# Intelligent Circuit Design and Implementation with Machine Learning

## PhD Dissertation Defense

Zhiyao Xie

Advisor: Yiran Chen

Department of ECE, Duke University

Feb 14, 2022



Duke UNIVERSITY

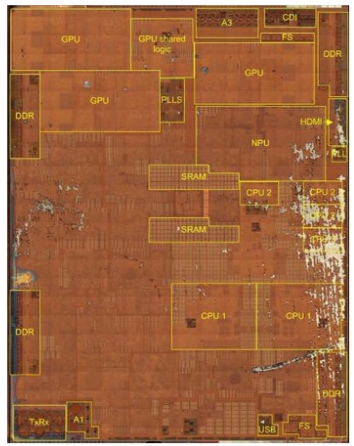
# Electronic Devices are Everywhere



*These images are found in public domain

# Designers Try to Deliver Generational Gains

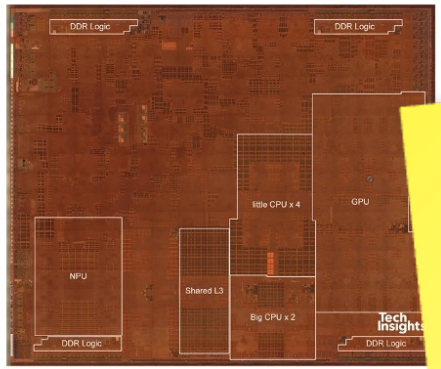
**iPhone 8, X**




Apple A11

10nm
4.3 B trasistors

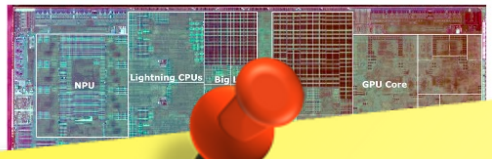
**iPhone XS, XR**




Apple A12

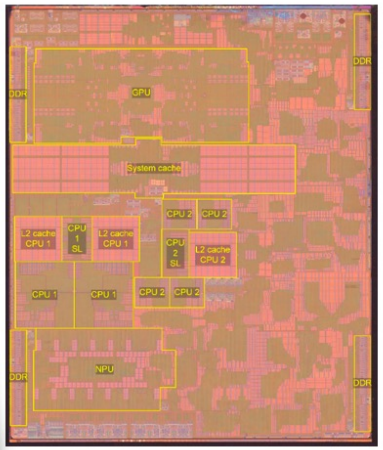
7nm
6.9 B trasistors

**iPhone 11**




**Looks good!**
**Any challenges?**

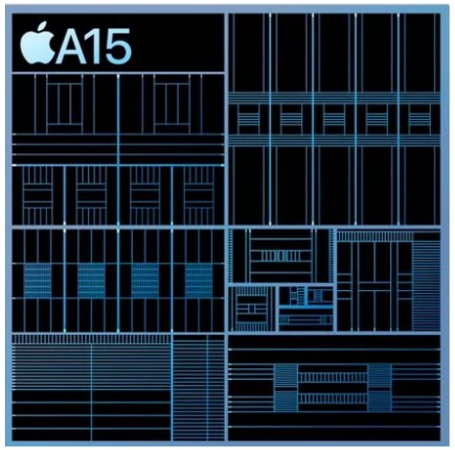
**iPhone 12**




Apple A14

5nm
11.8 B trasistors

**iPhone 13**




Apple A15

5nm
15 B trasistors

Increased **integration** and **architecture** improvements

*Source: TechInsights Inc.

