香港中文大學
The Chinese University of Hong Kong

# Machine Learning in EDA: When and How

Bei Yu
Department of Computer Science & Engineering
Chinese University of Hong Kong

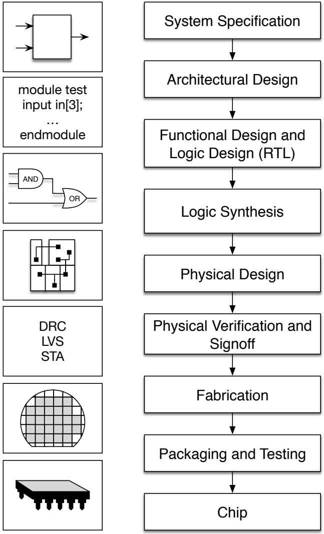
# Outline

1. When (machine learning integrated)
2. How (to solve unique challenges)
3. Future Direction

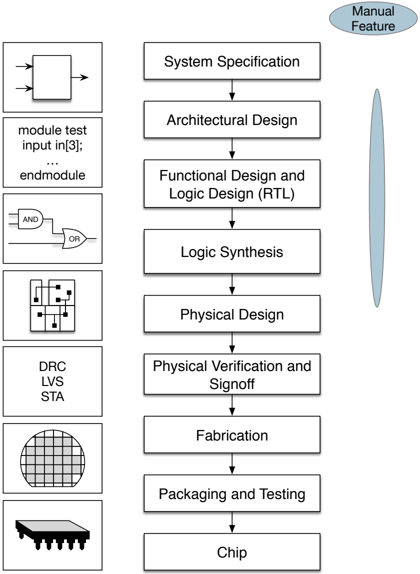
# Outline

1. When (machine learning integrated)
2. How (to solve unique challenges)
3. Future Direction

# When Machine Learning Integrated



# When Machine Learning Integrated



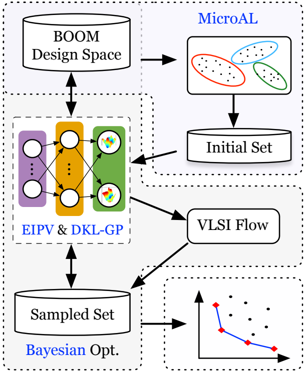
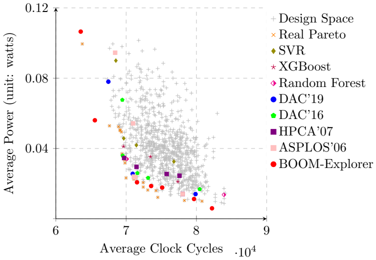


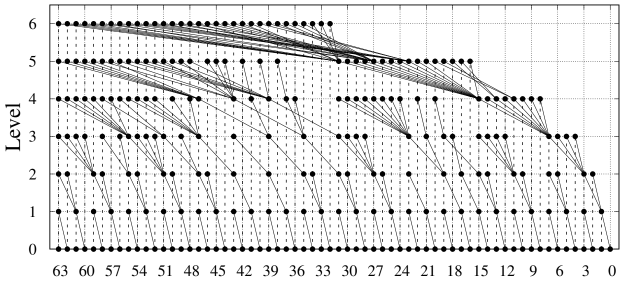

Table: Constraints of BOOM design specifications

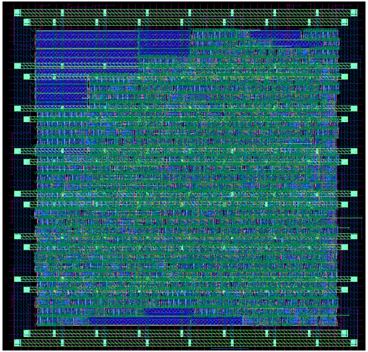
| Rule | Descriptions |
|---|---|
| 1 | FetchWdith $\geq$ DecodeWidth |
| 2 | RobEntry \| DecodeWidth [+] |
| 3 | FetchBufferEntry > FetchWidth |
| 4 | FetchBufferEntry \| DecodeWidth |
| 5 | fetchWidth = 2× ICacheFetchBytes |
| 6 | IntPhyRegister = FpPhyRegister |
| 7 | LDQEntry = STQEntry |
| 8 | MemIssueWidth = FpIssueWidth |

[+] The symbol "|" means RobEntry should be divisible by DecodeWidth




[1] Chen Bai et al. (2021). "BOOM-Explorer: RISC-V BOOM Microarchitecture Design Space Exploration Framework". In: *Proc. ICCAD*.



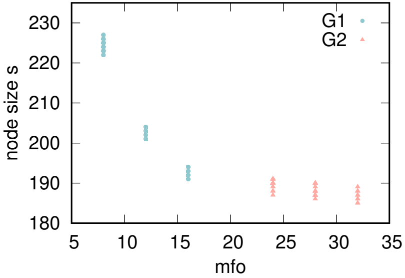



## Case Study: Adder Design

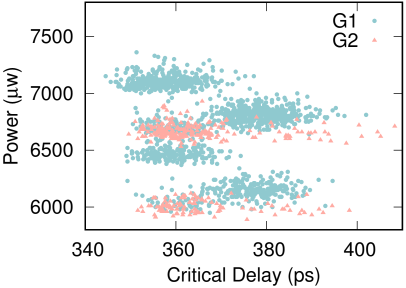
- Logic synthesis v.s. physical synthesis
- Constraints mapping between two synthesis stages is difficult.

