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
byu@cse.cuhk.edu.hk

July 30, 2023

ME

我明天要给电路设计的学者介绍机器学习在电路设计自动化中的应用，我需要介绍哪几点呢？

当你准备在电路设计自动化中介绍机器学习的应用时，以下是一些你可能需要包括的主题：

1. **机器学习的基本原理**：首先，你可能需要解释什么是机器学习以及其基本原理，包括监督学习、无监督学习、强化学习等。如果听众对这些概念不够熟悉的话，这会是一个很好的入门。
2. **机器学习在电路设计自动化中的作用**：接下来，你可以介绍一下机器学习如何应用于电路设计自动化，包括用于优化设计流程、预测电路性能、提高设计效率等方面。例如，深度学习可以用于模拟和预测电路性能，遗传算法可以用于优化电路布局等。
3. **实际应用案例**：你可以分享一些实际的应用案例，说明机器学习如何在电路设计自动化中发挥作用。这些案例可以包括使用机器学习进行故障检测、预测电路性能、优化电路布局等。
4. **当前的挑战和未来的机会**：在介绍完上述内容后，你可能还想讨论一下当前在这一领域中面临的挑战，比如数据的获取和处理，模型的训练和验证，以及应用的广度和深度等。同时，也可以探讨一下未来的发展趋势和可能的机会，比如利用最新的机器学习技术（如强化学习）进行更高级的电路设计自动化。
5. **工具和资源**：最后，你还可以提供一些相关的工具和资源，以供听众进一步了解和探索这一主题。这些工具和资源可以包括机器学习框架（如TensorFlow，PyTorch），电路设计软件，以及相关的学习和研究资料。

ME

太长了，请帮忙缩短到三点

当然，我可以帮你将这个话题简化为以下三个主要点：

1. **机器学习在电路设计自动化中的应用**：例如，使用深度学习模拟和预测电路性能，使用遗传算法优化电路布局等。
2. **实际案例与挑战**：分享实际应用案例，展示机器学习如何改变电路设计流程，并讨论在实施中遇到的挑战，例如数据获取和处理，模型训练和验证等。
3. **未来趋势和机会**：探讨利用最新的机器学习技术（如强化学习）在电路设计自动化的潜力和可能的发展方向。

我希望这样的概述对你的演讲有所帮助。祝你好运！


ME
太长了，请帮忙缩短到三点
当然，我可以帮你将这个话题简化为以下三个主要点：
1. 机器学习在电路设计自动化中的应用
ML Applications
路性能，使用遗传
算法优化电路布局等。
2. 实际案例与挑战：
Real Cases
机器学习如何改变电路设计流程，并讨论在实施
中遇到的挑战，例
训练和验证等。
3. 未来趋势和机会：
Future Directions
化学习）在电路设计自动化的潜力和
可能的发展方向。
我希望这样的概述对你的演讲有所帮助。祝你好运！

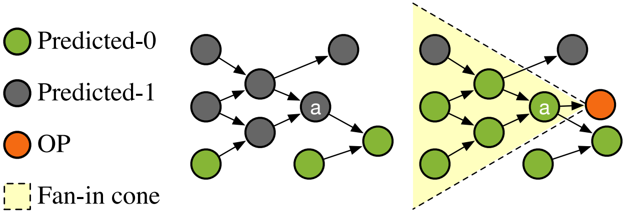
Outline

Outline

Circuit Representation

Example: Test Point Insertion[1]

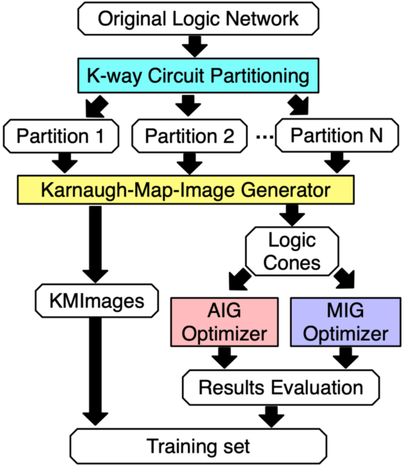
- Not every difficult-to-observe node has the same impact for improving the observability;
- Select the observation point locations with largest impact to minimize the total count.



(a) (b)

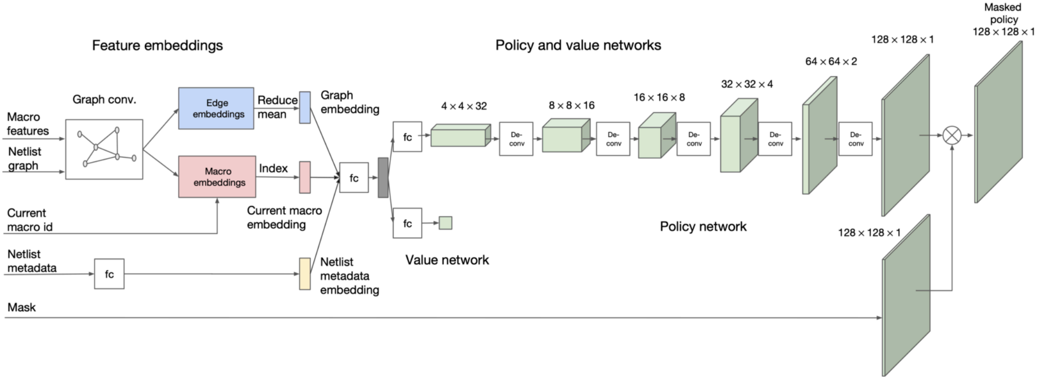
[1] Yuzhe Ma et al. (2019). "High Performance Graph Convolutional Networks with Applications in Testability Analysis". In: *Proc. DAC*, pp. 1–6.