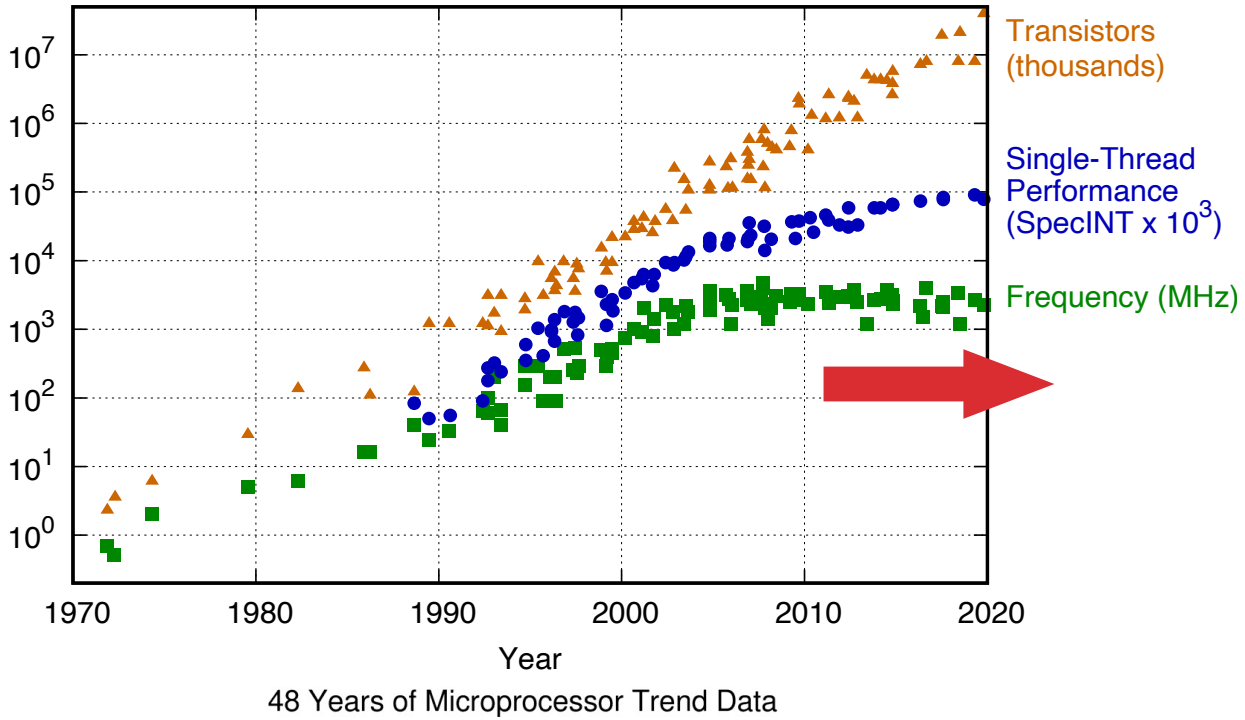


Duke UNIVERSITY

# Chip Design Challenges

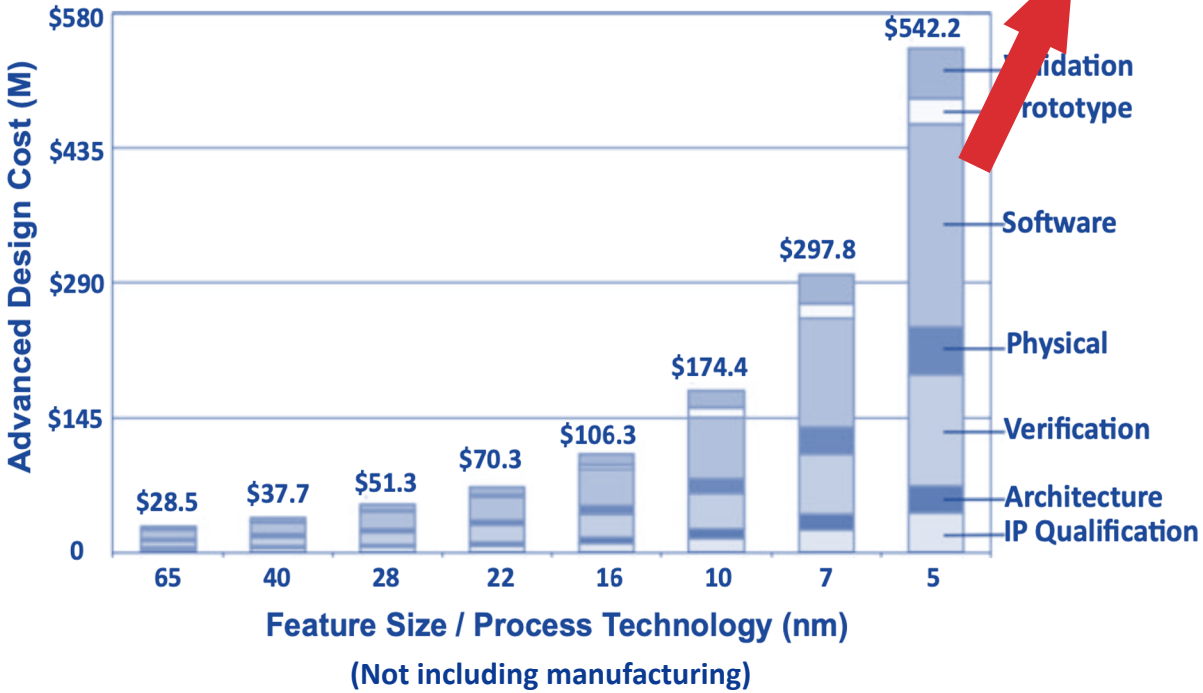
Diminishing performance gain and increasing design cost

## Per-Core Performance Gain is Diminishing




Partially collected by M. Horowitz et al. Plotted by Karl Rupp, 2020

## IBS Design Cost is Skyrocketing



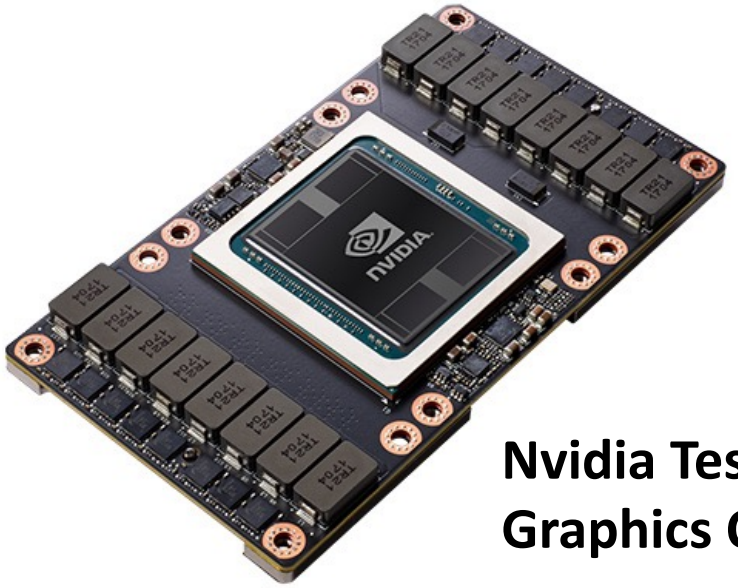

**International Business Strategies, 2020**



Duke UNIVERSITY

# Chip Design Challenges

Not only costly, also long turn-around time




**Nvidia Tesla V100 Graphics Card**

It took **several thousand** engineers **several** years to create, at an approximate development cost of **$3 billion**. – Jensen Huang, CEO of Nvidia