# Active Learning in Logic Synthesis [TCAD'19][2]

**Front-End Team Perspective:**




**Back-End Team Perspective:**

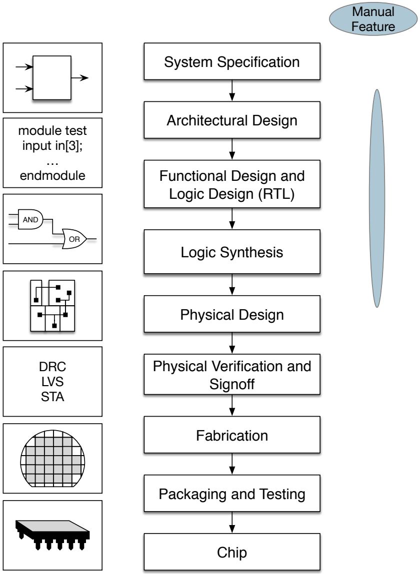



- Run design tools with all solutions is time-consuming.
- For 3K solutions, running time is $3000 \times 5 = 15$K mins.
- What we care: Pareto Frontier Curve

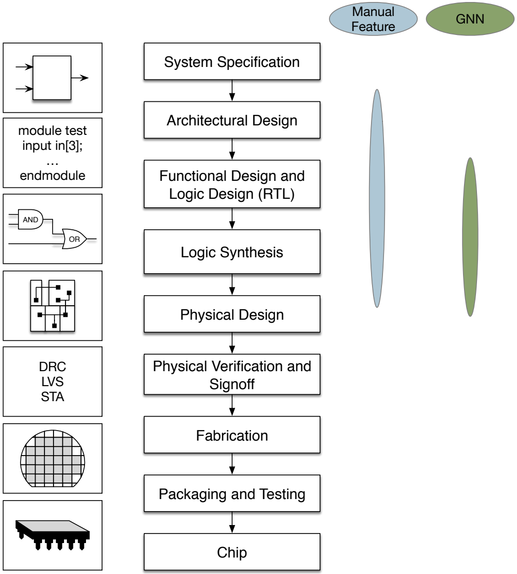
---

[2] Yuzhe Ma, Subhendu Roy, et al. (2019). "Cross-layer Optimization for High Speed Adders: A Pareto Driven Machine Learning Approach". In: *IEEE TCAD* 38.12, pp. 2298–2311.

# When Machine Learning Integrated

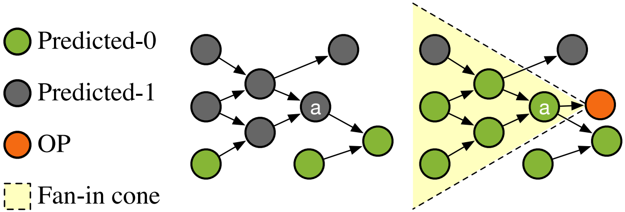


# When Machine Learning Integrated



# Example: Test Point Insertion [DAC'19][3]

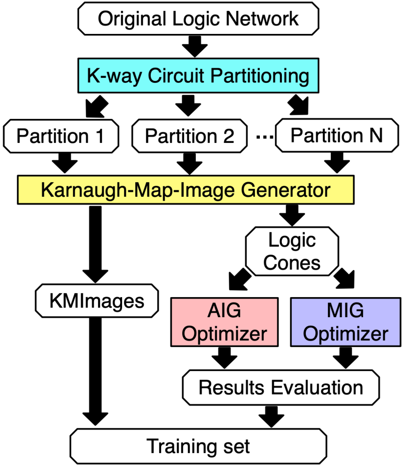
- Not every difficult-to-observe node has the same impact for improving the observability;
- Select the observation point locations with largest impact to minimize the total count.




(a) (b)

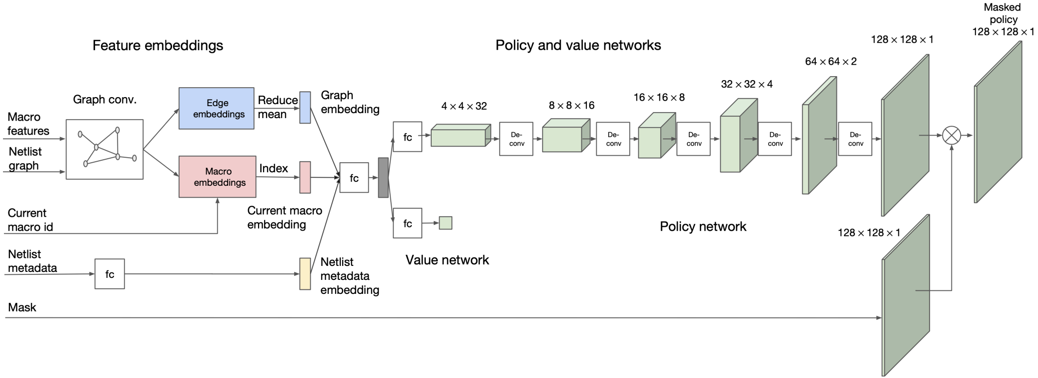
Yuzhe Ma, Haoxing Ren, et al. (2019). "High Performance Graph Convolutional Networks with Applications in Testability Analysis". In: *Proc. DAC*, pp. 1–6.

