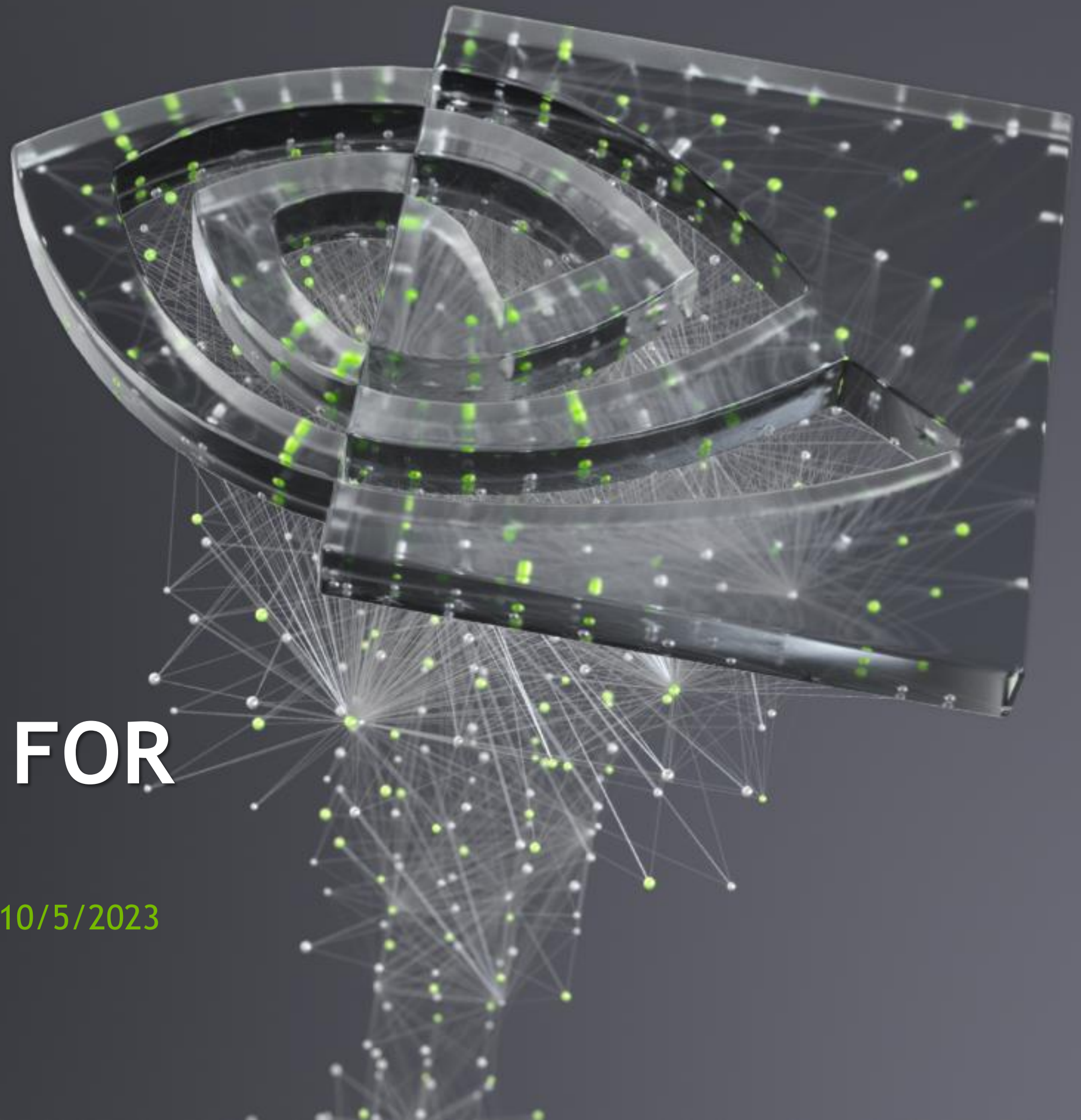




MACHINE LEARNING FOR EDA OPTIMIZATION

Chia-Tung (Mark) Ho, NVResearch ASIC/VLSI group, 10/5/2023





AGENDA

Background & Motivation

Machine Learning for EDA Optimization

Conclusions



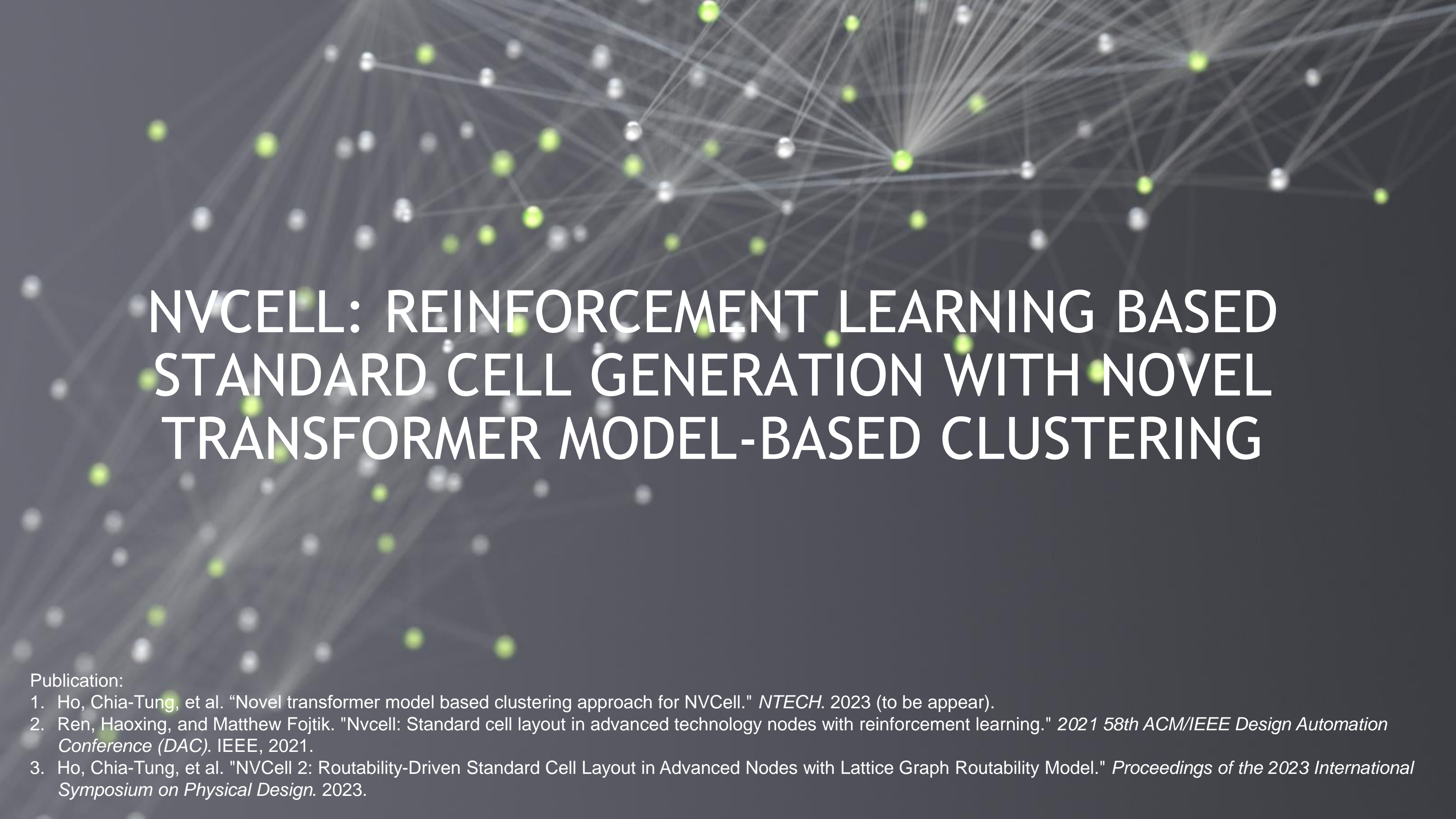
**BACKGROUND &
MOTIVATION**

BACKGROUND & MOTIVATION

- Optimization is one of the fundamental problems in EDA
- Goal: Improve power, performance, area, and cost (PPAC)
$$\text{Minimize } f(x_1, x_2, \dots, x_n)$$
$$\text{Subject to } g(x_1, x_2, \dots, x_n)$$
- The functions might be non-linear, non-convex, and discrete
- Design challenges of modern circuit design in advanced nodes
 - billions of transistors
 - Increasing number and complexity of design rules
 - Routability
 - Strict pattern rules
- ML opportunities: Improve the productivity, efficiency, and quality



MACHINE LEARNING FOR EDA OPTIMIZATION



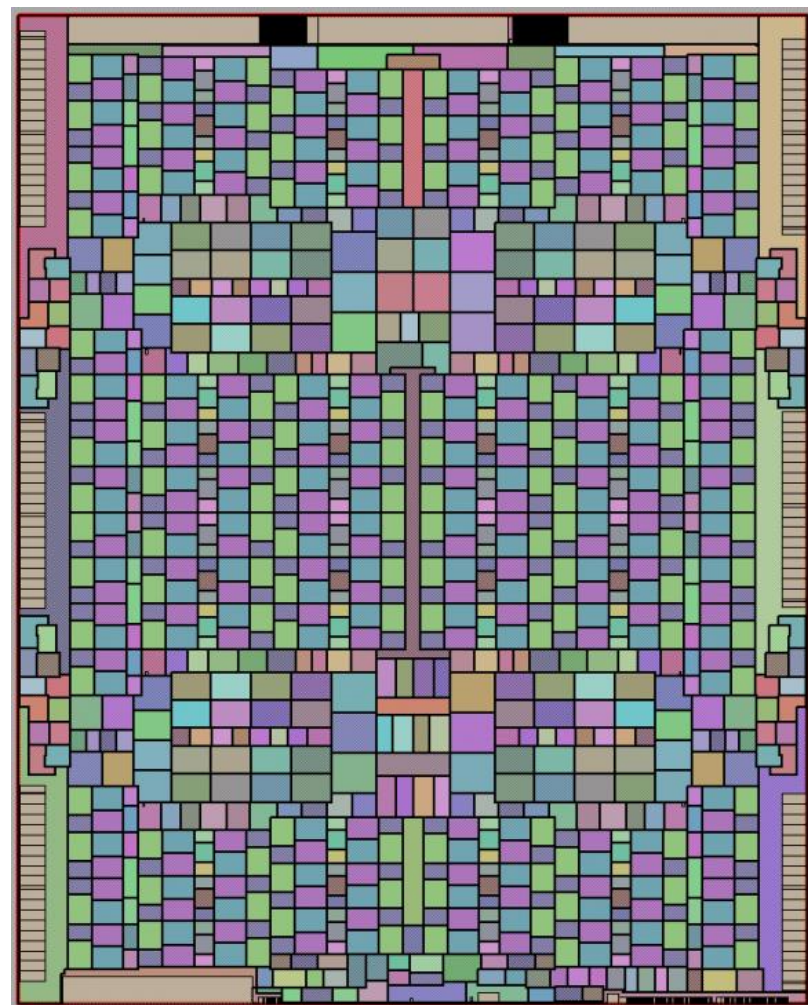
NVCELL: REINFORCEMENT LEARNING BASED STANDARD CELL GENERATION WITH NOVEL TRANSFORMER MODEL-BASED CLUSTERING