Example: Logic Synthesis[2]



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

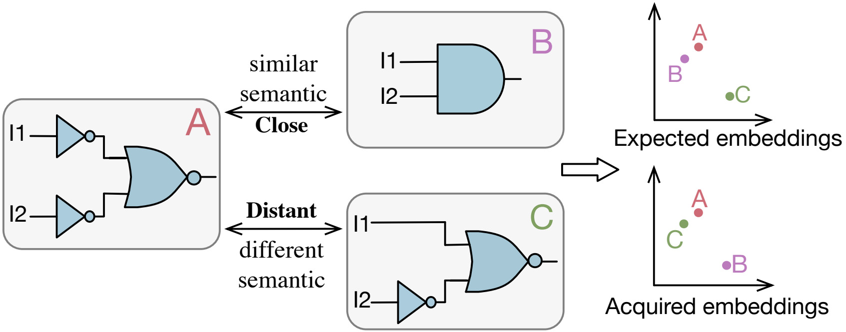
Example: Macro Block Placement[4]



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

Defect of previous 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

Question:

- What is the **universal** and **transferable** knowledge that is shared across different netlists?
- Can we **extract** the shared prior knowledge to enhance the ability of graph learning models?

Functionality Matters in Netlist Representation Learning

Ziyi Wang, CUHK
Chen Bai, CUHK
Zhuolun He, CUHK
Guangliang Zhang, HiSilicon
Qiang Xu, CUHK
Tsung-Yi Ho, CUHK
Bei Yu, CUHK
Yu Huang, HiSilicon

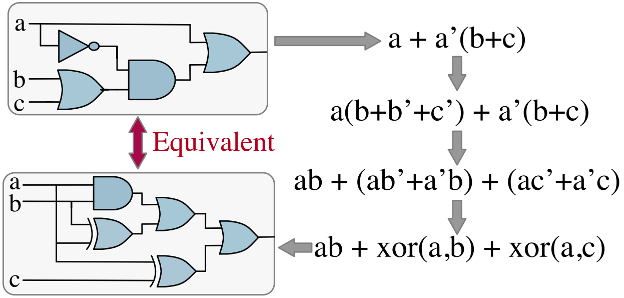
DAC 2022

Logic functionality: keep the same for a specific gate type across different designs.

- Can be transfered and generalized to unseen netlists, even with totally different topology!

Can we extract this information?

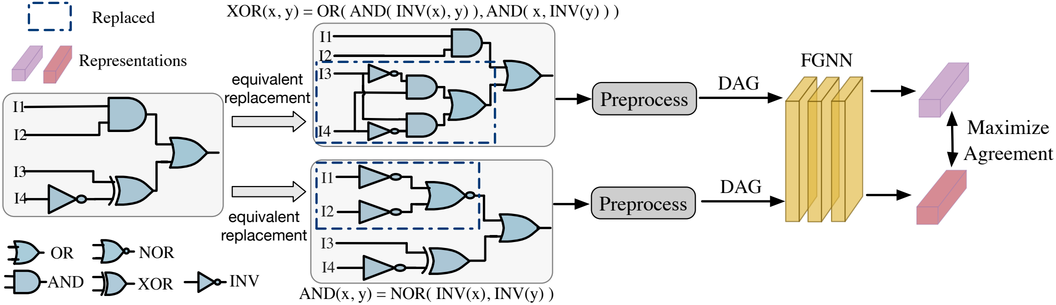
- **Yes**! –> **Key**: Boolean Equivalence



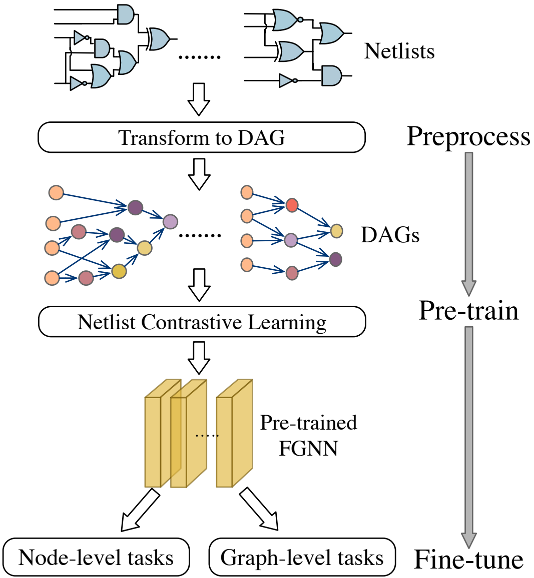
example of Boolean equivalence

Netlist Contrastive Learning Scheme

- Iterative random sub-netlist replacement.
- Positive sample pair share the **same functionality**, while having 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.

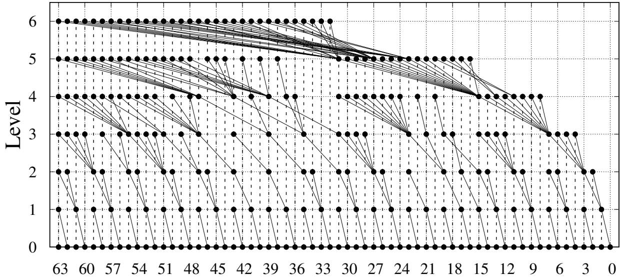


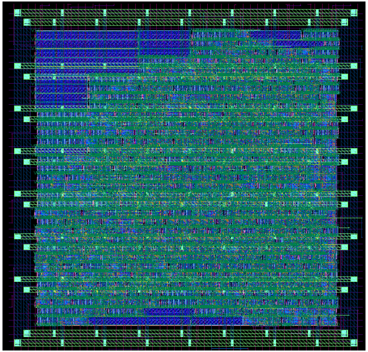
Overall Flow



Tool Parameter Exploration

Gaps Between Design Stages



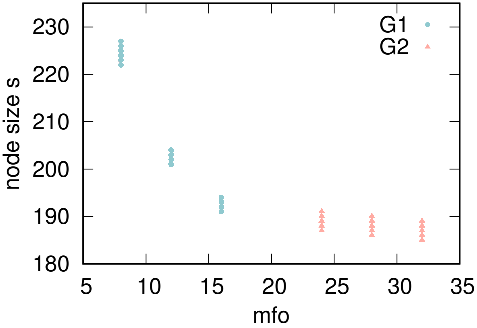


Case Study: Adder Design

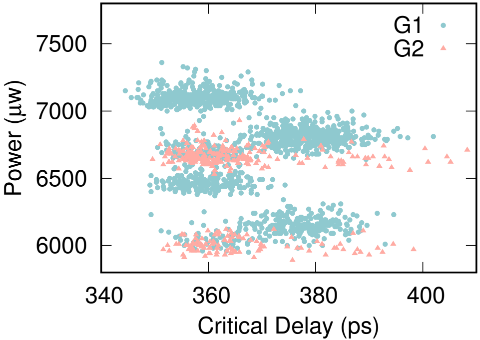
- Logic synthesis v.s. physical synthesis
- Constraints mapping between two synthesis stages is difficult.