# Example: Logic Synthesis [ICCAD'19][4]



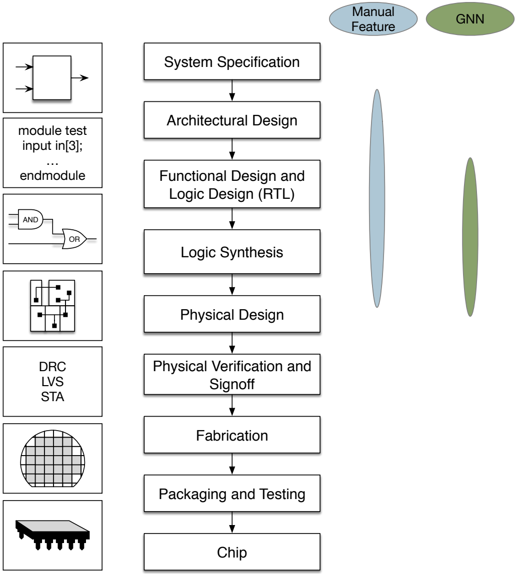

Walter Lau Neto et al. (2019). "LSOracle: A logic synthesis framework driven by artificial intelligence". In: *Proc. ICCAD*, pp. 1–6.

# Example: Macro Block Placement [Nature'21][5]

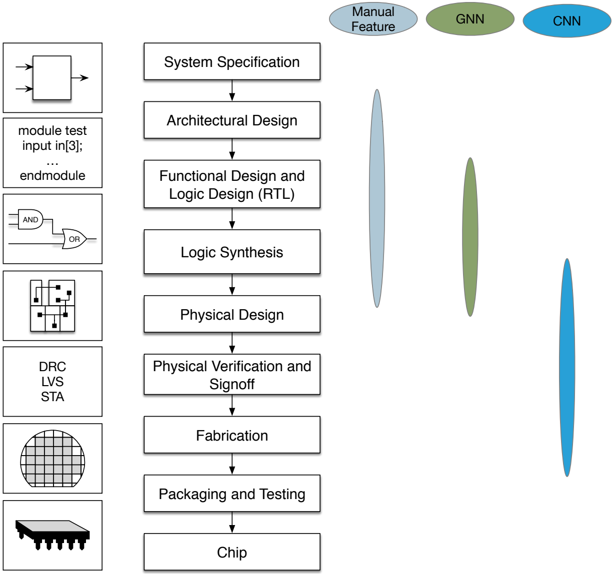


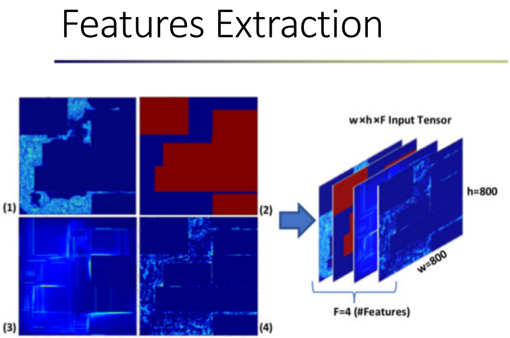

Azalia Mirhoseini et al. (2021). "A graph placement methodology for fast chip design". In: *Nature* 594.7862, pp. 207–212.

# When Machine Learning Integrated

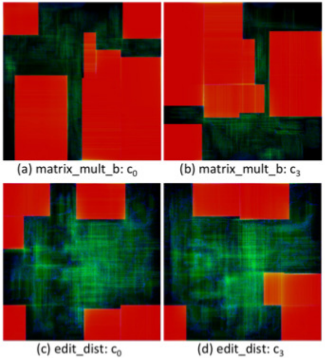


# When Machine Learning Integrated



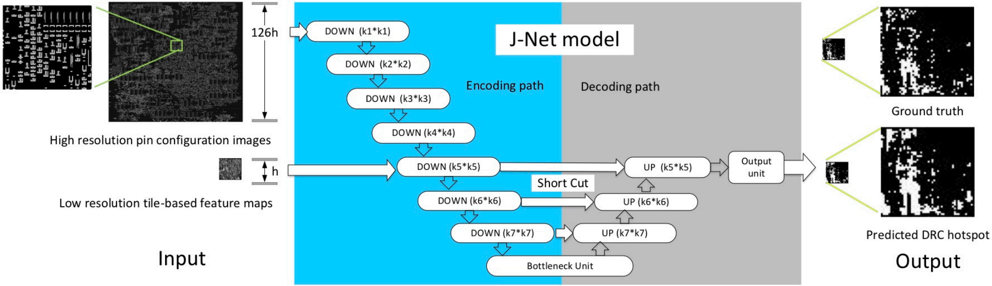



Input tensor constructed by stacking 2D features:
(1) Pin density, (2) macro (3) long-range RUDY, (4) RUDY pins