**Nvidia GPU Technology Conference (GTC), 2017**

# This is Real Problem!

## Challenges at advanced node

- **Pressure** from IPC and frequency
- Peak power keeps **increasing**
- Power delivery technique is
- **Increasing** design rules to m
- **Increasing** wire parasitics, ca
  wire delay and noise
- ……


## Inefficient chip design methodologies



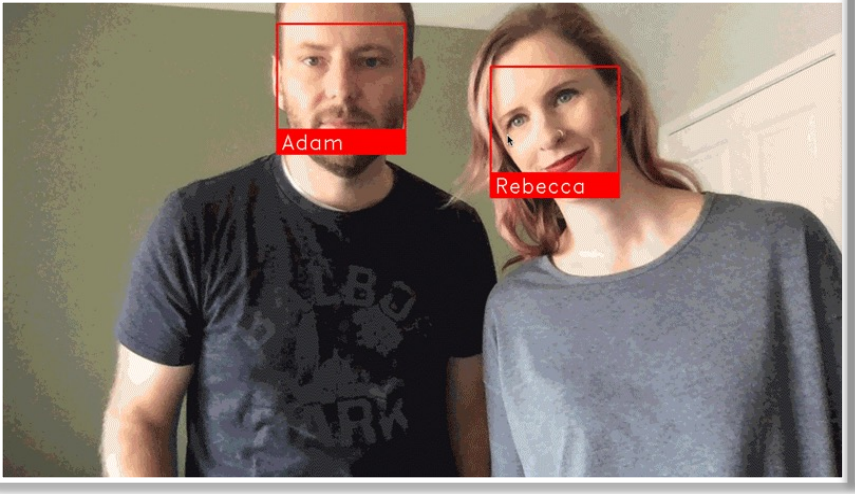

**For one Arm CPU core with ~3 million gates**

- e power simulation takes **~2 weeks**
- ation in physical design take **~1 week**
- s **repeatedly** constructed from scratch
- olutions rely on designer **intuition**
- ……

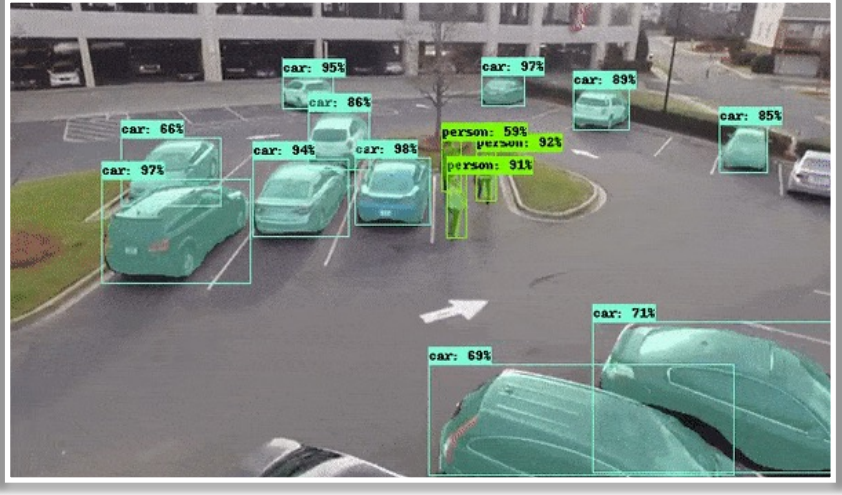
Source: The Kirin 990 SoC. TechInsights Inc.



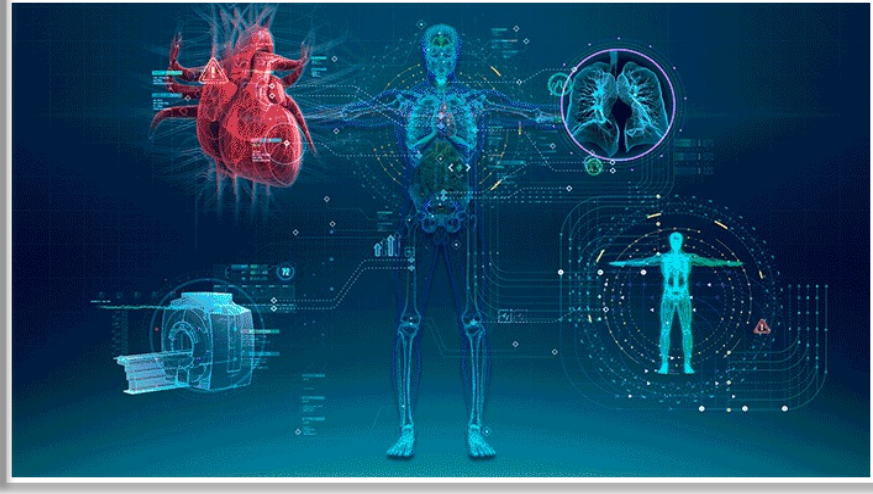
Duke UNIVERSITY




https://github.com/ageitgey/face_recognition




https://towardsdatascience.com/using-tensorflow-object-detection-to-do-pixel-wise-classification-702bf2605182



