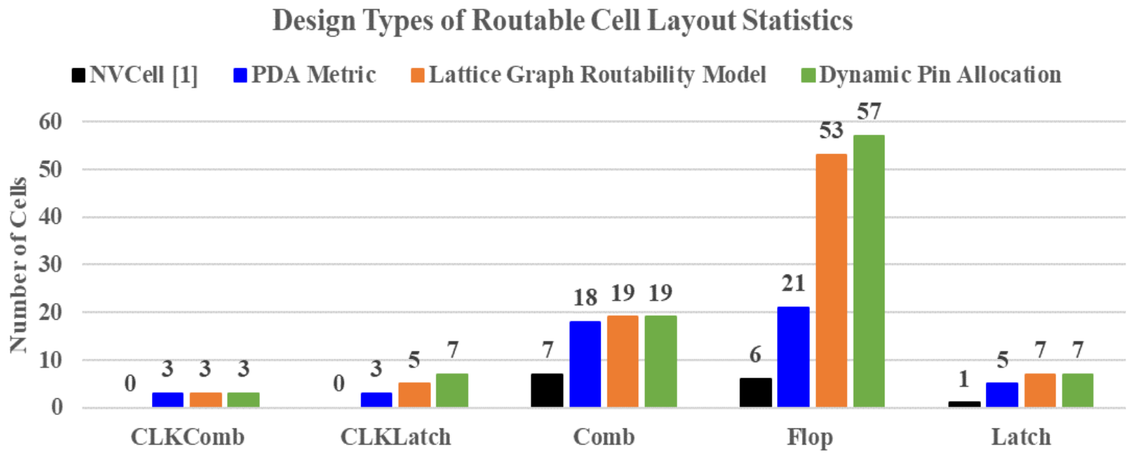


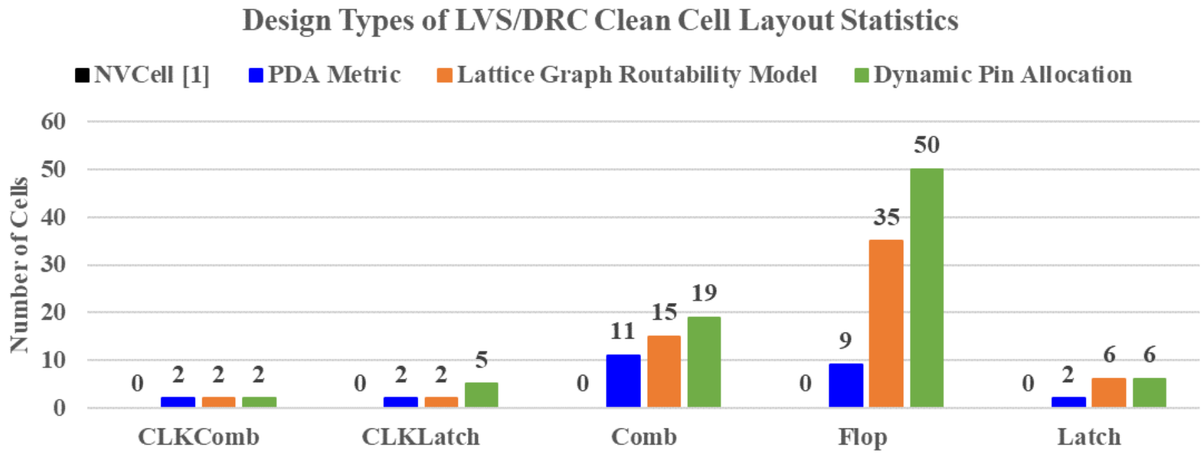






EXP II: ROUTABILITY EXPERIMENT

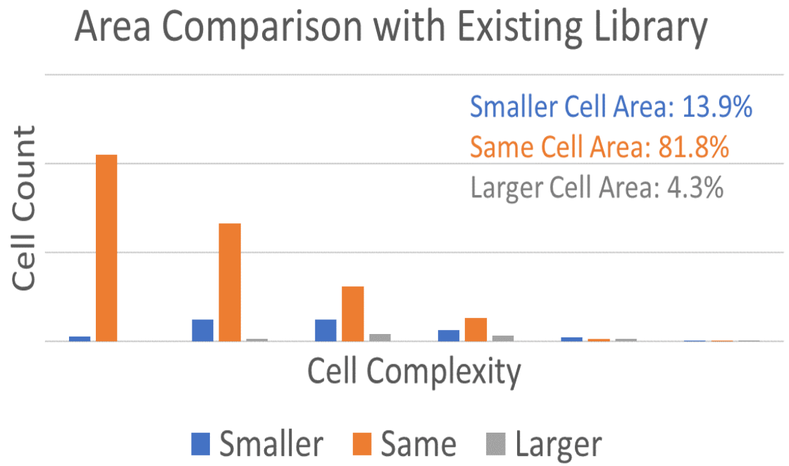




	Routable Cells (%)	*LVS/DRC Clean Cells (%)*
NVCell [1]	***14.9%***	***0.0%***
PDA Metric	***53.2%***	***27.7%***
Lattice Graph Rout. Model	***92.5%***	***63.8%***
Dynamic Pin Allocation	***98.9%***	***87.2%***