Publication:

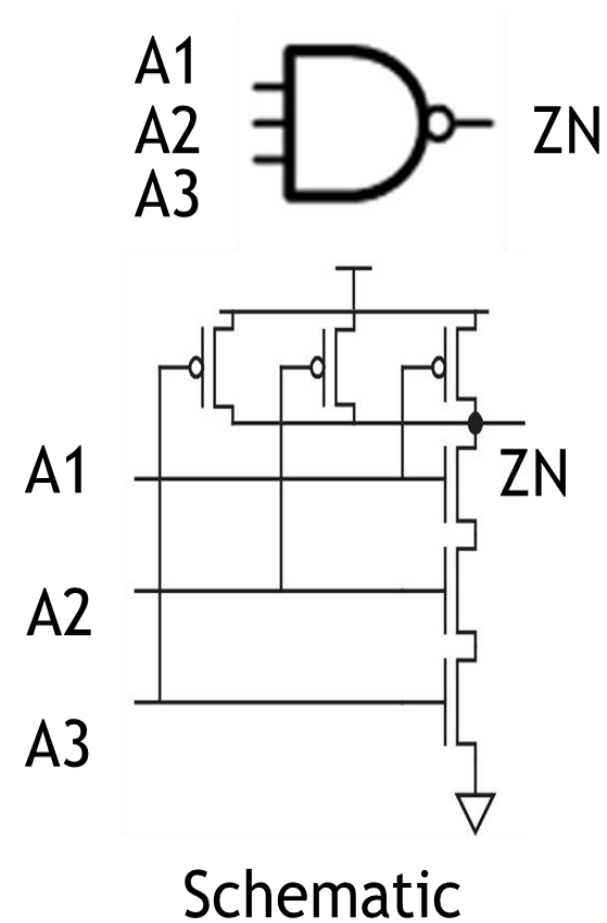
1. Ho, Chia-Tung, et al. "Novel transformer model based clustering approach for NVCell." *NTECH*. 2023 (to be appear).
2. Ren, Haoxing, and Matthew Fojtik. "Nvcell: Standard cell layout in advanced technology nodes with reinforcement learning." *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021.
3. Ho, Chia-Tung, et al. "NVCell 2: Routability-Driven Standard Cell Layout in Advanced Nodes with Lattice Graph Routability Model." *Proceedings of the 2023 International Symposium on Physical Design*. 2023.

STANDARD CELL LAYOUT AUTOMATION

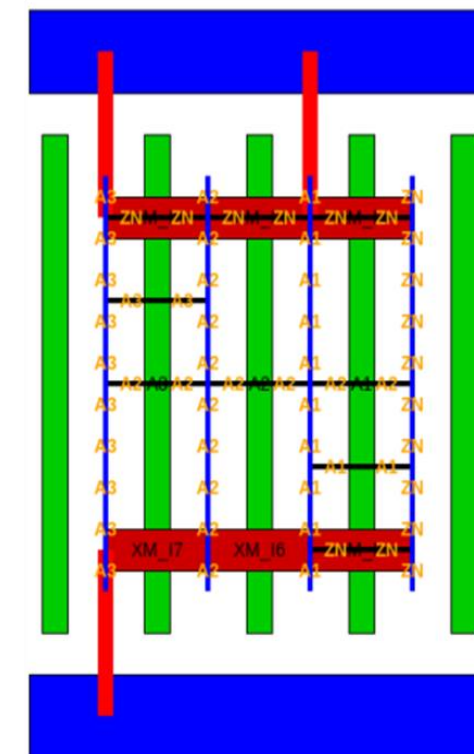
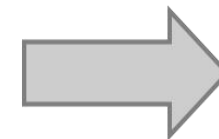
- Std cells are building blocks of digital design layout: AND, NOR, Flip-Flop, Adder, etc
- Layout mostly by hand today, long design turn around time for the library (a few months)
- Standard cell automatic layouts - [Fast design turn around time, More custom cell design, Design Technology Co-Optimization](#)



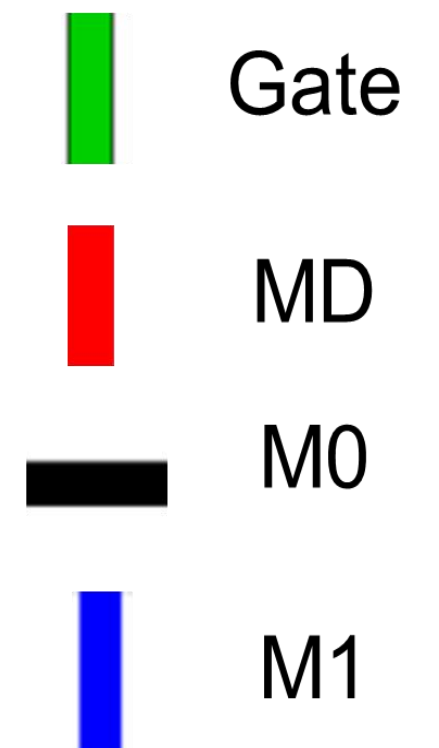
GA100 - 1.7B standard cells



Schematic



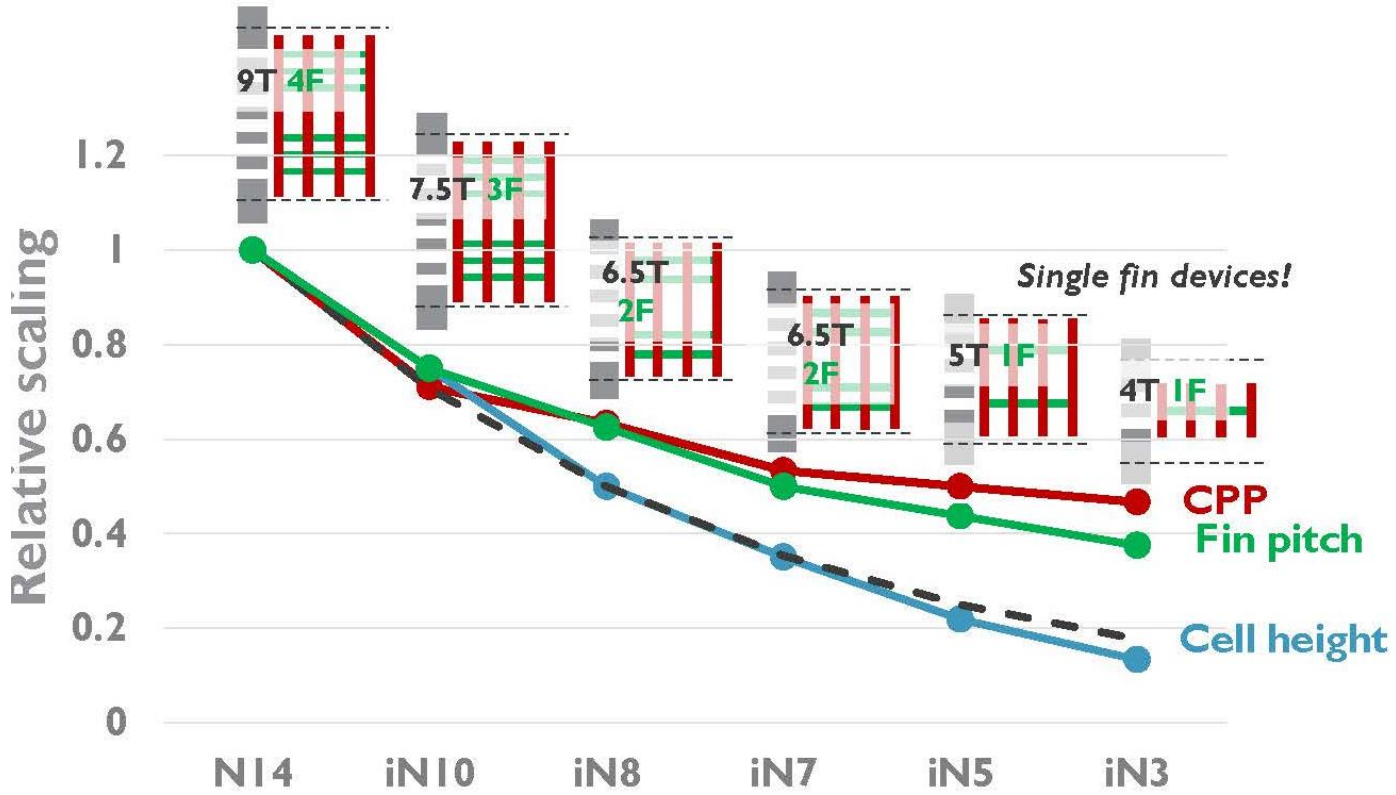
Simplified Grid-based layout diagram (Sticks)