http://matclinic.com/2017/05/18/the-team-behind-the-future-of-ai-in-healthcare/

Self-driving Cars

Autopilot Drone

Robots

Smart Home


Health Monitor

Personal Assistant

Manufacturing

Smart Grid

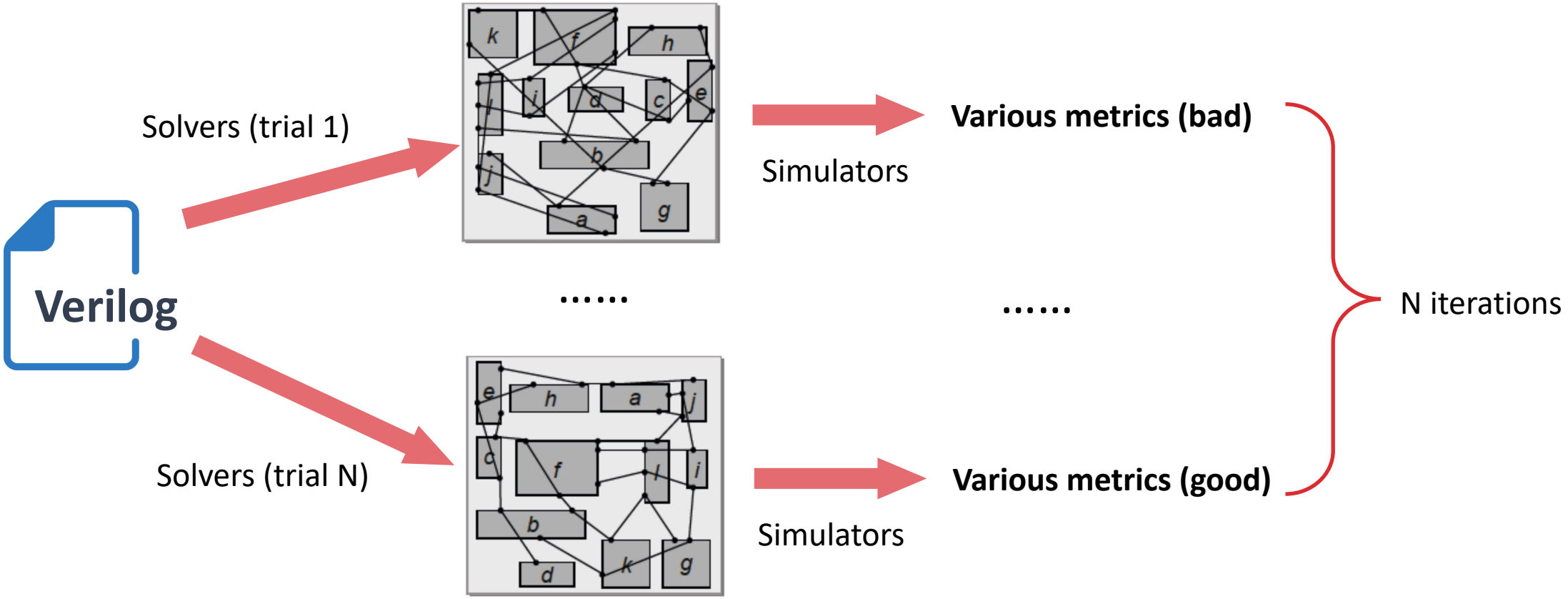
Financial Service

HPC

Security

Gaming

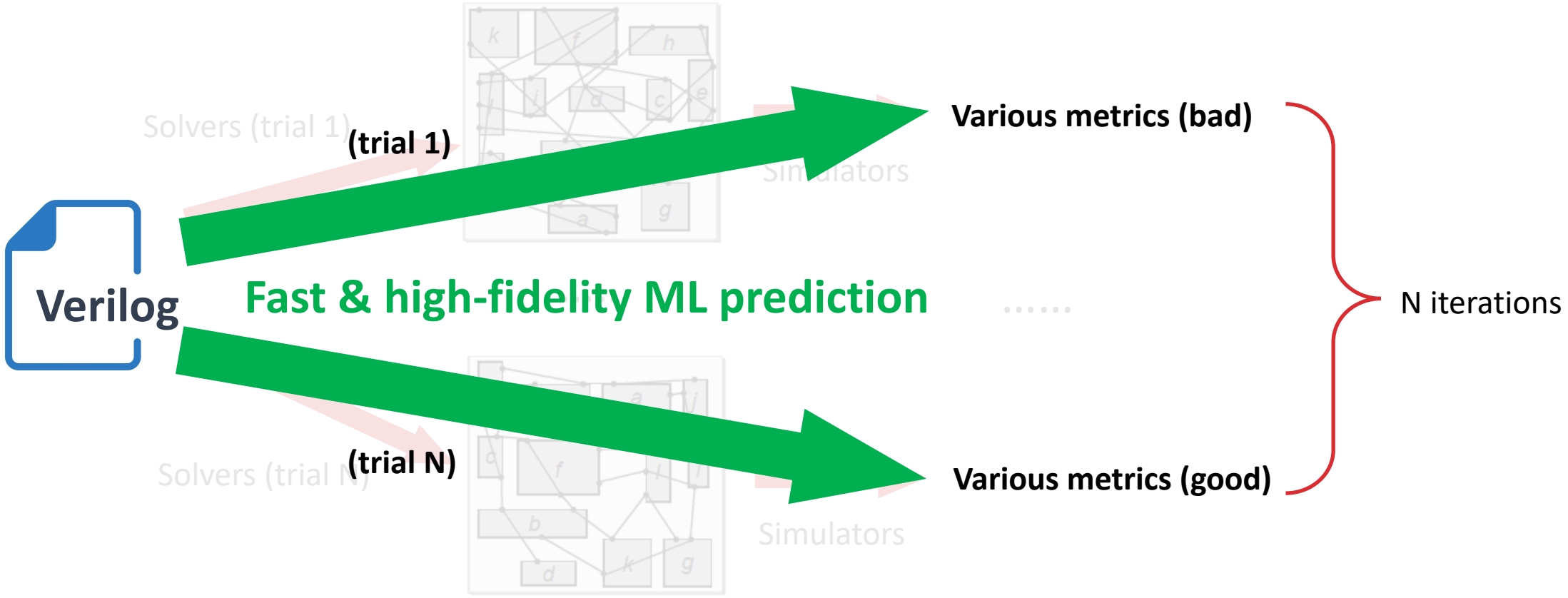
# Why ML/Intelligence Helps Circuit Design? An Example




- Producing solutions **repeatdly from scratch**

*Source: Kahng et al., VLSI physical design

# Why ML/Intelligence Helps Circuit Design? An Example




- Producing solutions **repeatdly from scratch**
- Why not learn from prior solutions?

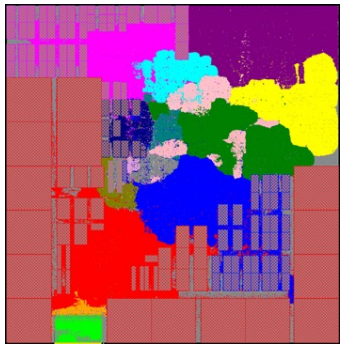


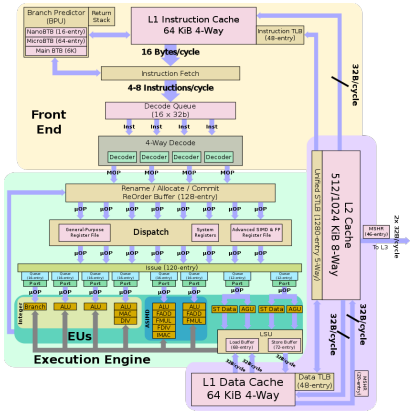