EXP III: MULTI-OBJECTIVE OPTIMIZATION

- Optimize the cell area, routability, and total wirelength together.
- Compared to lattice graph routability model, MOBOHB achieves
  - The number of smaller cell layouts: 18.05% impr.
  - The number of larger cell layouts: 54.30% reduction
- 94 hard-to-route cell benchmark.
  - 21 smaller cell width/ 35 same cell width/ 26 larger cell width in total 82 LVS/DRC clean cells.
- Overall, 13.9% smaller and 4.3% larger cell layouts over 1000 standard cells in the existing industrial cell library.



13.9% smaller/ 4.3% Larger (> 1000 cells)

CONCLUSIONS & FUTURE WORKS

CONCLUSIONS & FUTURE WORKS

- We propose a routability-driven standard cell synthesis framework using a novel **pin density aware congestion metric, lattice graph routability modelling approach, and dynamic external pin allocation methodology** to generate optimized layouts to improve the routability of standard cell designs in advanced nodes.
- Improve the routable and LVS/DRC clean cell layouts by 84.0% and 87.2%, respectively, compared to NVCell [1] using the 94 complex and hard to route standard cells.
- Able to generate smaller cell layouts for 13.9% of cells compared to an existing industrial standard cell library over 1000 cells through the MOBOHB process.
- Future Works include
    - Extend current approach for multi-height cell architecture
    - Utilize reinforcement learning technique to improve the performance of standard cell automation.

REFERENCE

[1] Haoxing Ren and Matthew Fojtik. Nvcell: Standard cell layout in advanced technology nodes with reinforcement learning. In 2021 58th ACM/IEEE Design Automation Conference (DAC), pages 1291–1294. IEEE, 2021.

[2] Pascal Van Cleeff, Stefan Hougardy, Jannik Silvanus, and Tobias Werner. Bonncell: Automatic cell layout in the 7-nm era. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(10):2872–2885, 2019.

[3] Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In International Conference on Machine Learning, pages 1437–1446. PMLR, 2018.

[4] Yoshihiko Ozaki, Yuki Tanigaki, Shuhei Watanabe, and Masaki Onishi. Multiobjective tree-structured parzen estimator for computationally expensive optimization problems. In Proceedings of the 2020 genetic and evolutionary computation conference, pages 533–541, 2020.

[5] Ang Lu, Hsueh-Ju Lu, En-Jang Jang, Yu-Po Lin, Chun-Hsiang Hung, Chun-Chih Chuang, and Rung-Bin Lin. Simultaneous transistor pairing and placement for cmos standard cells. In 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), pages 1647–1652. IEEE, 2015.

[6] Yih-Lang Li, Shih-Ting Lin, Shinichi Nishizawa, Hong-Yan Su, Ming-Jie Fong, Oscar Chen, and Hidetoshi Onodera. Nctucell: A dda-aware cell library generator for finfet structure with implicitly adjustable grid map. In Proceedings of the 56th Annual Design Automation Conference 2019, pages 1–6, 2019.

[7] Daeyeal Lee, Dongwon Park, Chia-Tung Ho, Ilgweon Kang, Hayoung Kim, Sicun Gao, Bill Lin, and Chung-Kuan Cheng. Sp&r: Smt-based simultaneous placeand-route for standard cell synthesis of advanced nodes. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 40(10):2142–2155, 2020.

[8] Chung-Kuan Cheng, Chia-Tung Ho, Daeyeal Lee, and Dongwon Park. A routability-driven complimentary-fet (cfet) standard cell synthesis framework using smt. In 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), pages 1–8. IEEE, 2020.

[9] Chung-Kuan Cheng, Chia-Tung Ho, Daeyeal Lee, Bill Lin, and Dongwon 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, 29(6):1178–1191, 2021.

[10] Chung-Kuan Cheng, Chia-Tung Ho, Daeyeal Lee, and Bill Lin. Multirow complementary-fet (cfet) standard cell synthesis framework using satisfiability modulo theories (smts). IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, 7(1):43–51, 2021.

[11] Vincent A Cicirello. On the design of an adaptive simulated annealing algorithm. In Proceedings of the international conference on principles and practice of constraint programming first workshop on autonomous search, 2007.

[12] Haoxing Ren and Matthew Fojtik. Standard cell routing with reinforcement learning and genetic algorithm in advanced technology nodes. In Proceedings of the 26th Asia and South Pacific Design Automation Conference, pages 684–689, 2021.

[13] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. The Journal of Machine Learning Research, 18(1):6765–6816, 2017.

THANK YOU