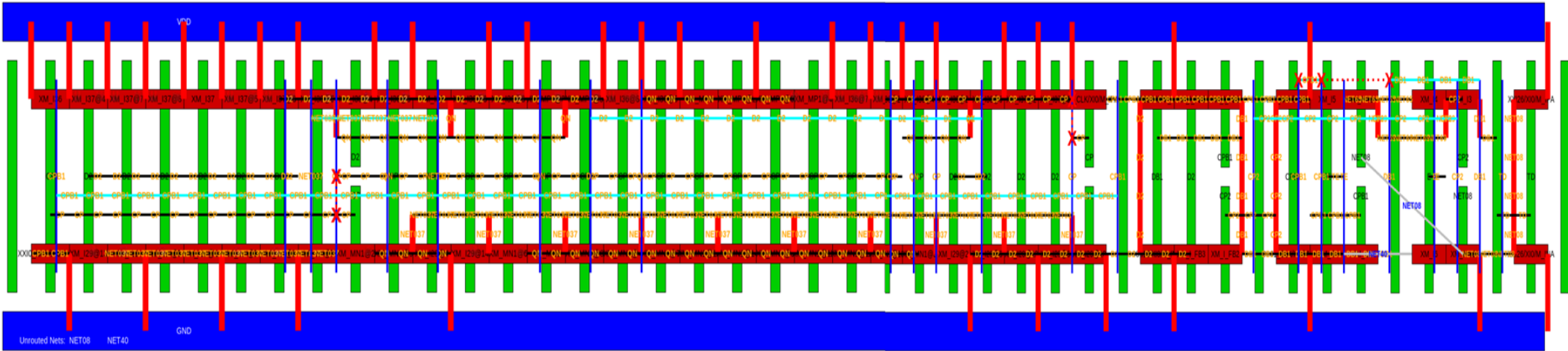
Standard ell

ROUTABILITY AND PPA-DRIVEN STANDARD CELL DESIGN AUTOMATION

- Standard cell layout design automation challenges as advancing beyond 5nm
 - Limited in-cell routing resource - less routing tracks (i.e., < 5 RTs)
 - Design rule complexity: Increasing number and complexity of design rules + strict patterning rules
 - Scalability: > hundreds of transistors cell designs
- Multi-Objective Optimization: Scalability, Routability, and high quality on PPA metrics



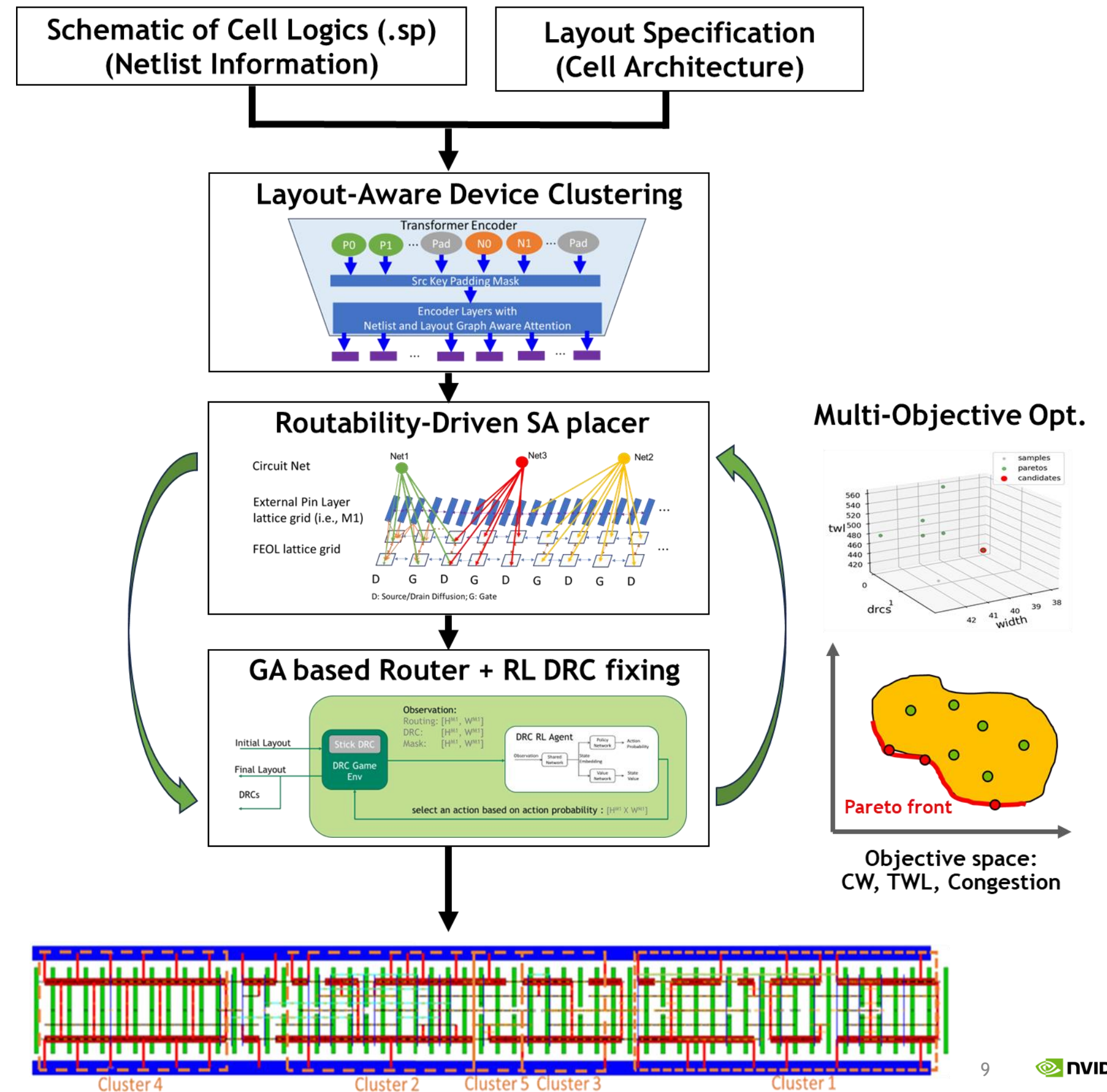
Standard Cell Scaling Roadmap from IMEC
 Source: <https://www.imec-int.com/en/imec-magazine/imec-magazine-november-2018/the-supervia-a-promising-scaling-booster-for-the-sub-3nm-technology-node>



Routability challenges of a Latch Design in advanced node

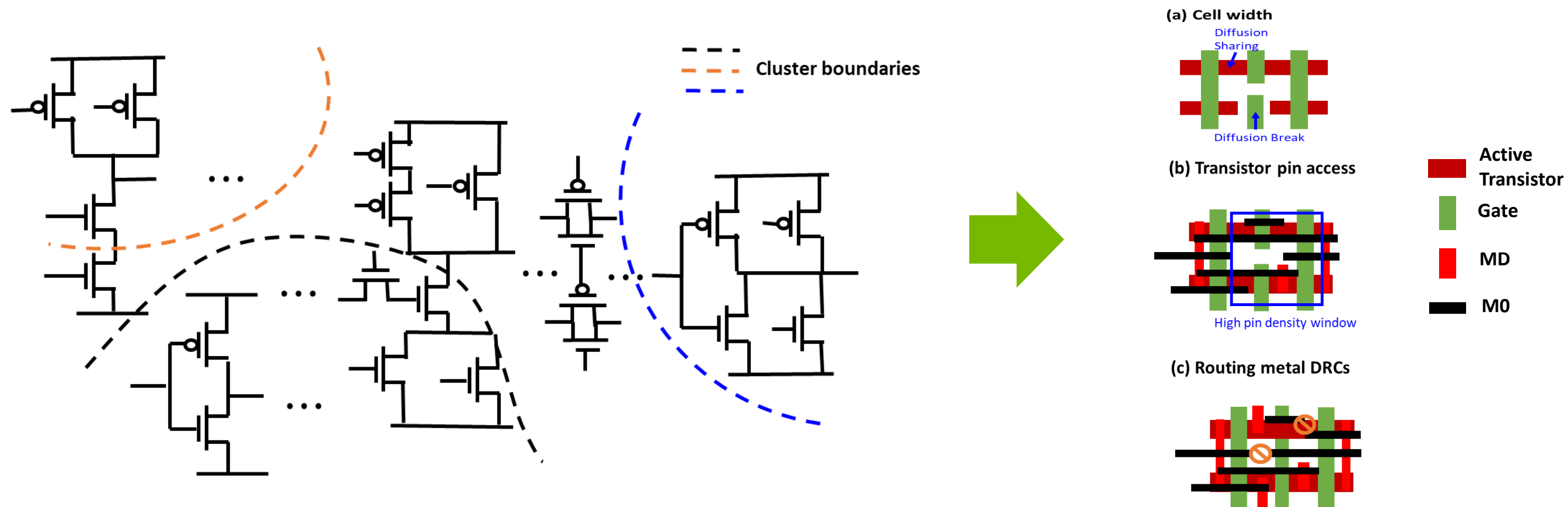
NVCELL: STANDARD CELL DESIGN AUTOMATION FRAMEWORK

- Layout-Aware Transformer Model Based Device Clustering
 - High-Quality clustering to reduce complexity, narrow down searching space, and assist finding routable solutions
- Lattice graph routability model in SA placer
 - Capture the local pin density and global net connections
- Reinforcement learning agent for DRC fixing
 - Model DRC fixing as a game to improve productivity and efficiency