Duke UNIVERSITY

# Simple Plug-in and Use of ML Engines?

**Images**

- 100s * 100s pixels
- No extra information
- Any human can tell the label
- Data is everywhere
- ……






**Circuits (Arm Neoverse N1 CPU core)**

- Millions of connected components
- 100s GB of raw information
- Need simulations to get the label
- Data is hard to get
- ……

## Innovative Customized Solutions are Desired!

# Many Excellent Exploration in Academia and Industry

**Increasing research efforts on ML for chip design automation**

UT Austin

UCSD

CUHK

TAMU

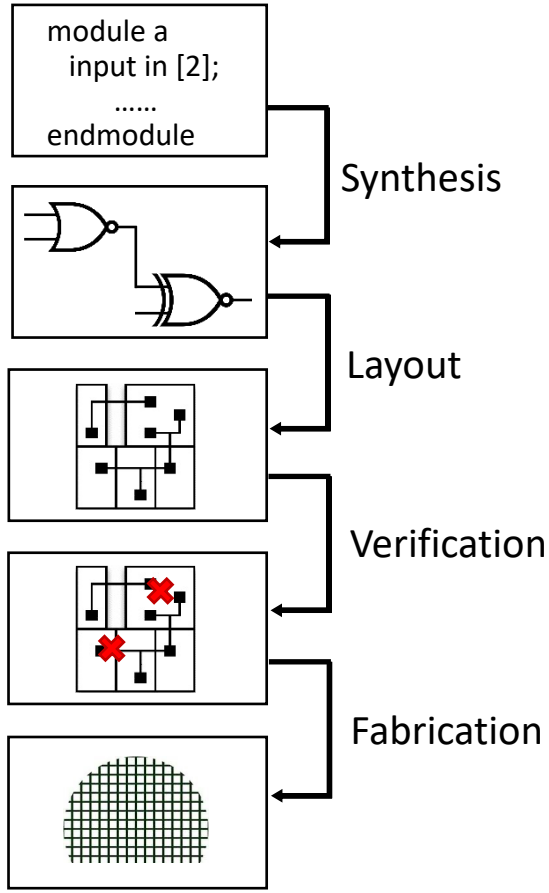
Duke

Cornell

Google

Nvidia

GaTech




**ML for EDA in commercial tools**

Cadence Innovus™

Synopsys ICC™ II

.....