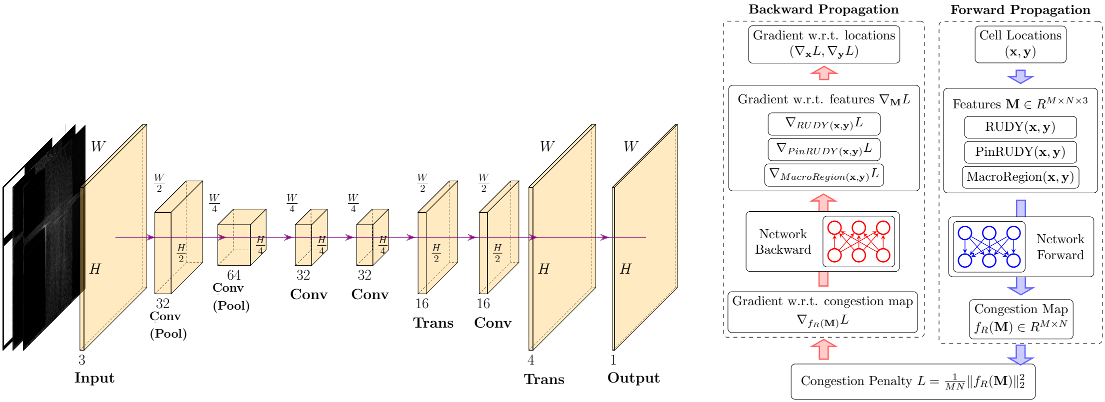

Input features for #DRV prediction.
Red: macro region
Green: global long-range RUDY
Blue: global RUDY pins

[6] Zhiyao Xie et al. (2018). "RouteNet: Routability Prediction for Mixed-Size Designs Using Convolutional Neural Network". In: *Proc. ICCAD*.




[7] Rongjian Liang et al. (2020). "DRC Hotspot Prediction at Sub-10nm Process Nodes Using Customized Convolutional Network". In: *Proc. ISPD*.

# CNN Driven Global Placement [DATE'21][8]



# Machine learning vs. traditional EDA methodologies

| | EDA Algorithms | ML |
|---|---|---|
| | Placement， route, synthesis, CTS, simulations, etc | Supervised， unsupervised, reinforcement learnings, etc |
| Pros | Known optimality, robust, less training data, good interpretability, Solve an abstract problem efficiently | GPU parallel computing, easy to design, end-to-end training on complex problems, Solve any problem by learning from its data |
| Cons | Over simplification of dynamic, complex problems | Rely too much on data, not leveraging the mechanics of the problem |

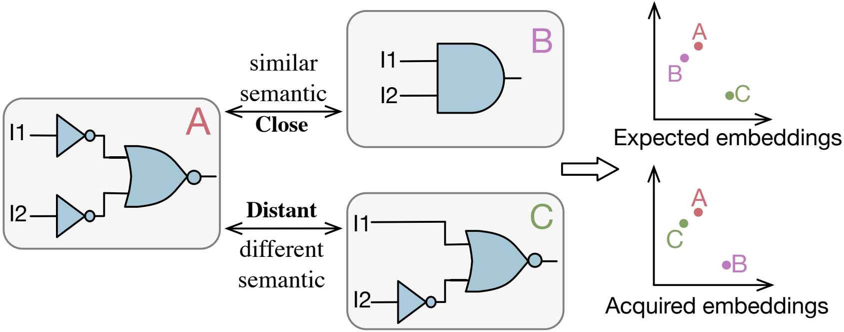
Yu Huang, “EDA Advanced Technologies”.

# Outline

# Challenge 1: Circuit Representation

# Defect of Previous GCN Works

- Previous works only focus on the graph structural information, which varies greatly across netlists.
- **We should extract general knowledge!**




Previous Structural GNN fail to capture the underlying semantic

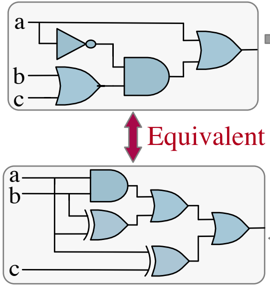
# Gate Functionality and Boolean Equivalence [DAC'22][9]

**Logic functionality**: keep the same across different designs.

- Generalized to unseen netlists, even with totally different topology!

**Can we extract this information?**

- **Yes!** –> **Key**: Boolean Equivalence




a + a'(b+c)

a(b+b'+c') + a'(b+c)

ab + (ab'+a'b) + (ac'+a'c)

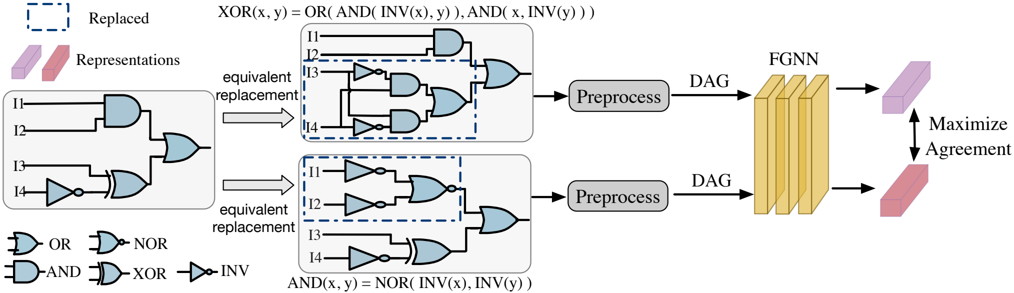
ab + xor(a,b) + xor(a,c)