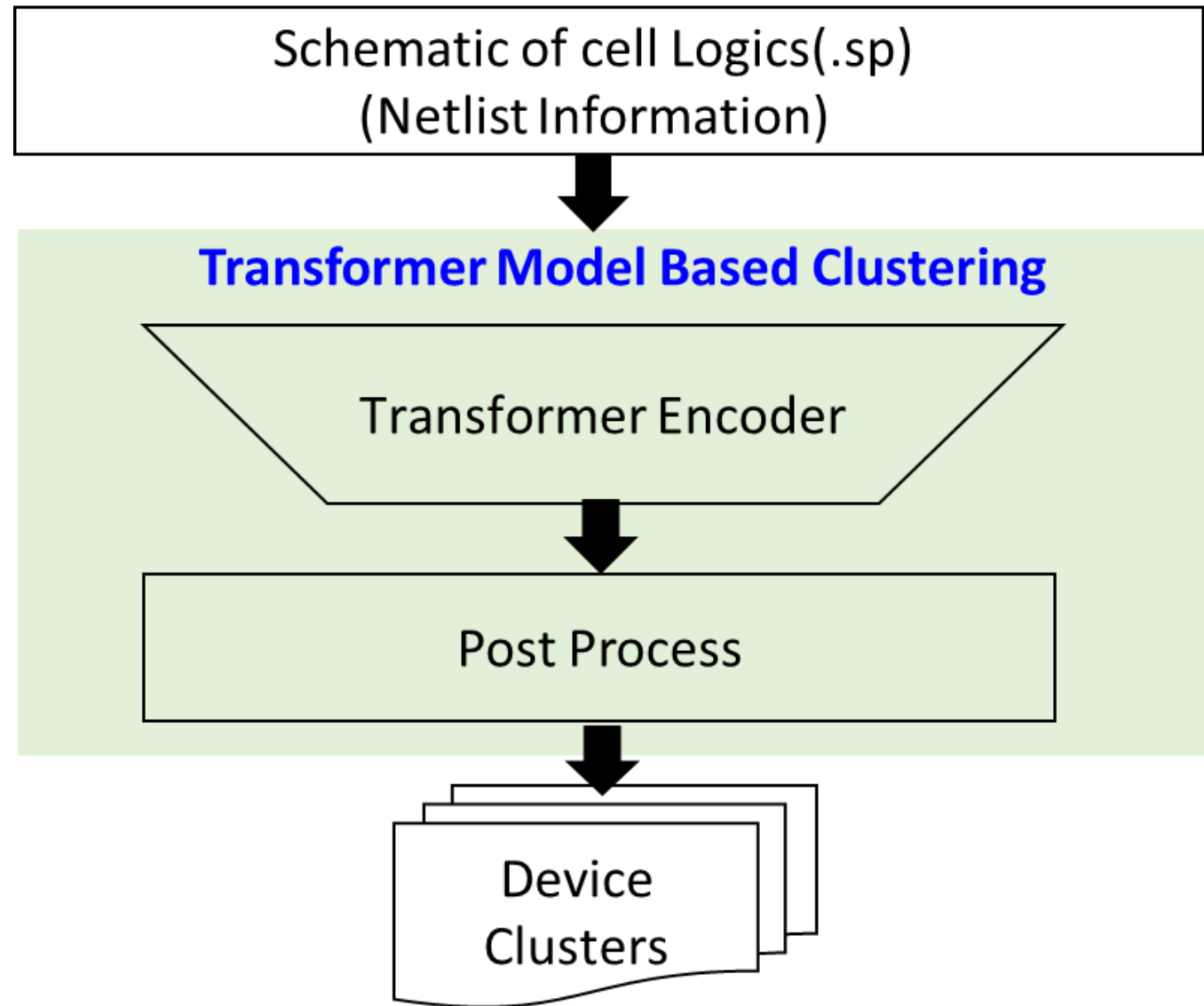
LAYOUT-AWARE DEVICE CLUSTERING

- High quality clustering should consider transistor layout: Diffusion break/sharing, Transistor pin access, and Routing metal DRCs
- Reduce complexity, Narrow down searching space, and Assist finding routable layouts
- ▶ -> Transformer model-based clustering approach



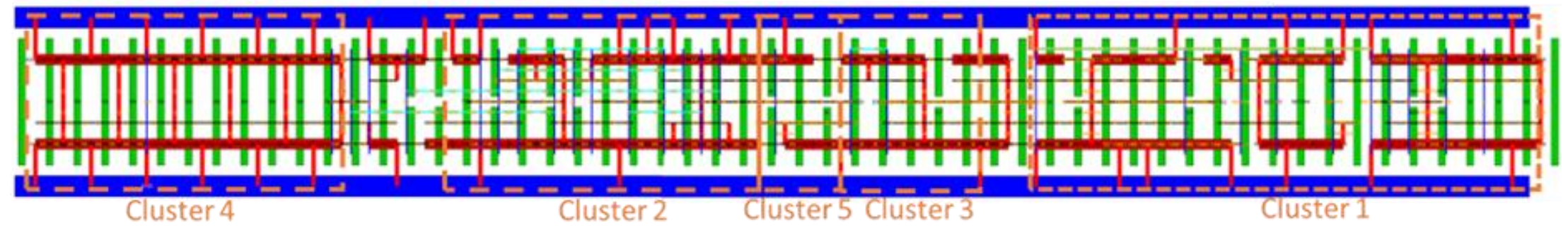
Global receptive field, netlist information, and device placement relations

LAYOUT-AWARE DEVICE CLUSTERING

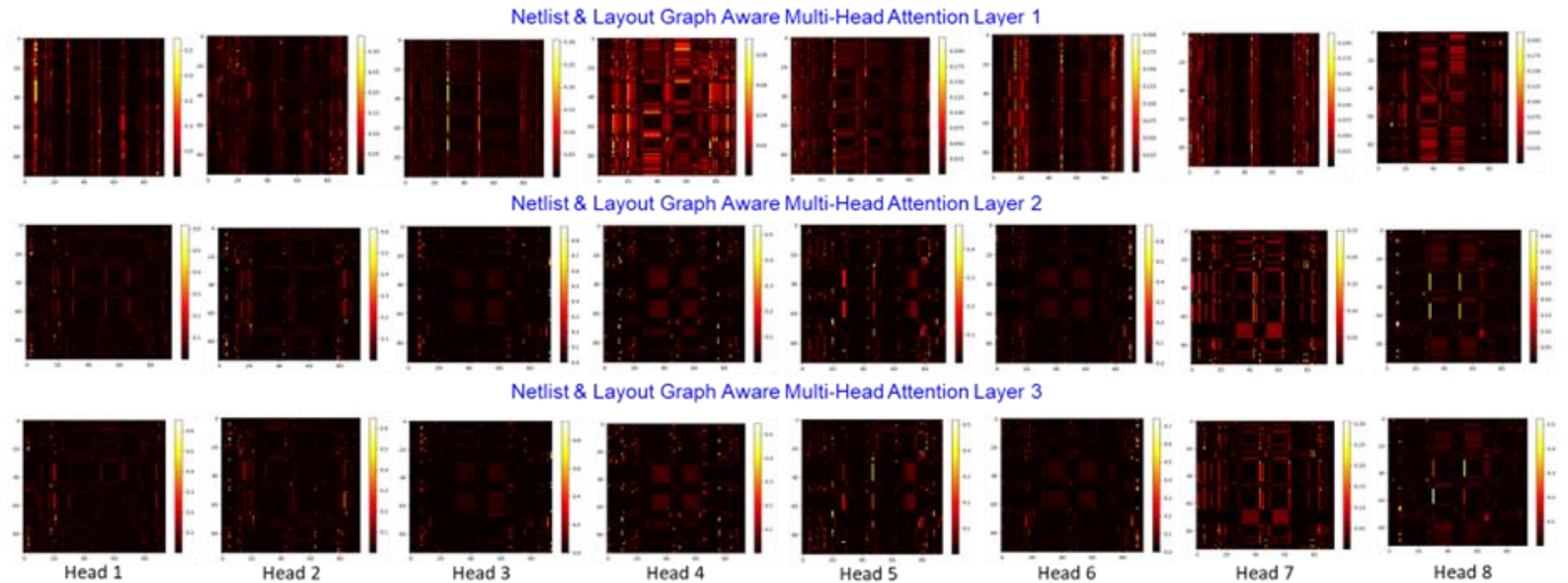


Generated LVS/DRC Clean Latch Design (~ 100 devices)

Manual Cell Width = 58 / Generated Cell Width = 56 TWL = 671



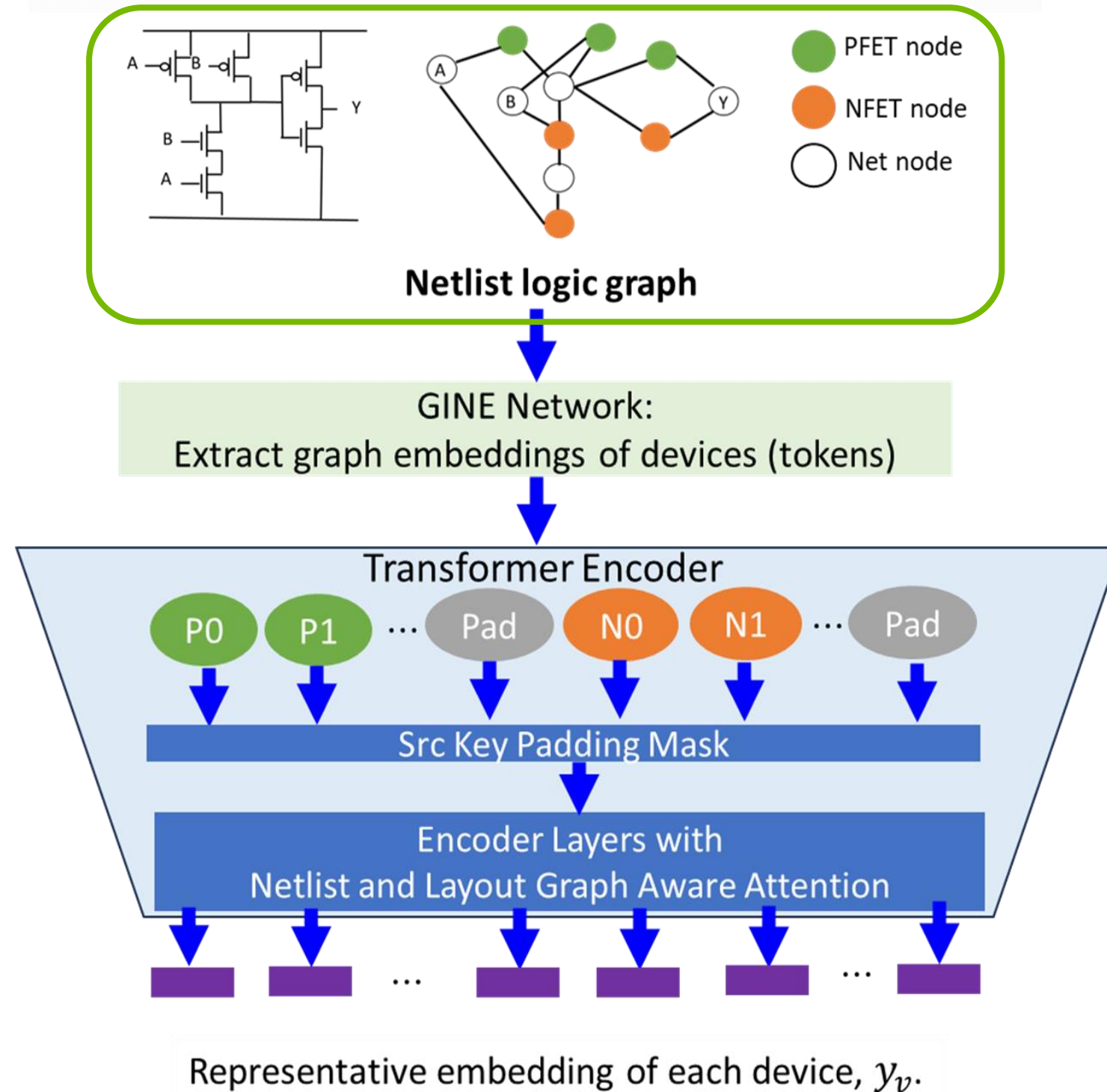
Attention heat map of the Generated LVS/DRC Clean Latch Design (~ 100 devices)