Why Smart Sampling ?

Front-End Team Perspective:



Back-End Team Perspective:

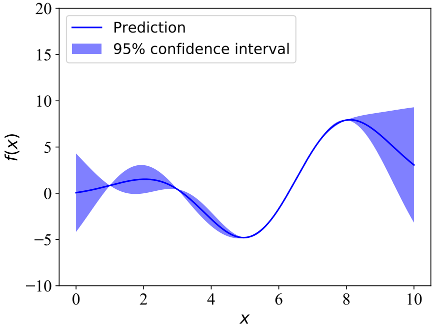


- Run design tools with all solutions is time-consuming.
- For 3K solutions, running time is $3000 \times 5 = 15$K mins.
- What we care: Pareto Frontier Curve

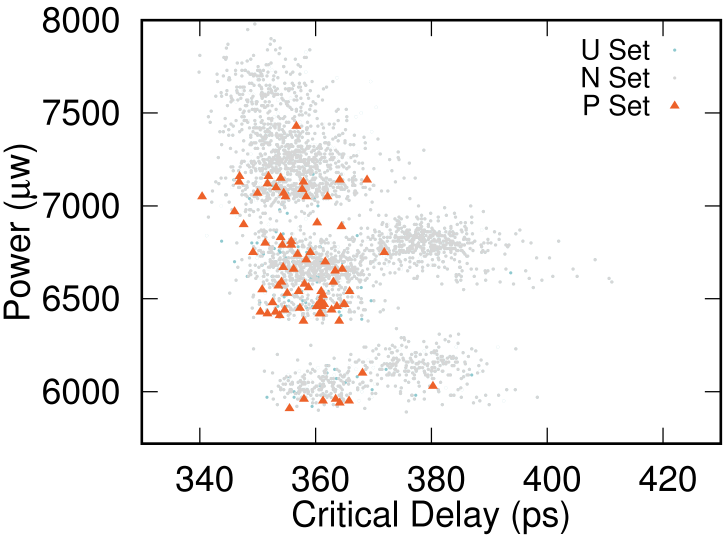
Active Learning Flow [TCAD'19][1]

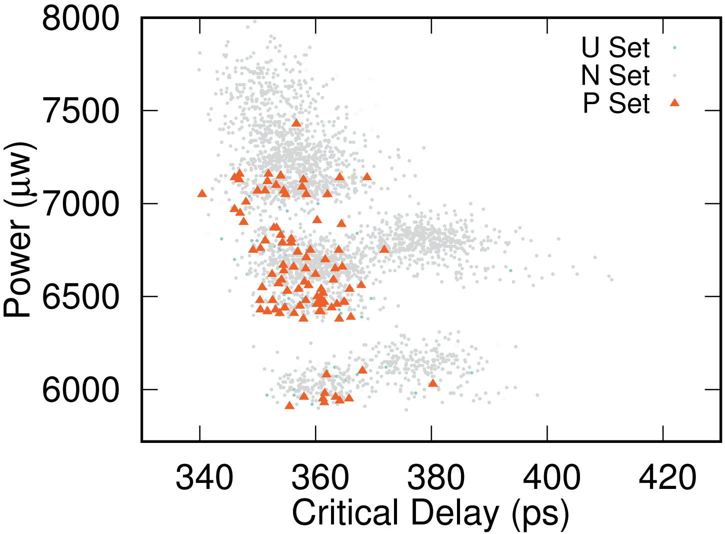
Regression

- Gaussian process model;
- A prediction consists of a mean and a variance;
- Off-the-shelf library for implementation.