---

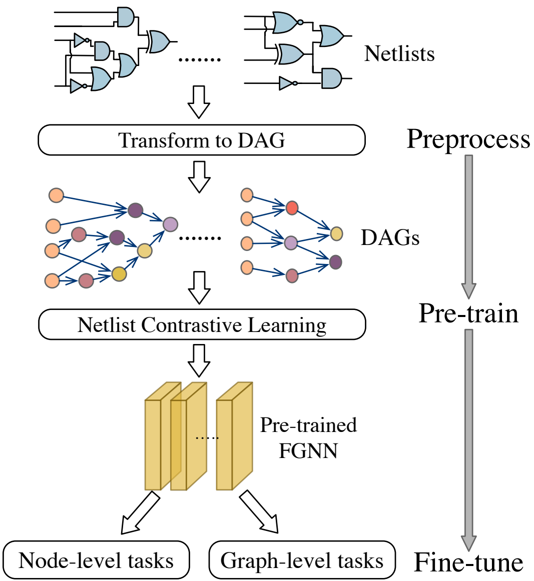
[9] Ziyi Wang, Chen Bai, et al. (2022). "Functionality Matters in Netlist Representation Learning". In: *Proc. DAC*.

# Netlist Contrastive Learning Scheme

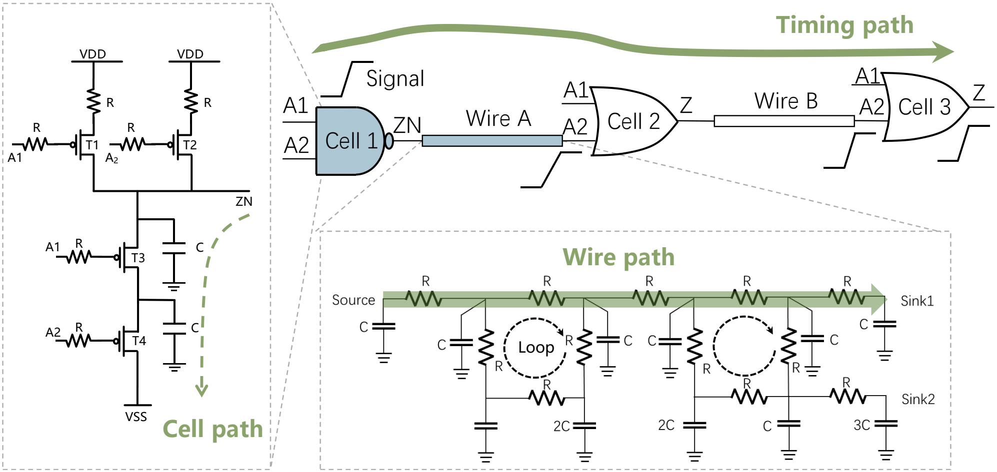
- Iterative random sub-netlist replacement.
- Positive sample pair share **same functionality**, while w. totally **different topology**.
- Maximizing agreement between positive samples: embedding of netlists with **similar semantic** (functionality) tend to be **close**
- Minimizing agreement between negative samples: distinguish from netlists with different semantic, even with similar topology.



# Pre-trained Circuit Learning Model



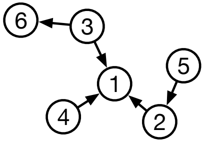
# Challenge 2: Timing Modeling

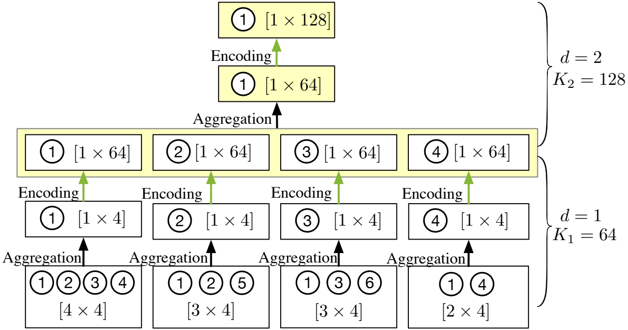



One example of the timing path in the netlist, cell (cell path) and wire (wire path).

# Why GCN cannot model timing well?

- Timing needs to capture path information
- GCN is only good at node embedding and graph embedding

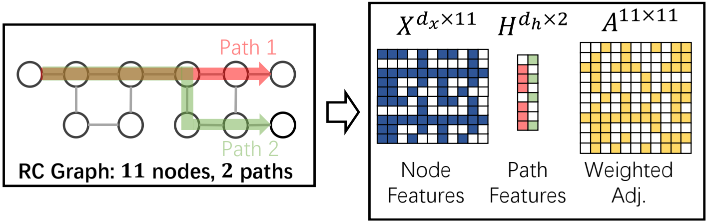


# Current Progress: Wire Timing [DATE'23][10]

The RC net graph $\mathcal{G}$ is represented with:

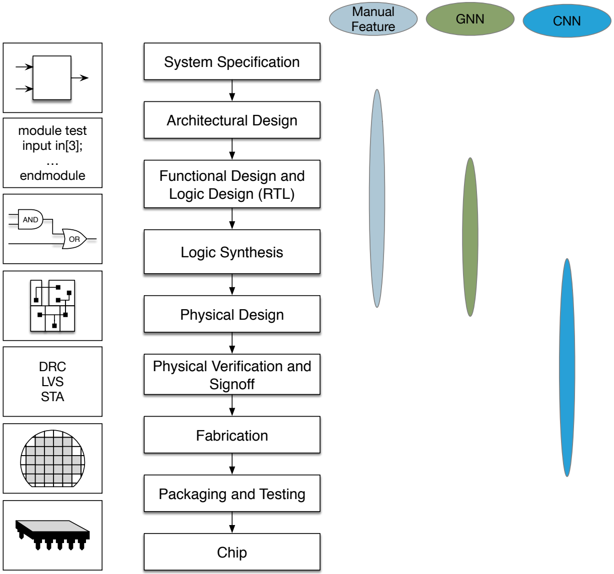
- Node feature matrix $\boldsymbol{X}$ for each capacitance
- Path feature matrix $\boldsymbol{H}$ for each wire path
- Weighted adj. matrix $\boldsymbol{A}$ for each resistance
- Label matrix for real wire slew and delay



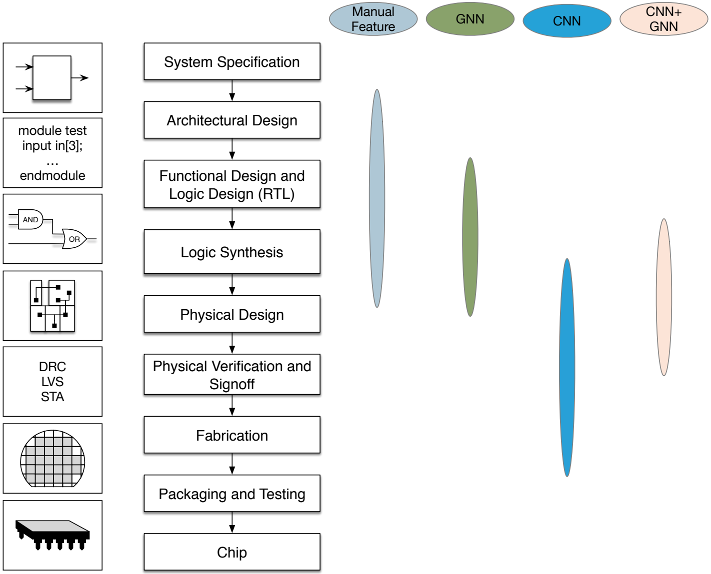

Yuyang Ye et al. (2023). "Fast and Accurate Wire Timing Estimation Based on Graph Learning". In: *Proc. DATE*.

# Challenge 3: Netlist+Layout: Multimodality

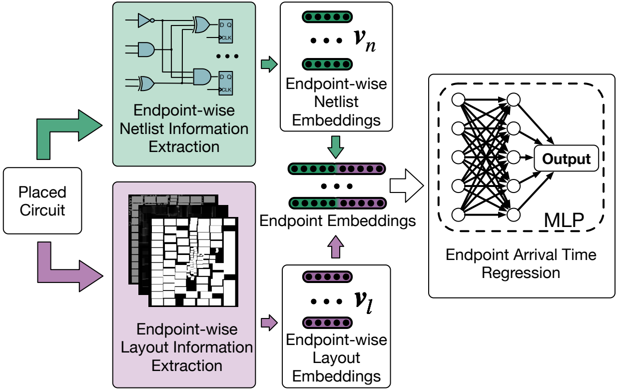
# Netlist+Layout: Multimodality



# Netlist+Layout: Multimodality



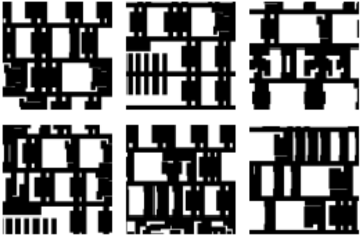
# GCN + CNN Flow [DAC'23][11]



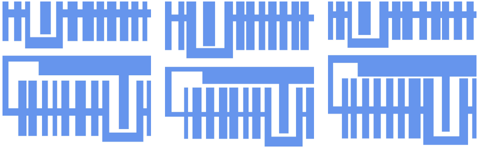

- Customized GNN: extract information from netlists
- CNN with novel masking technique: extract information from layouts.

[11] Ziyi Wang, Siting Liu, et al. (2023). "Realistic Sign-off Timing Prediction via Multimodal Fusion". In: *Proc. DAC*.

# Challenge 4: Constrained AIGC



Original Layout Patterns [ICCAD'20]



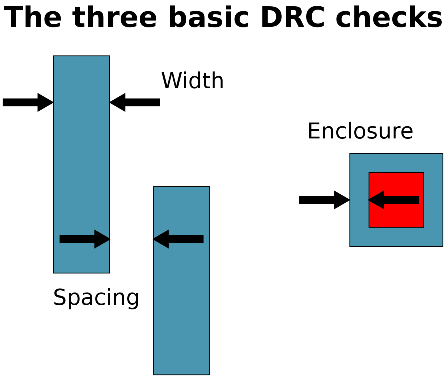
(a) (b) (c)

Generated Layout Patterns (Ours)

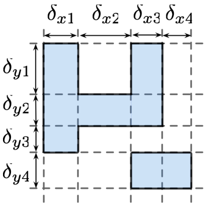
VLSI layout patterns provide critical resources in various designs for manufacturability research, from early technology node development to back-end design and sign-off flows[DAC'19][12].

[12]Haoyu Yang et al. (2019). "DeePattern: Layout pattern generation with transforming convolutional auto-encoder". In: *DAC*, pp. 1–6.