TRANSFORMER ENCODER ARCHITECTURE

Goal: Given netlist logic graph, learn the relationship between device pairs in the Layout graph

(a) Transformer Encoder Architecture

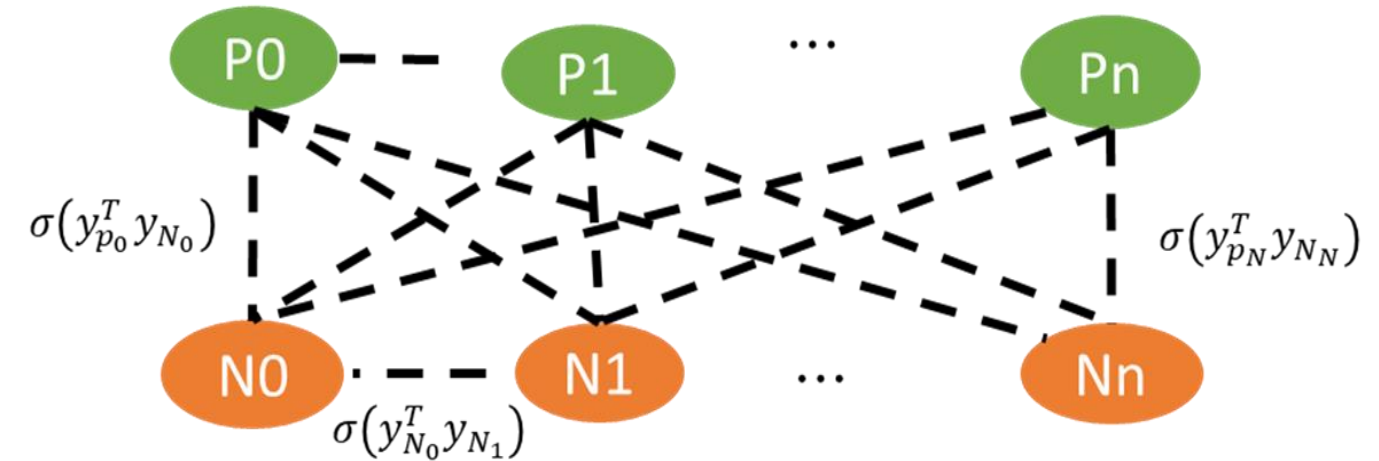


(b) Training Flow: Similarity loss (L_{sim}) from layout graph

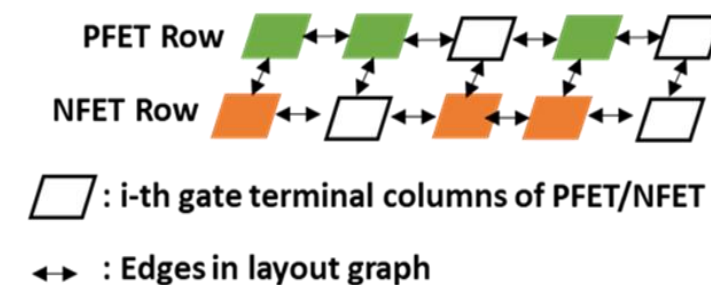
$$L_{sim} = \sum_v \left(- \sum_{u \in N_l(v)} \log(\sigma(y_v^T y_u)) - \sum_{k \sim rand} \log(\sigma(-y_v^T y_k)) \right)$$

$\sigma(y_v^T y_u)$: Preferred clustering probability of two devices.

$N_l(v)$: The neighbor of device v in the layouts

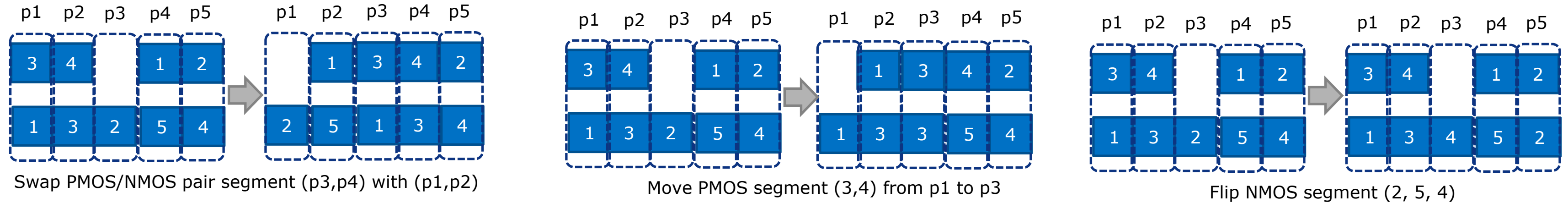


Layout graph (Neighbor columns all connected)



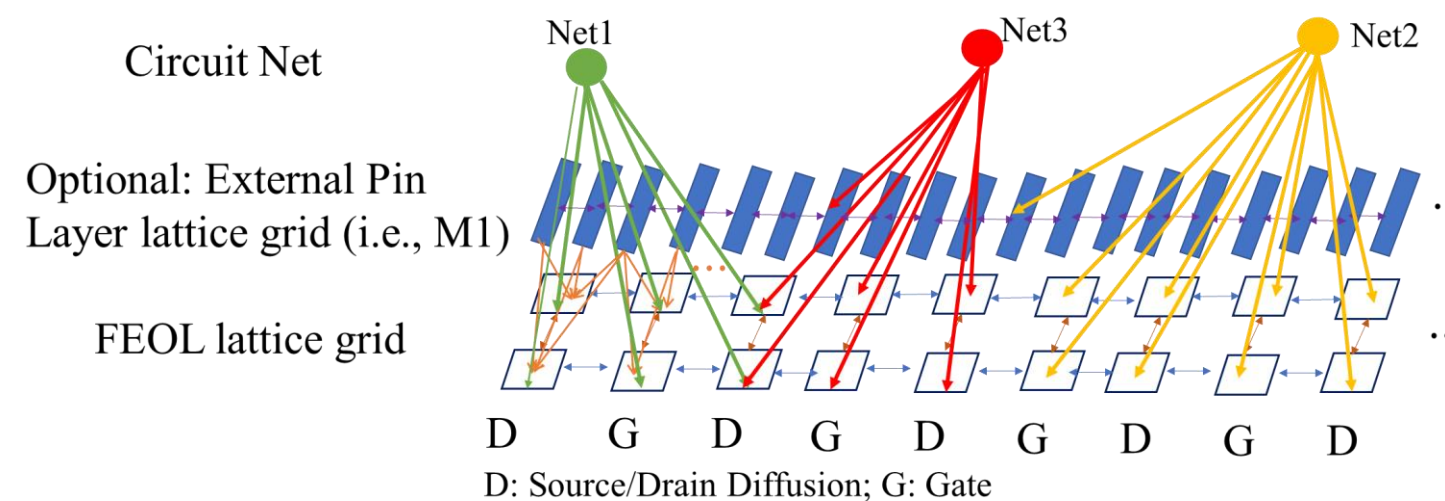
ROUTABILITY-DRIVEN PLACEMENT

- Simulated Annealing based algorithm for placement: Swap, Move, Flip



Swap, move, and flip of placement sequence

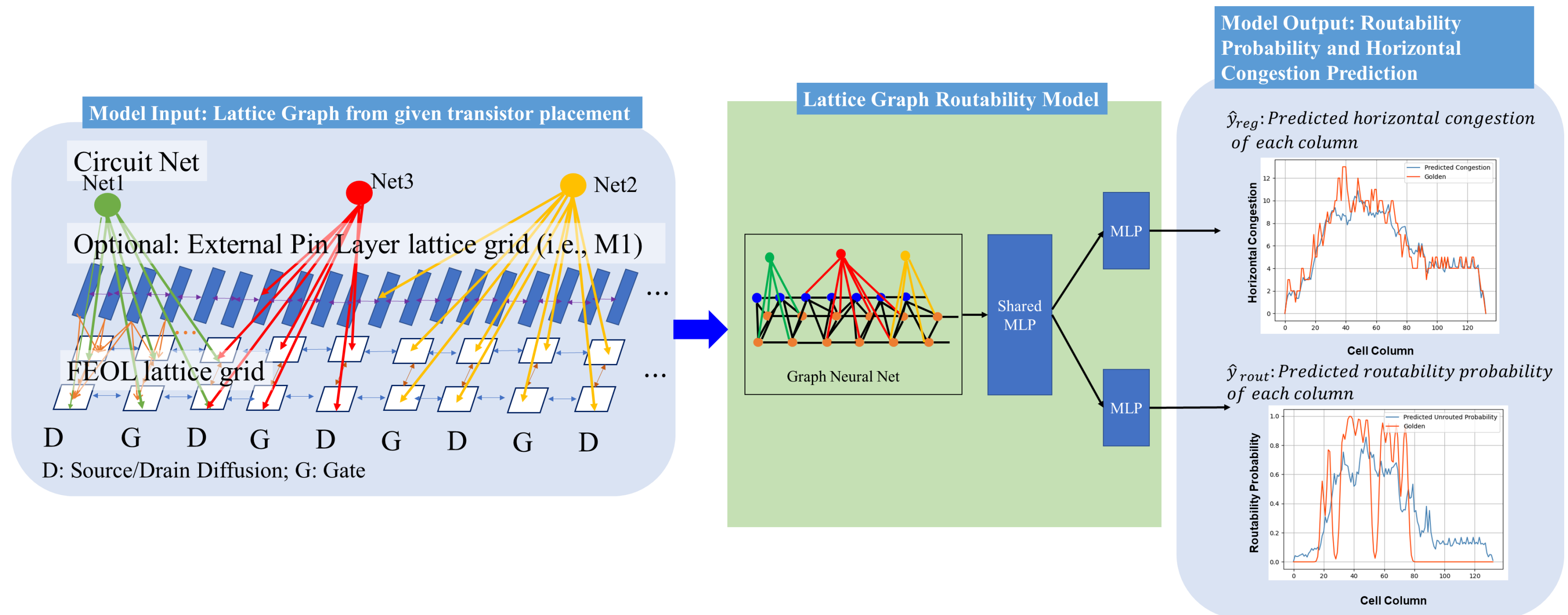
- Routability Model: Lattice graph routability model predicts congestion and routability probability
 - Capture the interactions of local pin access and global nets given the placement