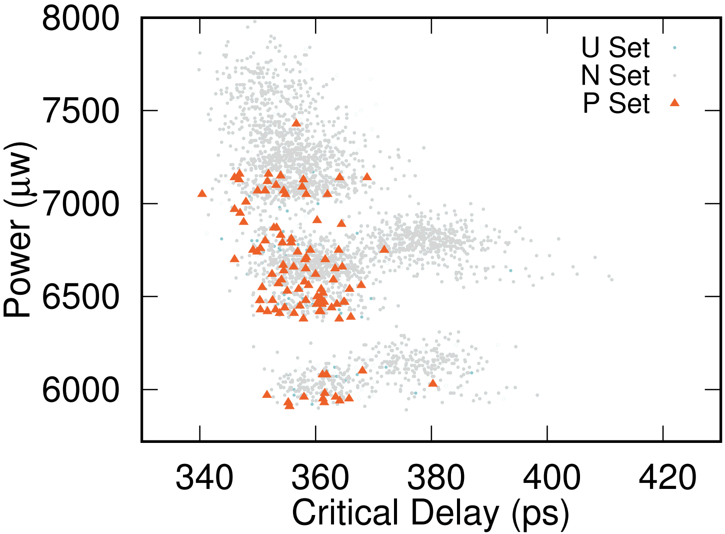


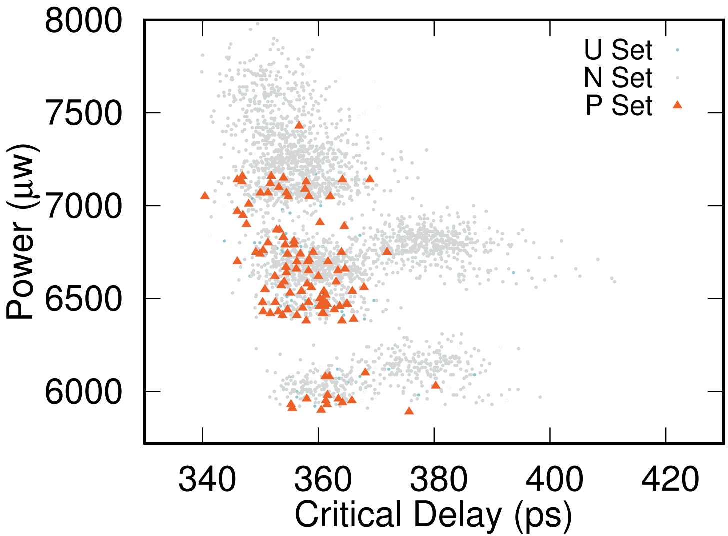
[1] Y. Ma, S. Roy, J. Miao, J. Chen and B. Yu, "Cross-Layer Optimization for High Speed Adders: A Pareto Driven Machine Learning Approach", TCAD'19.


8000
7500
7000
6500
6000
Power (µw)
340
360
380
400
420
Critical Delay (ps)
U Set
N Set
P Set

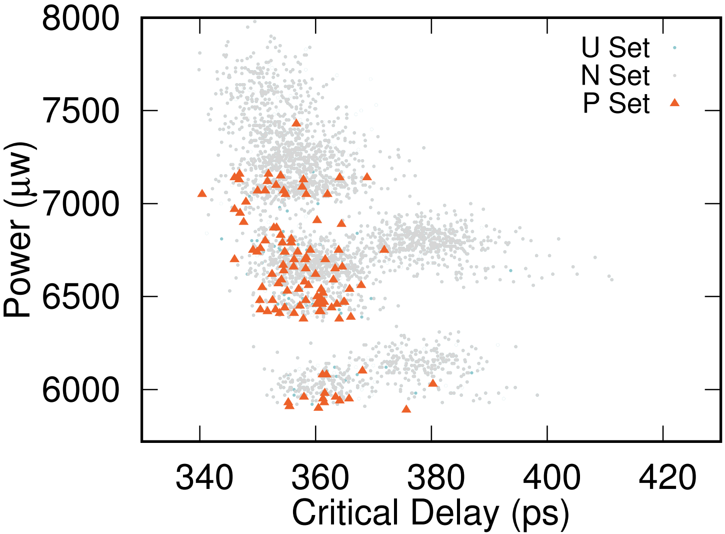


8000
7500
7000
6500
6000
Power (μw)
340
360
380
400
420
Critical Delay (ps)
U Set
N Set
P Set

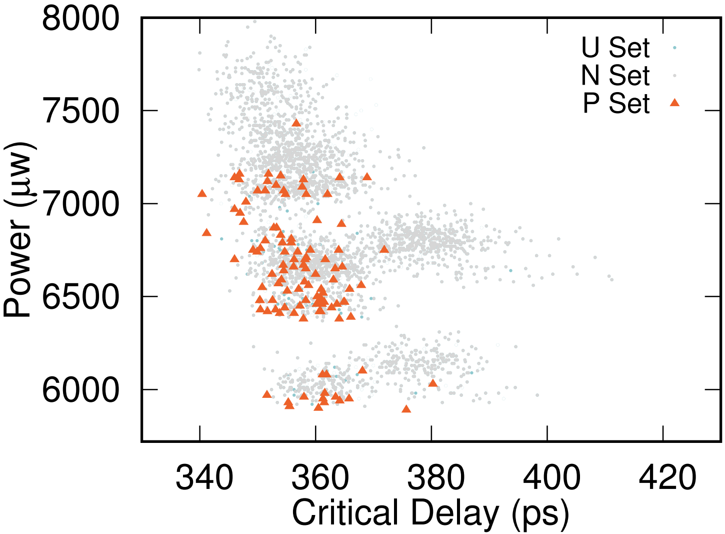

8000
7500
7000
6500
6000
Power (μw)
340
360
380
400
420
Critical Delay (ps)
U Set
N Set
P Set



U Set
N Set
P Set
8000
7500
7000
6500
6000
340
360
380
400
420
Critical Delay (ps)
Power (µw)

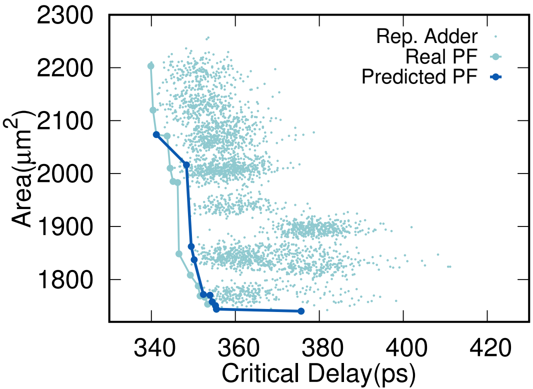


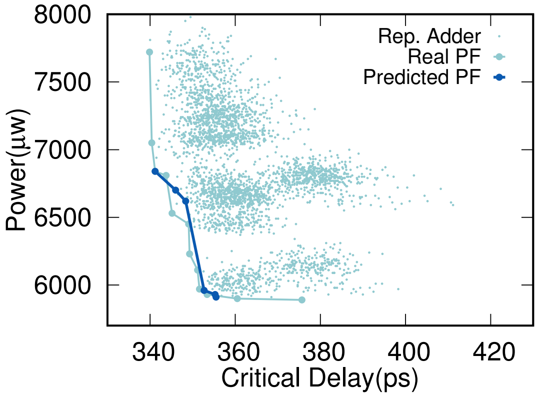
U Set
N Set
P Set
Power (μw)
8000
7500
7000
6500
6000
340
360
380
400
420
Critical Delay (ps)