# An End-to-End Learning Solution?




- Maybe No
- Gap between Discrete Rules and Continuous DNN Model




Topology: $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$

Geometry: $\Delta_x = [\delta_{x1}, \delta_{x2}, \delta_{x3}, \delta_{x4}]$

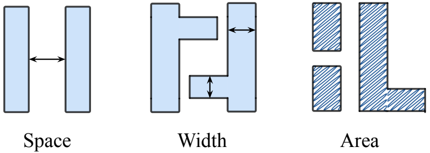
$\Delta_y = [\delta_{y1}, \delta_{y2}, \delta_{y3}, \delta_{y4}]$

## Squish Pattern [US Patent'14][13]

- Lossless and efficient representation method
- Encodes layout into pattern topology matrix and geometric information
- Problem #1: information density of each pixel is still not satisfactory

[13] Frank E Gennari and Ya-Chieh Lai (Sept. 2014). *Topology design using squish patterns*. US Patent 8,832,621.

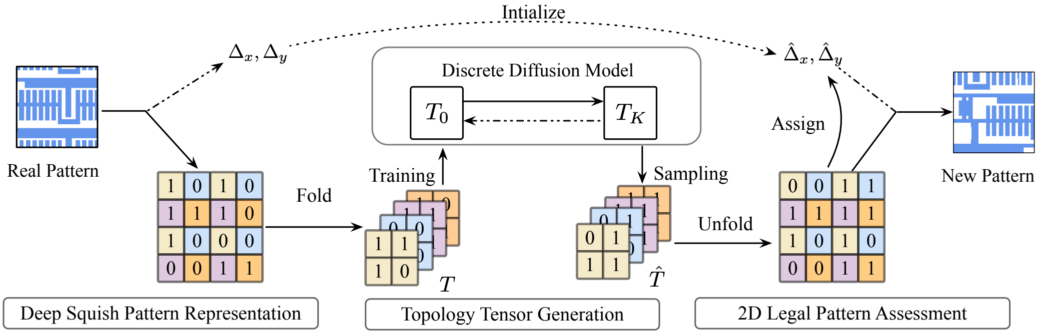
# Pattern Legalization




Examples of DRC Rule

## Finding legal distance vector for each topology

- Solving a Linear System (1D pattern) [DAC'19].
- Using Exist Distance Vector (2D pattern) [ICCAD'20]
- Problem #3: 2D pattern introduces non-linear constraint, hard to solve!




[14] Zixiao Wang et al. (2023). "DiffPattern: Layout Pattern Generation via Discrete Diffusion". In: *Proc. DAC*.

# Outline

1. When (machine learning integrated)
2. How (to solve unique challenges)
3. Future Direction

R-CNN: Region-Based CNN

Input image

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slides from Justin Johnson

Machine Learning is to Fit A Function $f(x)$

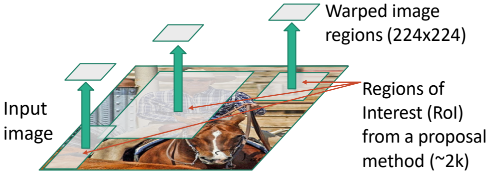
R-CNN: Region-Based CNN


Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.


Slides from Justin Johnson

Machine Learning is to Fit A Function $f(x)$

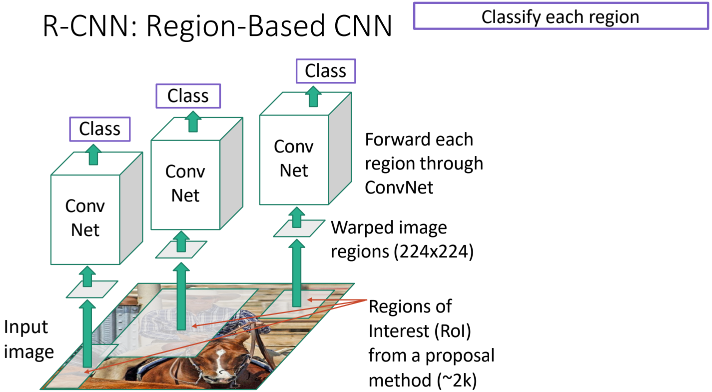
R-CNN: Region-Based CNN




Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slides from Justin Johnson

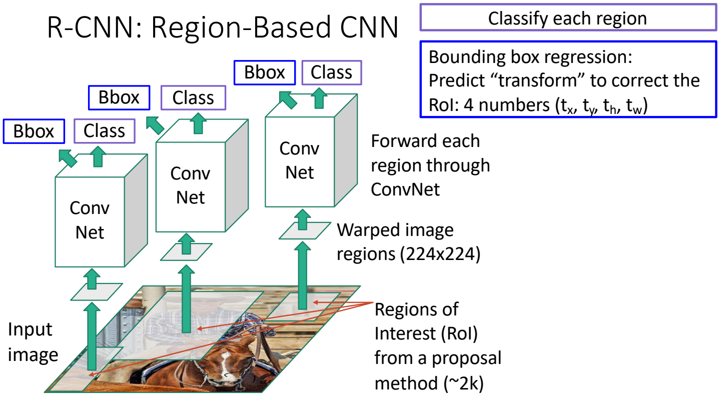
Machine Learning is to Fit A Function $f(x)$