Lattice graph routability model

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



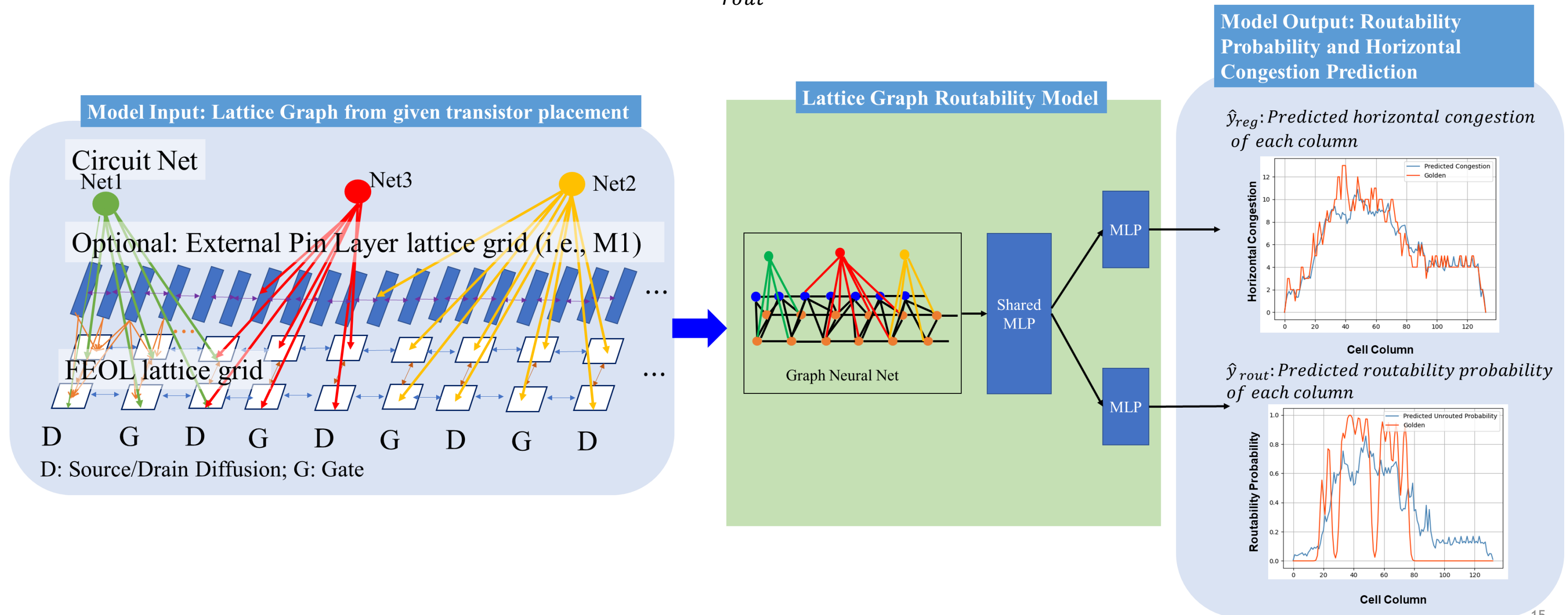
TRAINING LATTICE GRAPH ROUTABILITY MODEL

- Regression Loss Function:

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

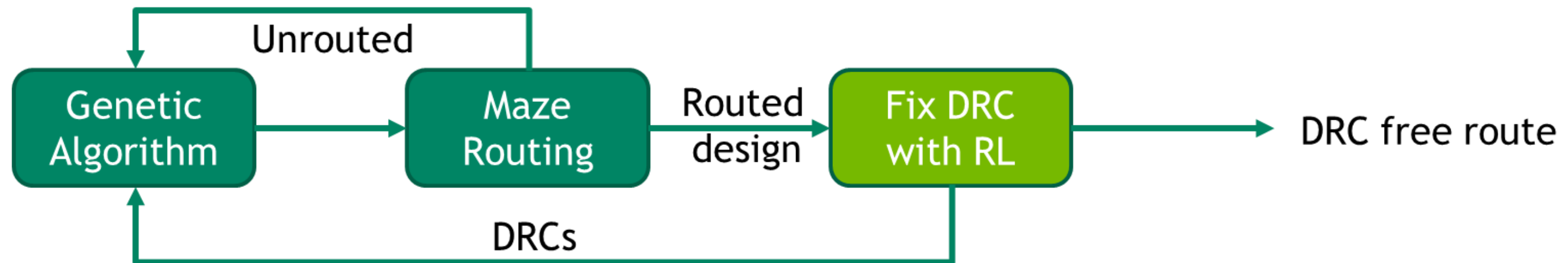
- Routability Probability Loss Function:

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



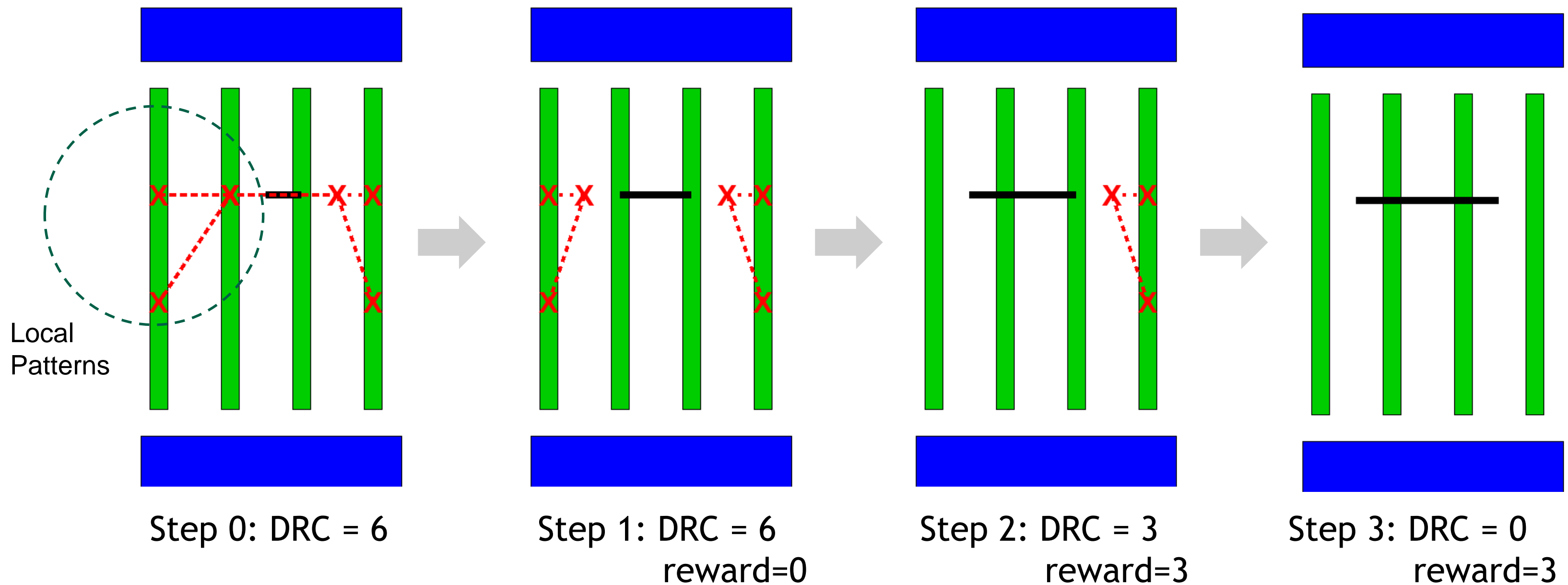
ROUTING

- 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

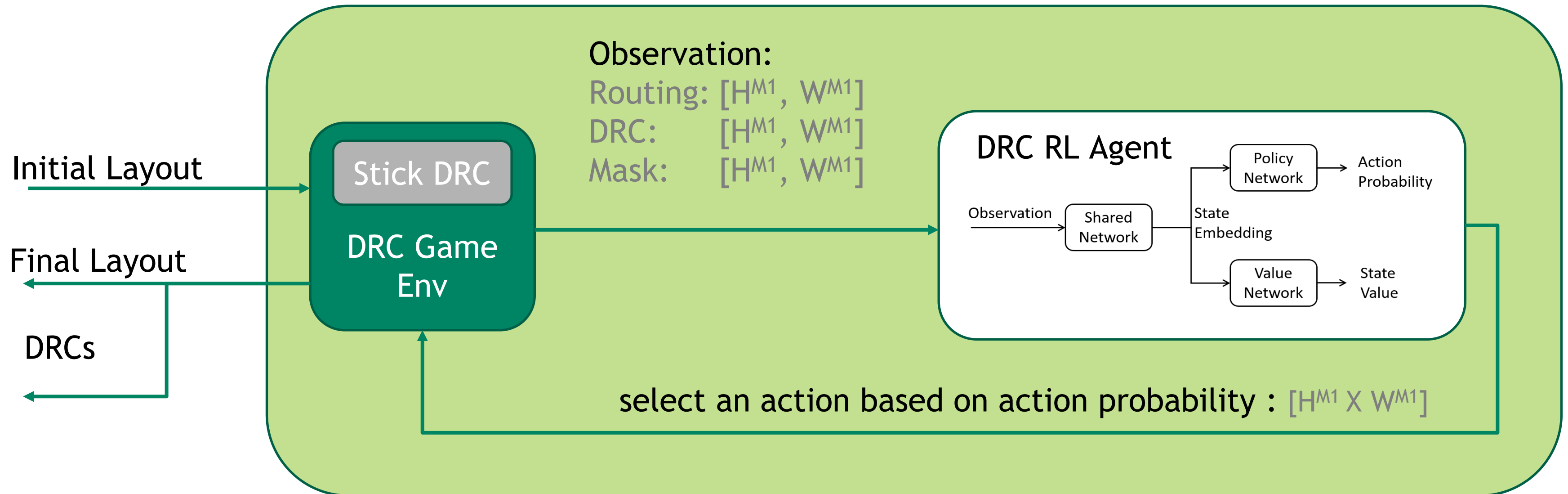


GAME OF FIXING DRC

Adding additional M0 grid to reduce DRCs

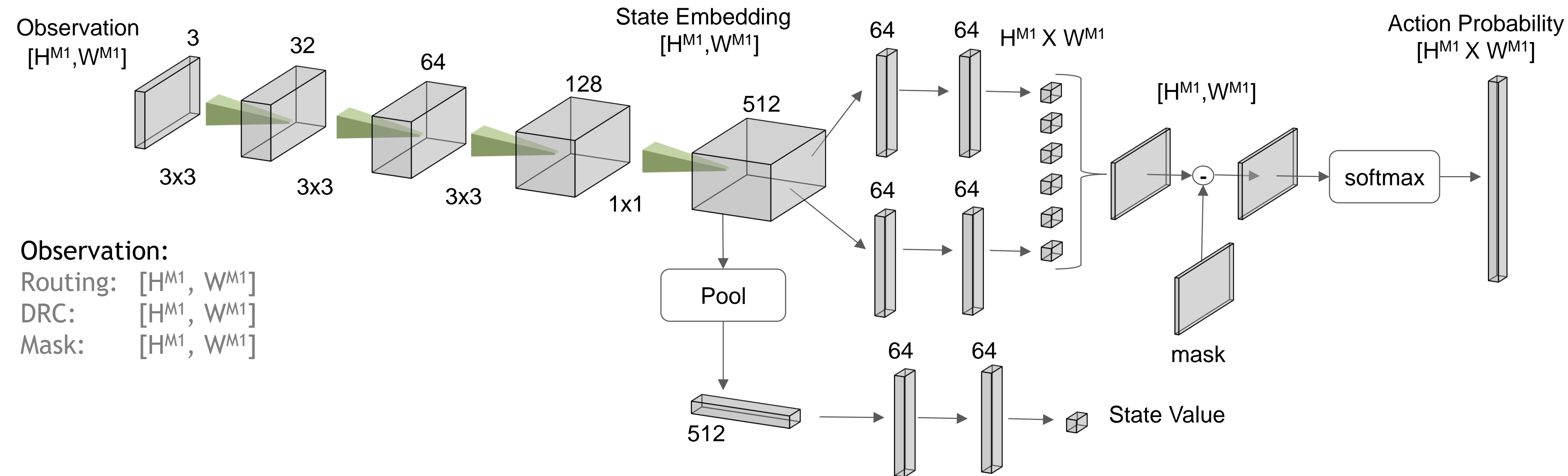


FIX DRC WITH RL



DRC RL MODEL

Agnostic to design size

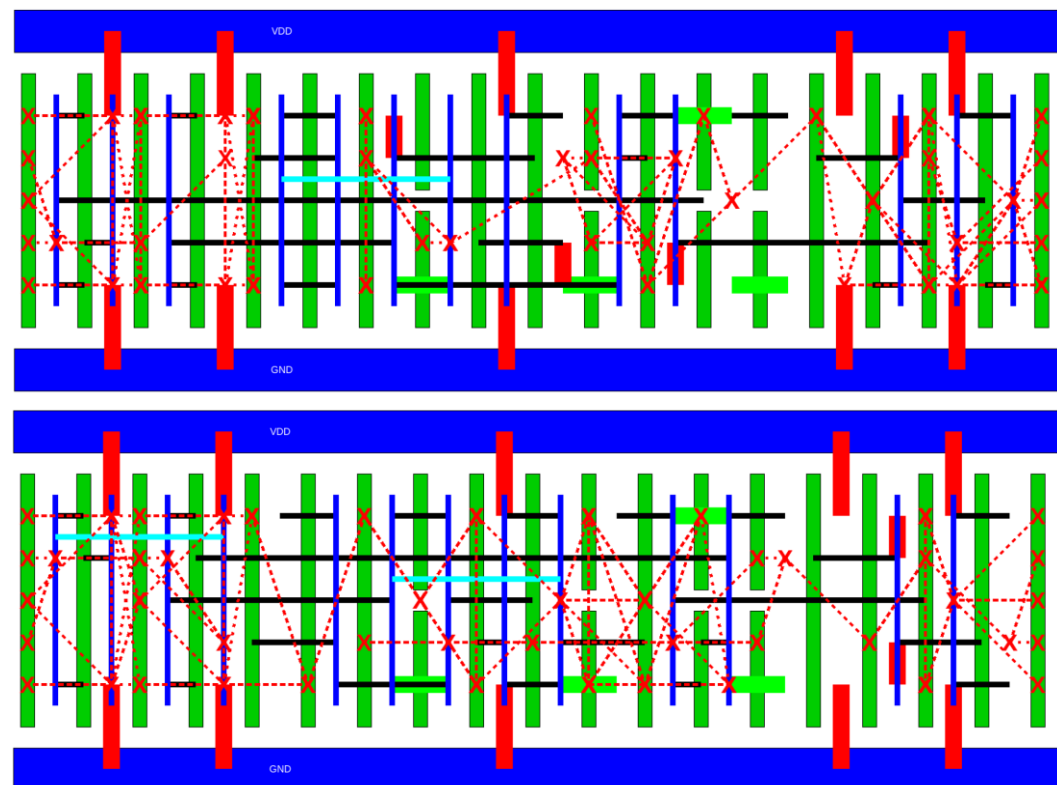


DRC AGENT TRAINING

RL algorithm: PPO2 in stable-baselines

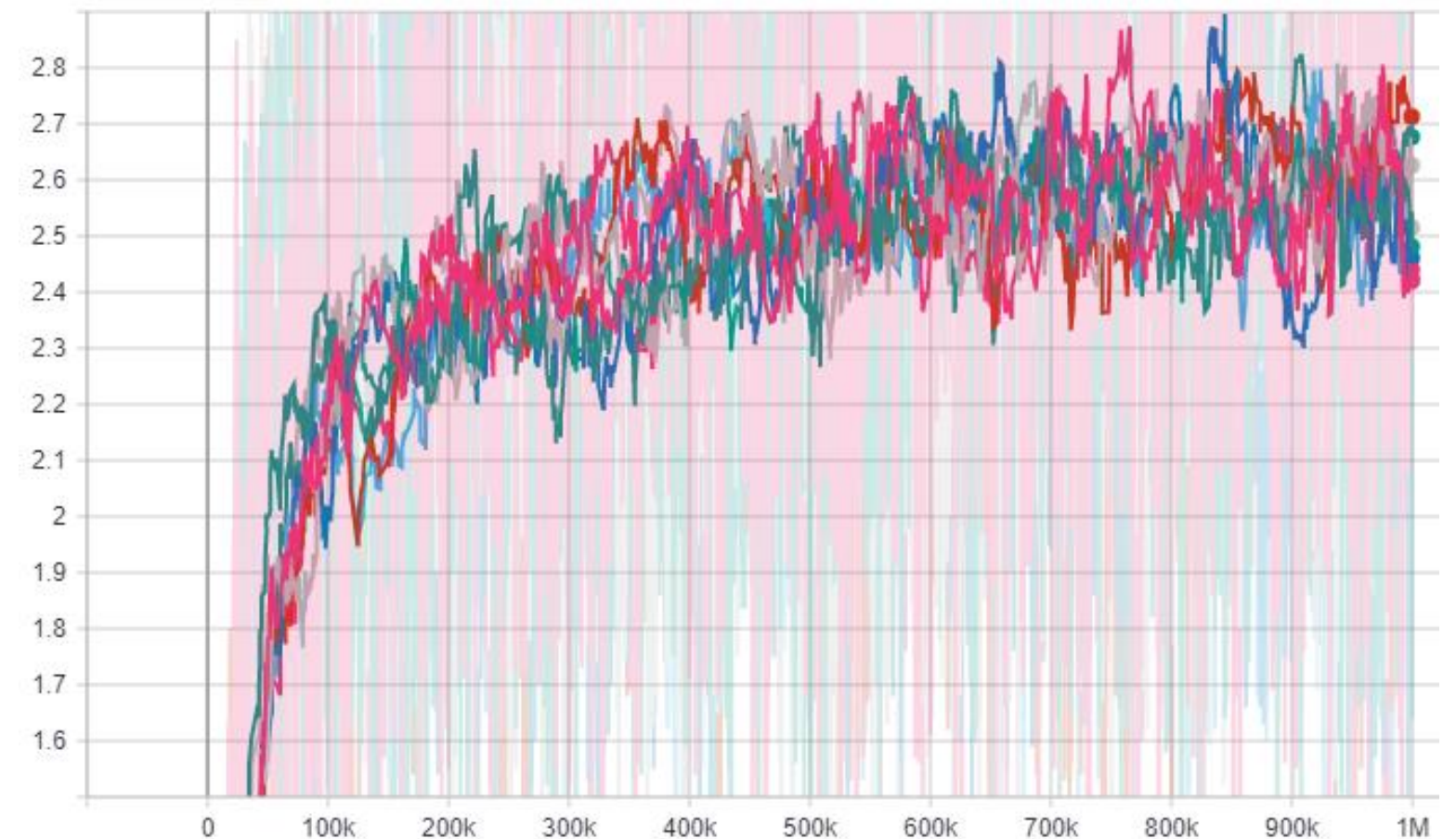
Training set: 10000 random maze routes for a flip-flop cell