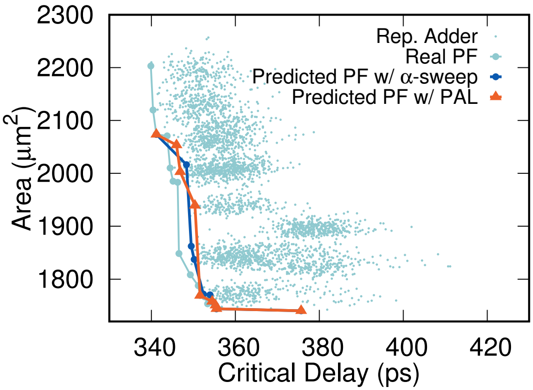
8000
7500
7000
6500
6000
Power (μw)
340
360
380
400
420
Critical Delay (ps)
U Set
N Set
P Set

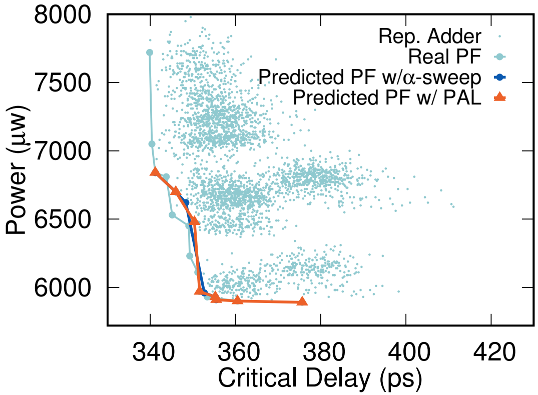
Pareto-Frontier Results





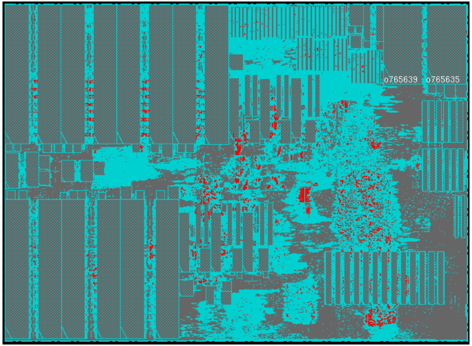
Pareto-Frontier Results





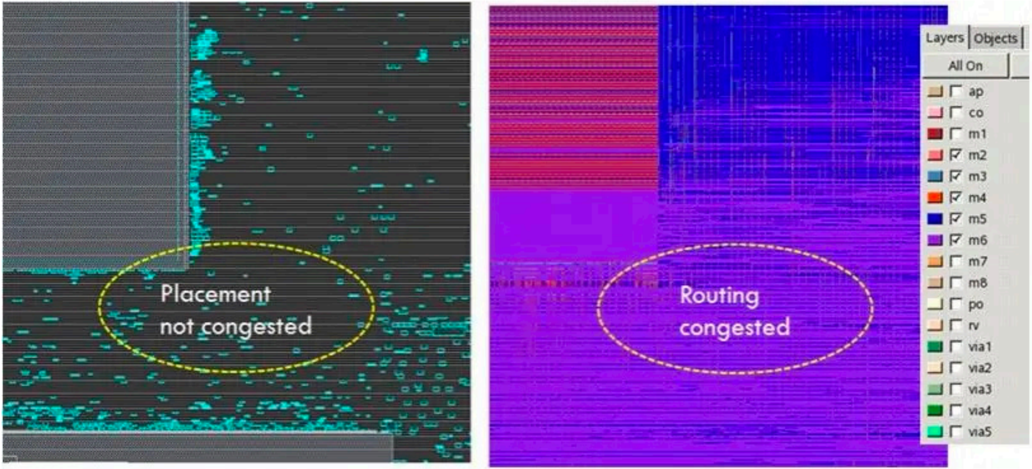
Placement & Routing

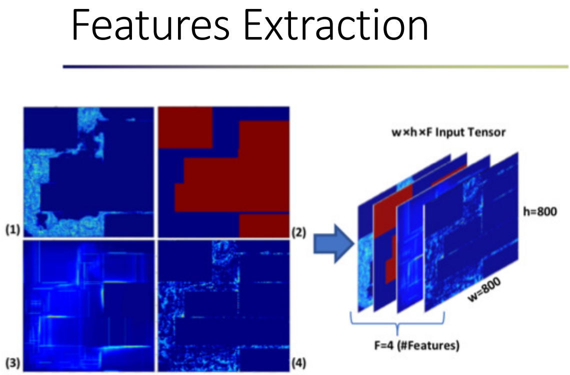
Challenge: Larger and Larger Scale



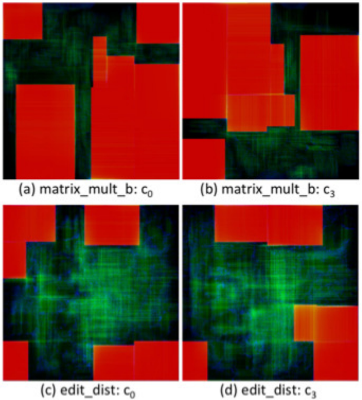
- Back-end is time consuming
- Accurate connectivity should be predictable
- Better estimation means efficient design-to-market budget

Complicated Relationship in Placement & Route





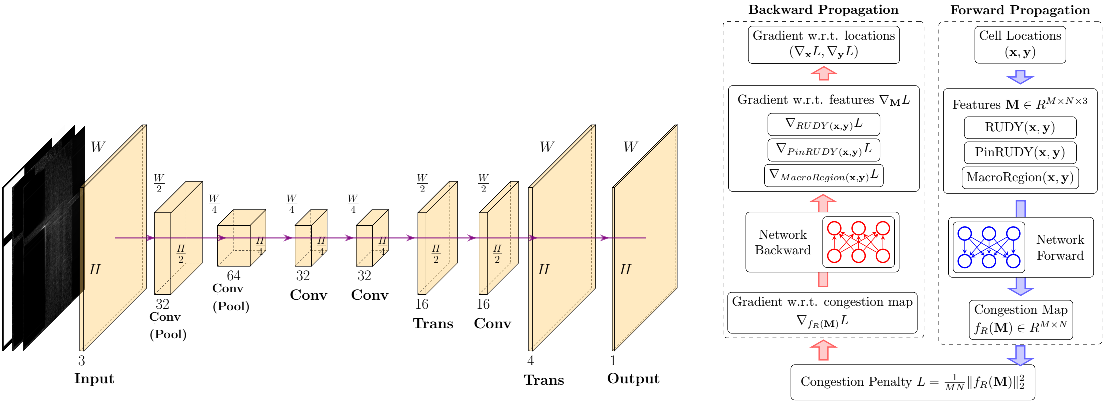
Input tensor constructed by stacking 2D features:
(1) Pin density, (2) macro (3) long-range RUDY, (4) RUDY pins