**ML** for Chip Design

Traditional Chip Design

DARPA Electronics Research Initiative (ERI) – Design

Goal: 24 hours turnaround time & no human



Duke UNIVERSITY

# My Related Works

**PPA**

- **Power**
  - **Power & Power Delivery Challenges**
    [ICCAD'20], [ASPDAC'20],
    [MICRO'21] (Best Paper Award)
- **Performance**
  - **Timing & Interconnect Challenges**
    [ICCAD'20], [ASPDAC'21], [TCAD'21]
- **Area**
  - **Routability Challenges**
    [ICCAD'18], [DATE'18], [ICCAD'21]

**Overall Design Flow Tuning**
[ASPDAC'20]

Covered in this talk

# Outline of The Dissertation

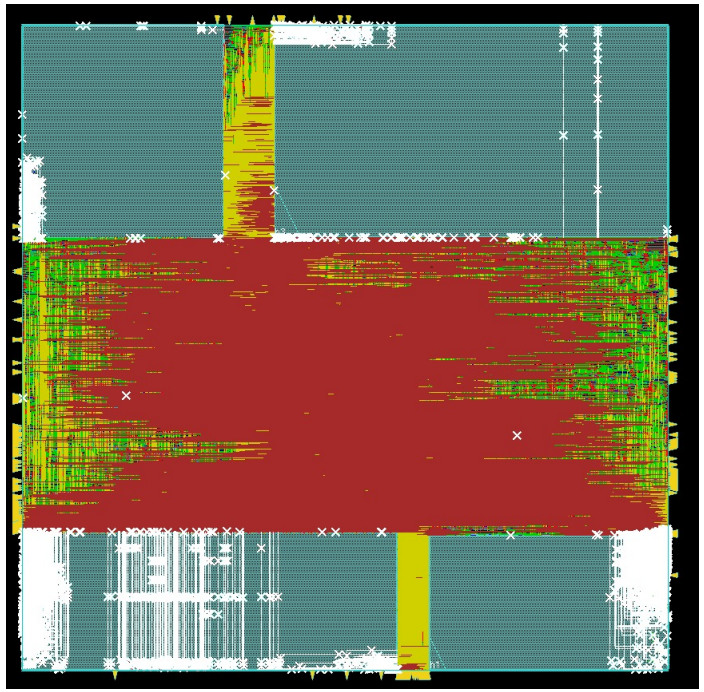
- Case Study 1: Routability Challenges (~10 min)
- Case Study 2: Timing & Interconnect Challenges (~10 min)
- Case Study 3: Power & Power Delivery Challenges (~30 min)

# Case Study 1:

# Routability Challenges

# Routability Background

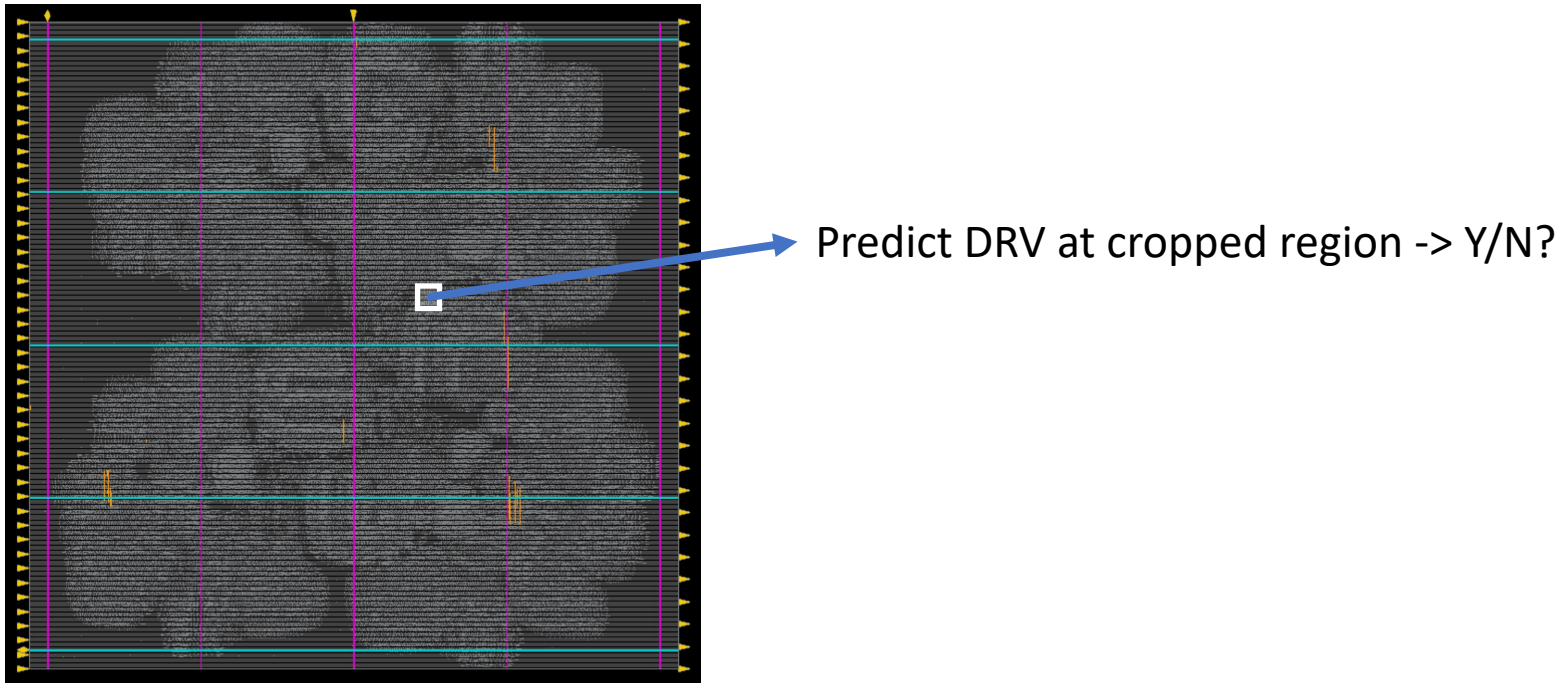
- Design Rule Checking (DRC)
  - Manufacturing requirements
  - Less DRC violations (DRV) -> better routability
- Need DRV mitigation at early stages
  - Requires routability prediction/estimation
- Existing solutions
  - Fast Trial Global Routing (TR) for fast estimation
  - Global Routing (GR) for accurate estimation