Generalizes to all the cells



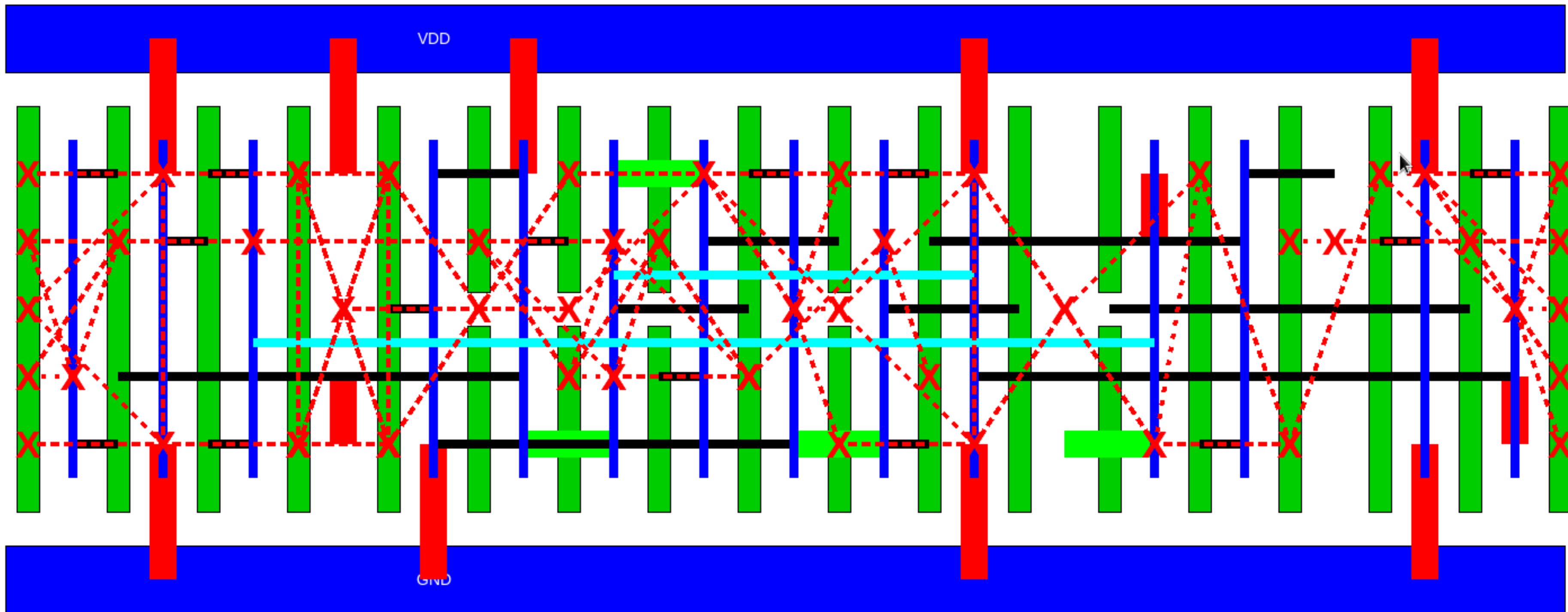
Random route 1

Random route 2



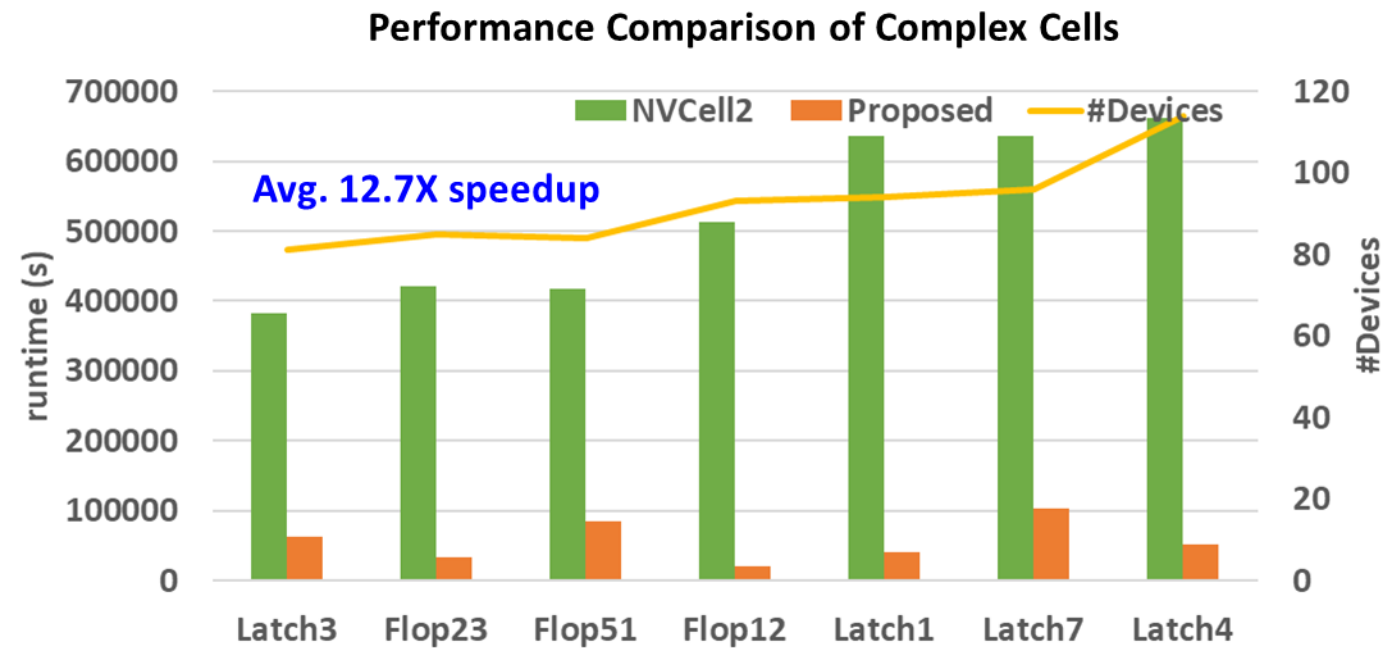
Reward history of 9 training runs

DRC FIXING EXAMPLE

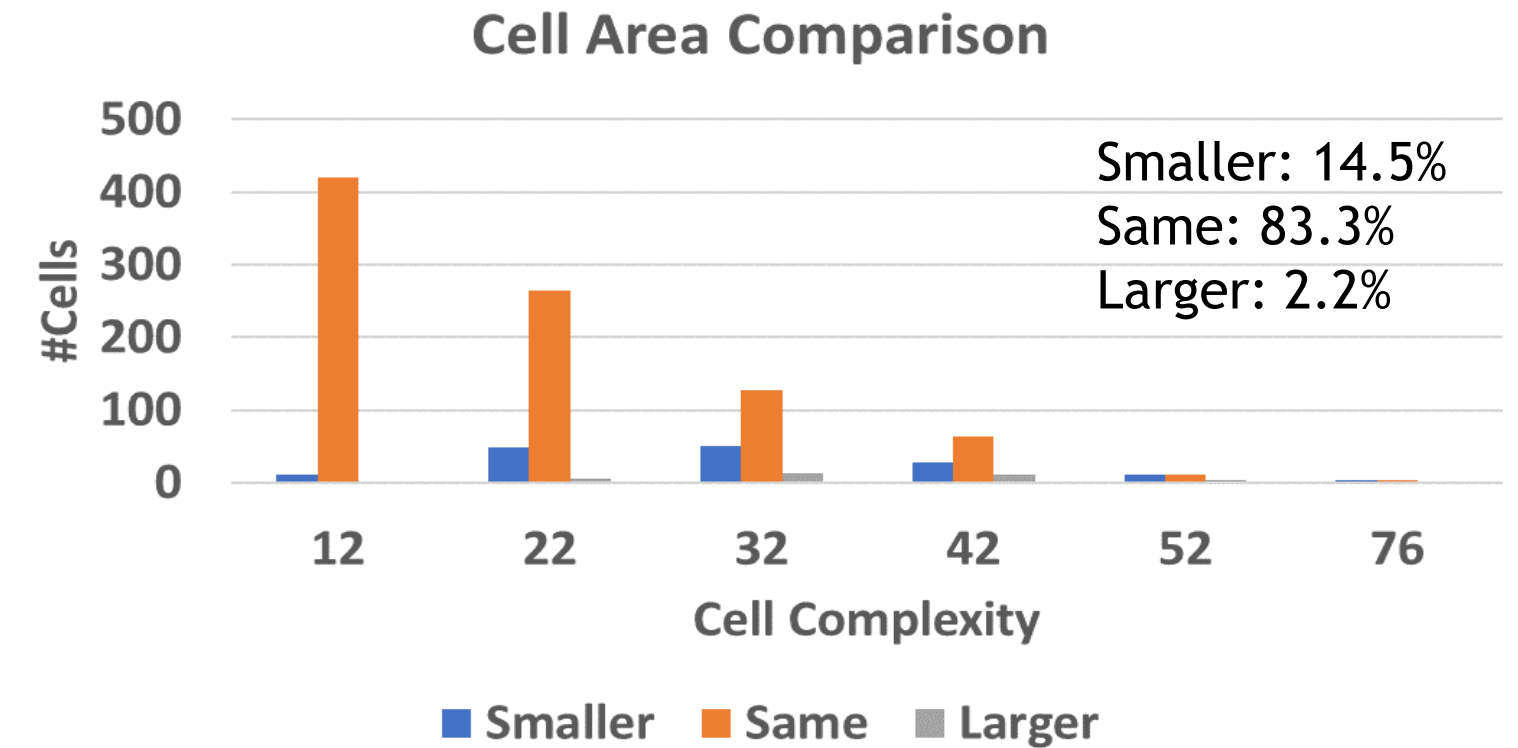


LAYOUT AND PERFORMANCE EVALUATION

Create 100% cells in industrial standard cell library



Achieved 12.7X speedup on average



	Success Rate (%)
NVCe1 (DAC 2021)	0.0%
NVCe2 (ISPD 2023)	87.2%
Layout-Aware Clustering	100%

On a difficult routing benchmark (94 cells)

	Success Rate (%)	Cell Width Comparison		
		Smaller	Same	Larger
NVCe1 (DAC 2021)	91.2%	11.8%	77.6%	1.8%
NVCe2 (ISPD 2023)	98.8%	13.7%	80.1%	4.3%
Layout-Aware Clustering	100%	14.5%	83.3%	2.2%

On entire cell library (over 1000 Cells)

Improved PPA metrics up to performance 7%, power 8%, and area 4%.

A network diagram consisting of numerous small circular nodes connected by thin, light-colored lines. The nodes are primarily white, with several highlighted in a bright green color. The connections form a complex, interconnected web that is denser in some areas and sparser in others. The background is a dark, solid color, making the white and green nodes stand out.

CONCLUSIONS

CONCLUSIONS

- We can leverage ML to improve chip design automation productivity and QoR.
- Transformer model based and generative model can be leveraged to improve the efficiency and solution quality for EDA optimizations.
- Algorithms + GPU acceleration + ML: A new EDA computing paradigm!

NVIDIA GH200

Grace Hopper Superchip

Processor for the Era of Accelerated Computing and Generative AI



72-Core Grace CPU
500 GB LPDDR5X
400 GB / sec
4 PFLOPS Hopper GPU
141 GB HBM3e
5 TB / sec

NVIDIA AI ENTERPRISE 4.0
ENTERPRISE-GRADE GENERATIVE AI PLATFORM

End-to-End, Fine-Tune to Deployment LLM Library

New Multi-GPU TensorRT LLM

4,500 Packages – 10,000 Dependencies

Multi Cloud, Datacenter, Workstations



servicenow snowflake
CLEAR ML DOMINO run:ai Weights & Biases
Dell Technologies Hewlett Packard Enterprise Canonical Ubuntu Microsoft aws Google Cloud
Lenovo SUPERMICR Z HP Red Hat vmware Microsoft Azure ORACLE CLOUD Infrastructure

Source: SIGGRAPH 2023