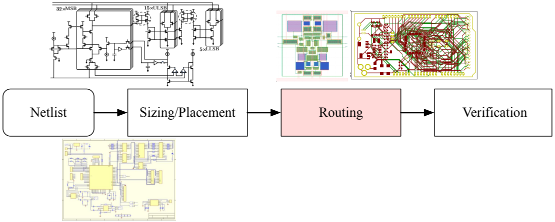
Input features for #DRV prediction.
Red: macro region
Green: global long-range RUDY
Blue: global RUDY pins

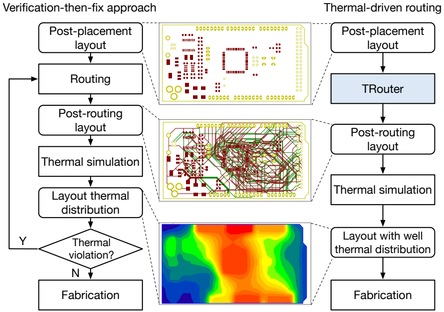
[3] Xie+, "RouteNet: Routability prediction for Mixed-Size Designs Using Convolutional Neural Network", ICCAD'18.

CNN Driven Global Placement [DATE'21][1]



[1] Siting Liu et al. (2021). "Global Placement with Deep Learning-Enabled Explicit Routability Optimization". In: *Proc. DATE*.

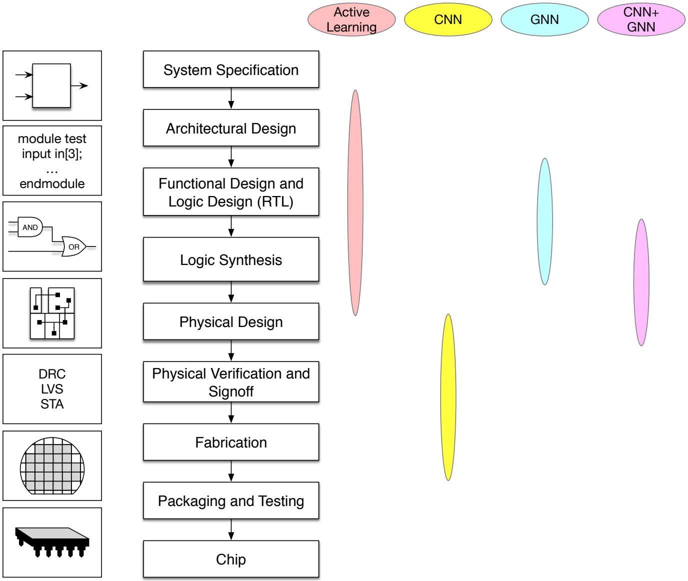




Challenge and Solution

- ☹ Challenge: It is difficult to reduce thermal distributions on the board;
- ☺ Solution: Integrated reliability verification into routing stage.

Conclusion[2]



[2] Some important ML techniques are not covered here: e.g. reinforcement learning; metric learning; unsupervised learning; casual inference; federate learning.

Outline

Cadence Revolutionizes Verification

To dramatically improve verification productivity, it's time for a new generation of verification tools that optimizes across multiple runs of multiple engines and leverages big data and AI. Watch to learn more about Verisium AI-Driven Verification.

Tackling PCB Layout Automation with AI-Based System Design Technology

Corporate VP Michael Jackson discusses how Allegro X AI combines traditional physical design algorithms with AI techniques to produce compelling PCB layout.

DSO.ai: Achieve PPA Targets Faster with the World's First AI Application for Chip Design

Synopsys DSO.ai™ searches for optimization targets in very large solution space of chip design by using AI engines to enhance power, performance, and area. DSO.ai facilitates massive scaling in the exploration of choices by building on training data and applying learnings across projects to accelerate tape-out and achieve PPA targets.

Explore DSO.ai

VSO.ai

Synopsys VSO.ai™ is the industry's first AI-driven verification solution aimed at achieving higher coverage quality faster so that bugs don't go undetected. The technology identifies and eliminates redundancies in regressions and automates root cause analysis. VSO.ai examines the RTL and stimulus to infer coverage and highlight coverage gaps, saving significant time and ensuring a high ROI on tests.

Explore VSO.ai

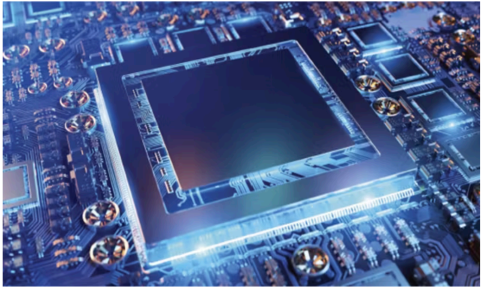
TSO.ai

Synopsys TSO.ai™ is the first autonomous AI application for semiconductor test. It optimizes test program generation in complex designs to achieve maximum defect coverage with fewer test patterns. The result is faster automatic test pattern generation runtime and lower silicon test costs.