DRC violations (white) on circuit layout

# Previous ML-based Routability Estimators

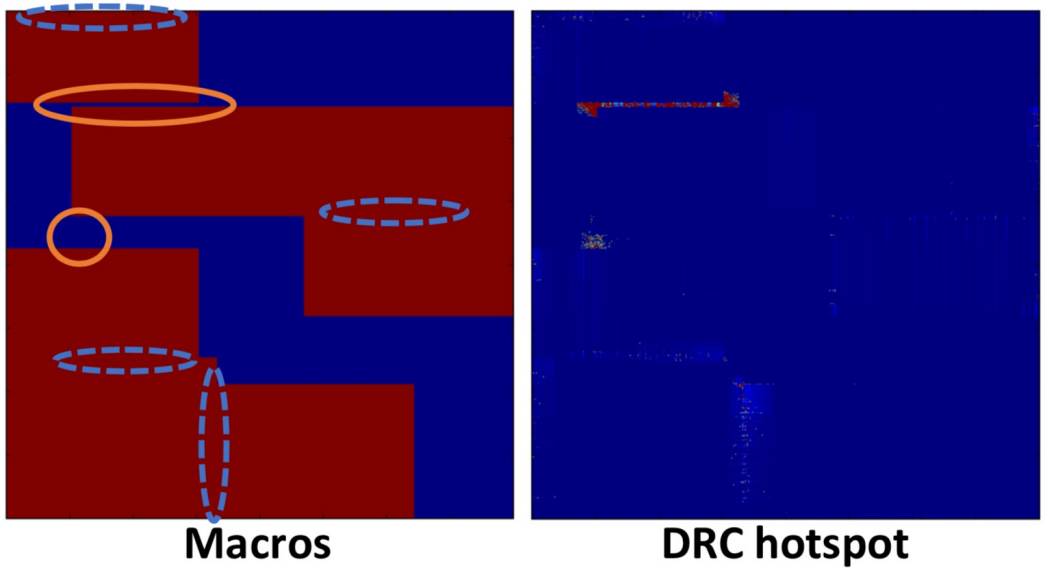
- Previous routability (DRV) estimations
  - ML model on small cropped regions
  - **Limited** receptive field and **missing** global information