Slides from Justin Johnson

**Machine Learning is to Fit A Function $f(x)$**

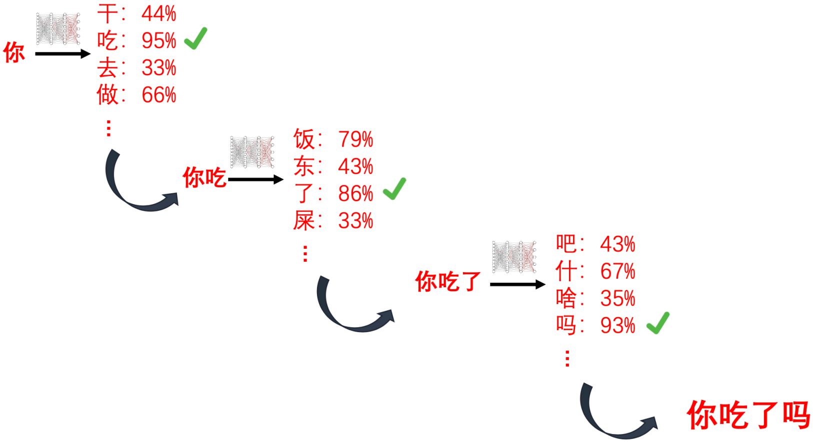
# How Machine Can Understand Picture?




Slides from Justin Johnson

Machine Learning is to Fit A Function $f(x)$

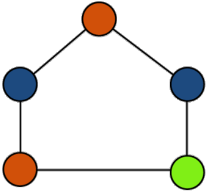
# How Machine Can Understand Text?



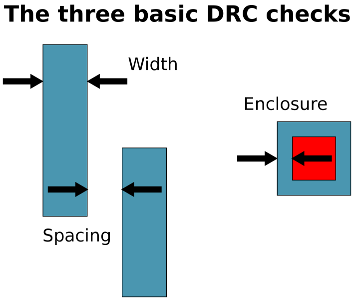

Machine Learning is to Fit A Function $f(x)$

# Challenges to AI

- Combinatorial Problem (e.g. Coloring)
- Handling Complicated Rules

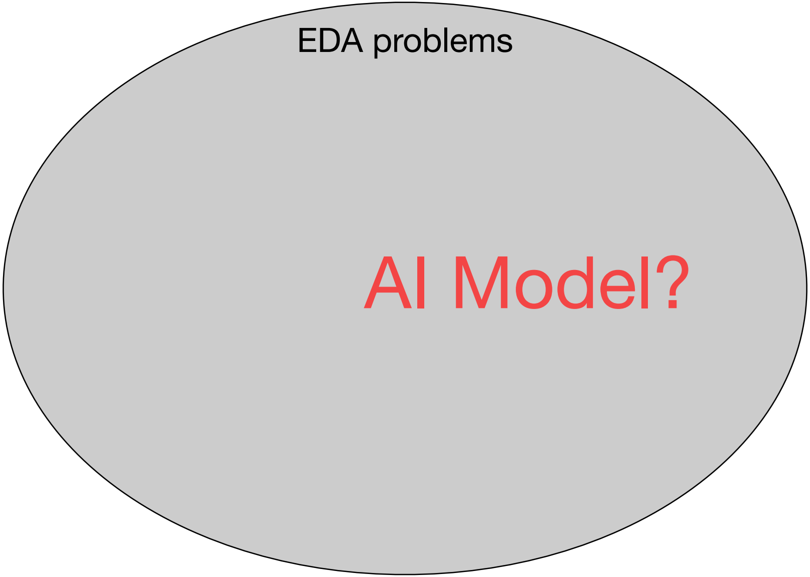


(a) Graph Coloring

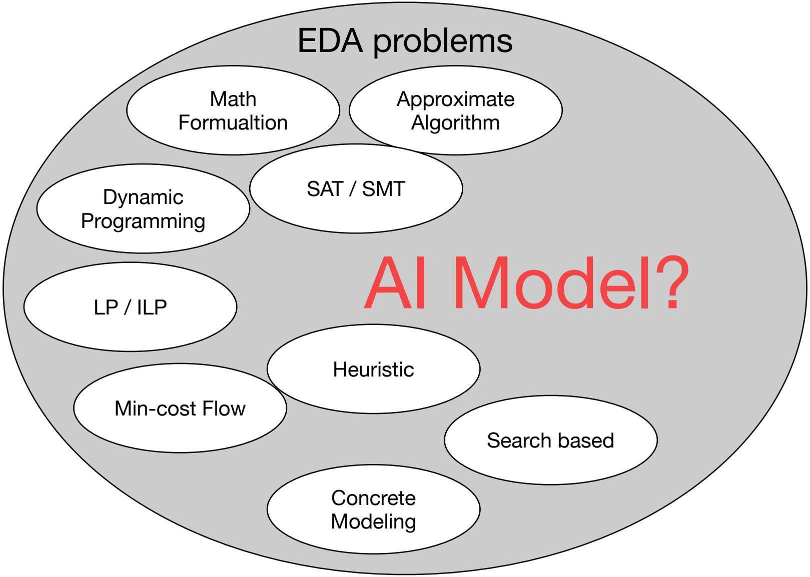



(b) Design Rule Checking

# Know the Boundary!



# Know the Boundary!



# THANK YOU!