Explore TSO.ai

WHITE PAPER

Using machine learning methods in production-ready engineering solutions for IC verification



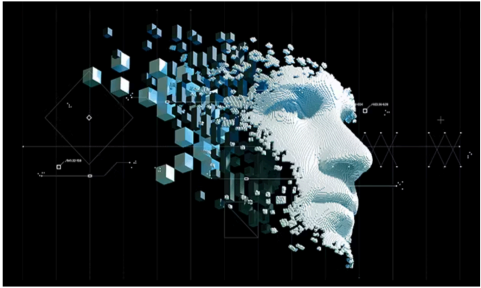
Utilizing ML to produce consistent, verifiable, and correct answers for SPICE-level IC verification

Solido High-Sigma Verifier, part of Solido Variation Designer, is an ML-enabled solution that provides full brute-force accurate verification results at 4, 5, and 6+ sigma, with orders-of-magnitude less simulation runtime. With Solido High-Sigma Verifier, design and verification teams can increase verification accuracy and coverage, while significantly reducing design schedule times. Solido High-Sigma Verifier is an example of a level 3 ML algorithmic design, which is quickly approaching level 4 through large-scale production use and iterative

WHITE PAPER

Reducing electronic systems design complexity with AI

Reading time: 7 minutes



This paper explores crucial electronic systems design areas where artificial intelligence (AI) technologies — including machine learning and deep learning — can be applied to minimize or eliminate the complexity of electronic system design work.

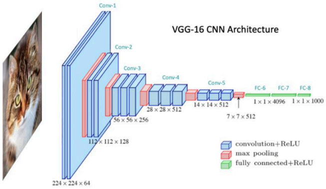
Siemens' goal is to deliver AI-enhanced tools that help electrical engineers and designers:

- Make informed decisions, increasing efficiency

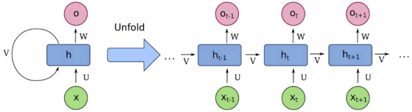
Outline

Future Direction

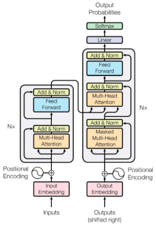
过去十年科研界大部分工作都在尝试各种网络架构（就是函数）



CNN



RNN



Transformer

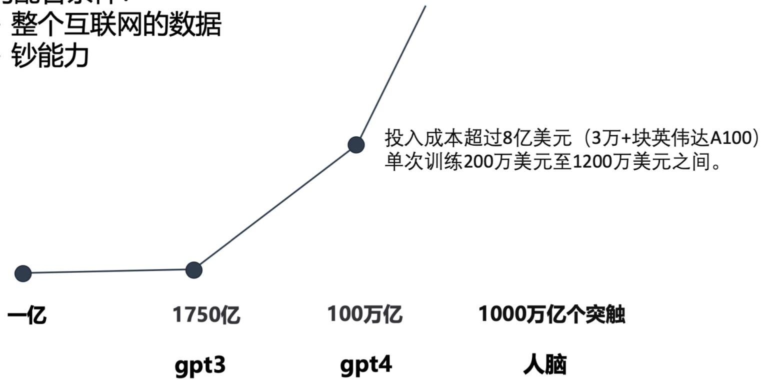
不同任务
不同场景
不同数据
针对性的进行网络设计

NN：Neural Network

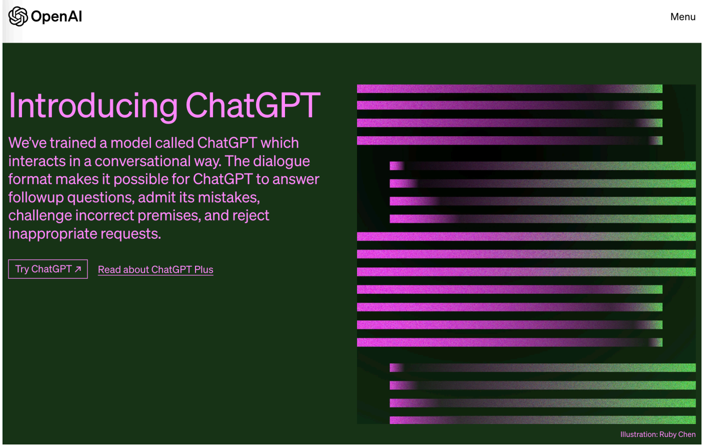
人工智障?

大模型需要的配合条件:

- 大数据 -> 整个互联网的数据
- 大算力 -> 钞能力



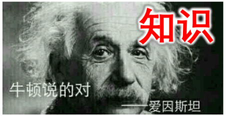
ChatGPT



- in dialog form
- can answer consecutive questions
- can admit its mistakes
- challenge people to say wrong conclusions
- reject inappropriate requests

https://openai.com/blog/chatgpt

知识的本质是信息的压缩



苹果熟了会往下落
椰子熟了也会往下掉
梅老板比我重
打篮球投篮是抛物线
月球围着地球转，地球绕着太阳转
开车太快容易撞
在月球上的重力比地球轻
地球离太阳越近，公转速度越快
黑洞能吸收万物
⋮

$$F = ma$$

$$F = G\frac{m_1 \times m_2}{r^2}$$

两个铁球同时落地
预测某个时刻的月球轨迹
计算煤老板在月球上的重力有多少
计算要多少个恒星发动机才能流浪地球
⋮

矩阵 a= tensor([[0.3348, -0.1575, -0.0552, 0.4606, -0.4005],
[1.2785, -0.0594, -2.4832, -0.1052, -1.5892],
[-0.3722, -0.1244, -0.6635, 0.6163, -0.8171],
[-1.4294, -0.6973, -0.0049, -0.8030, -1.2963]]) 矩阵 tensor([[-1.6391, -1.8575],
[1.4069, 0.8627],
[0.4147, -0.0830],
[0.7041, 0.1294],
[-1.6289, 0.2686]])请计算a和b的矩阵乘法

矩阵 A 是一个 4x5 矩阵，矩阵 B 是一个 5x2 矩阵。由于矩阵 A 的列数（5）与矩阵 B 的行数（5）相等，所以这两个矩阵可以相乘。乘积矩阵的大小为 4x2。

矩阵相乘的方法是将第一个矩阵的每一行与第二个矩阵的每一列相乘，然后求和。以下是矩阵相乘的结果：

css Copy code

```css
[ [-0.1878, -0.6610],
  [ 1.5415, -3.5917],
  [ 0.3561,  0.1316],
  [-0.9432,  2.0052]
]
```

Correct Answer:

$$\begin{bmatrix} 1.2766, & 0.0329 \\ 6.1799, & -2.6547 \\ -0.8886, & 0.1450 \\ 0.8300, & -0.1394 \end{bmatrix}$$

Challenges to AI

S

矩阵 a= tensor([[0.3348, -0.1575, -0.0552, 0.4606, -0.4005],
[1.2785, -0.0594, -2.4832, -0.1052, -1.5892],
[-0.3722, -0.1244, -0.6635, 0.6163, -0.8171],
[-1.4294, -0.6973, -0.0049, -0.8030, -1.2963]]) 矩阵 tensor([[-1.6391, -1.8575],
[1.4069, 0.8627],
[0.4147, -0.0830],
[0.7041, 0.1294],
[-1.6289, 0.2686]])请计算a和b的矩阵乘法

根据您提供的矩阵，计算矩阵 a 乘以矩阵 b 的矩阵乘法结果：

css Copy code

```css
import torch

a = torch.tensor([[0.3348, -0.1575, -0.0552, 0.4606, -0.4005],
                  [1.2785, -0.0594, -2.4832, -0.1052, -1.5892],
                  [-0.3722, -0.1244, -0.6635, 0.6163, -0.8171],
                  [-1.4294, -0.6973, -0.0049, -0.8030, -1.2963]])

b = torch.tensor([[-1.6391, -1.8575],
                  [1.4069, 0.8627],
                  [0.4147, -0.0830],
                  [0.7041, 0.1294],
                  [-1.6289, 0.2686]])

result = torch.matmul(a, b)
print(result)
```

Regenerate response

运行代码后，会得到矩阵乘法的结果：

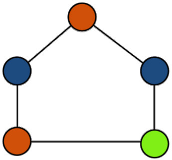
Send a message.

Correct Answer:

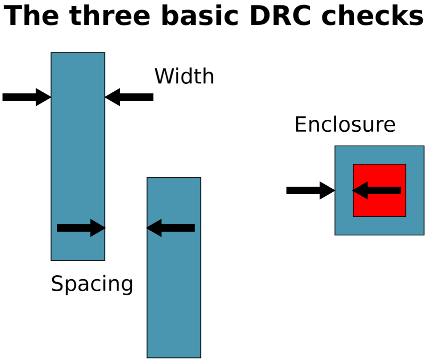
$$\begin{bmatrix} 1.2766, & 0.0329 \\ 6.1799, & -2.6547 \\ -0.8886, & 0.1450 \\ 0.8300, & -0.1394 \end{bmatrix}$$

Challenges to AI

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

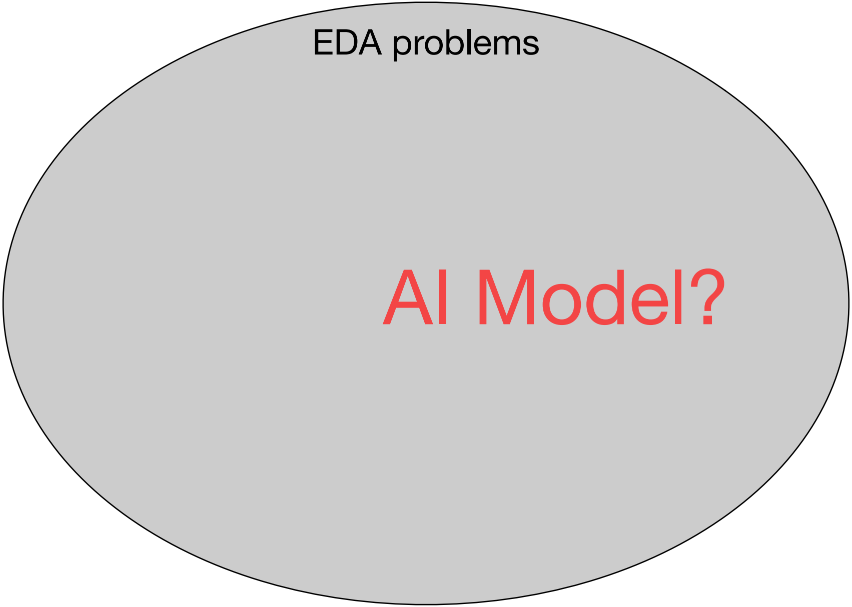


(a) Graph Coloring

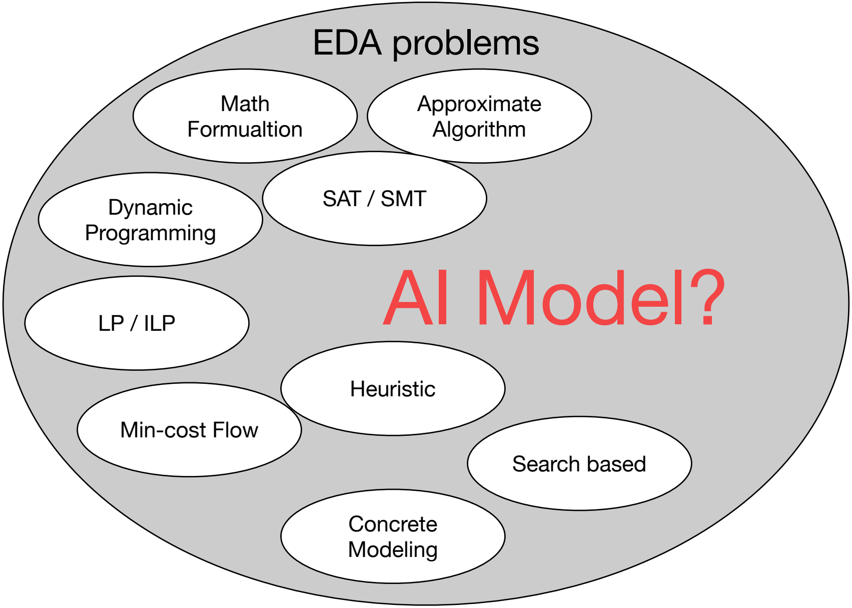


(b) Design Rule Checking

Know the Boundary!



Know the Boundary!



THANK YOU!