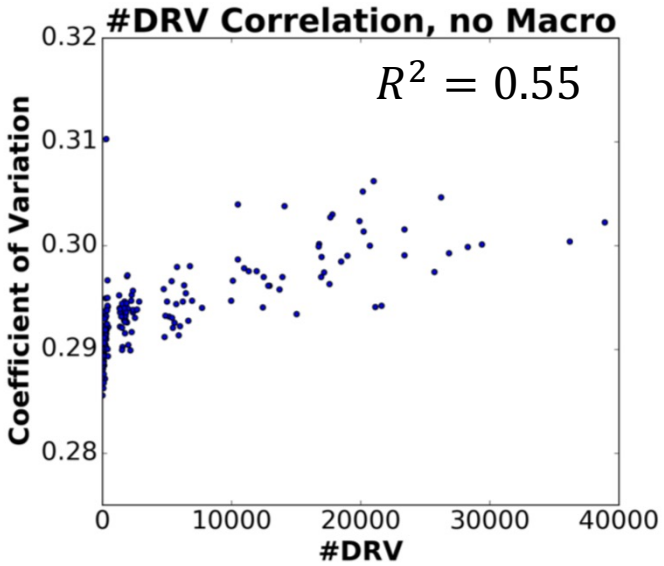


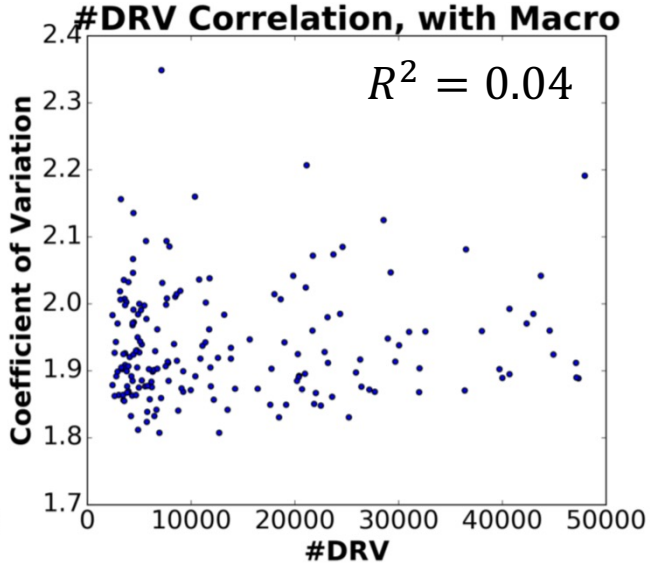

Example: a layout only with cells

# Previous ML-based Routability Estimators

- Previous routability (DRV) estimations
  - ML model on small cropped regions
  - **Limited** receptive field and **missing** global information
  - When macros present, **less resemblance** among different regions of layout



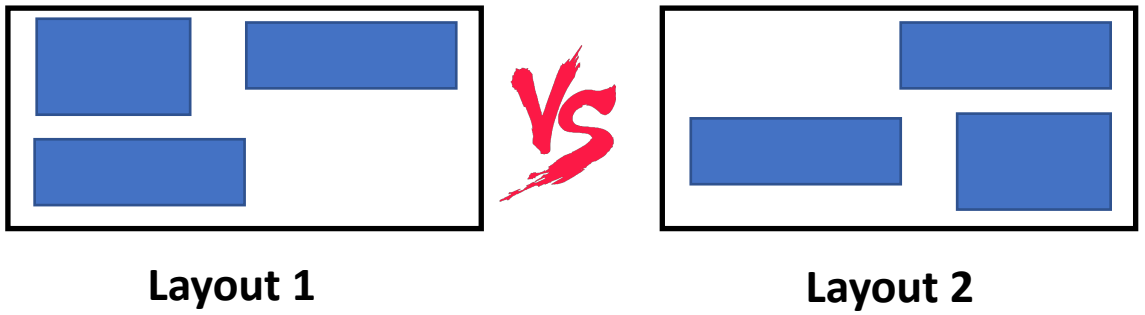



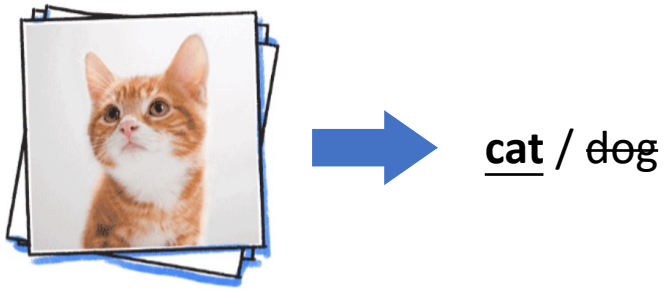



Duke UNIVERSITY

# First Deep Learning Method for Routability Prediction

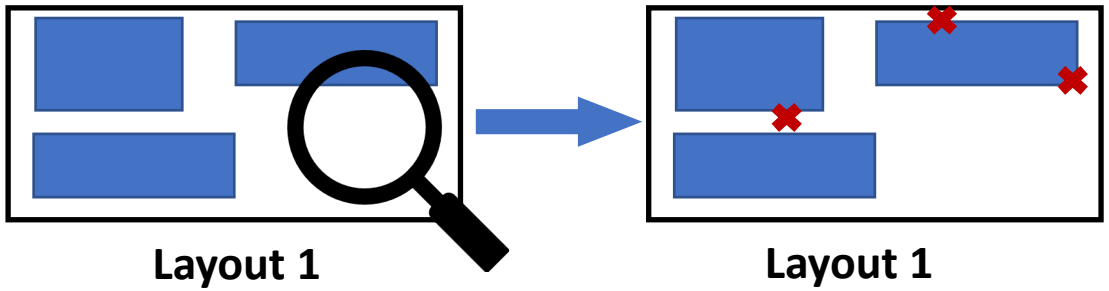
- Task 1: which one will result in less DRV count?

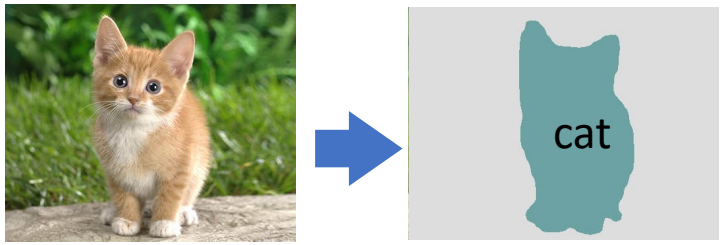



Customized CNN methods

- Task 2: where are DRC violations?



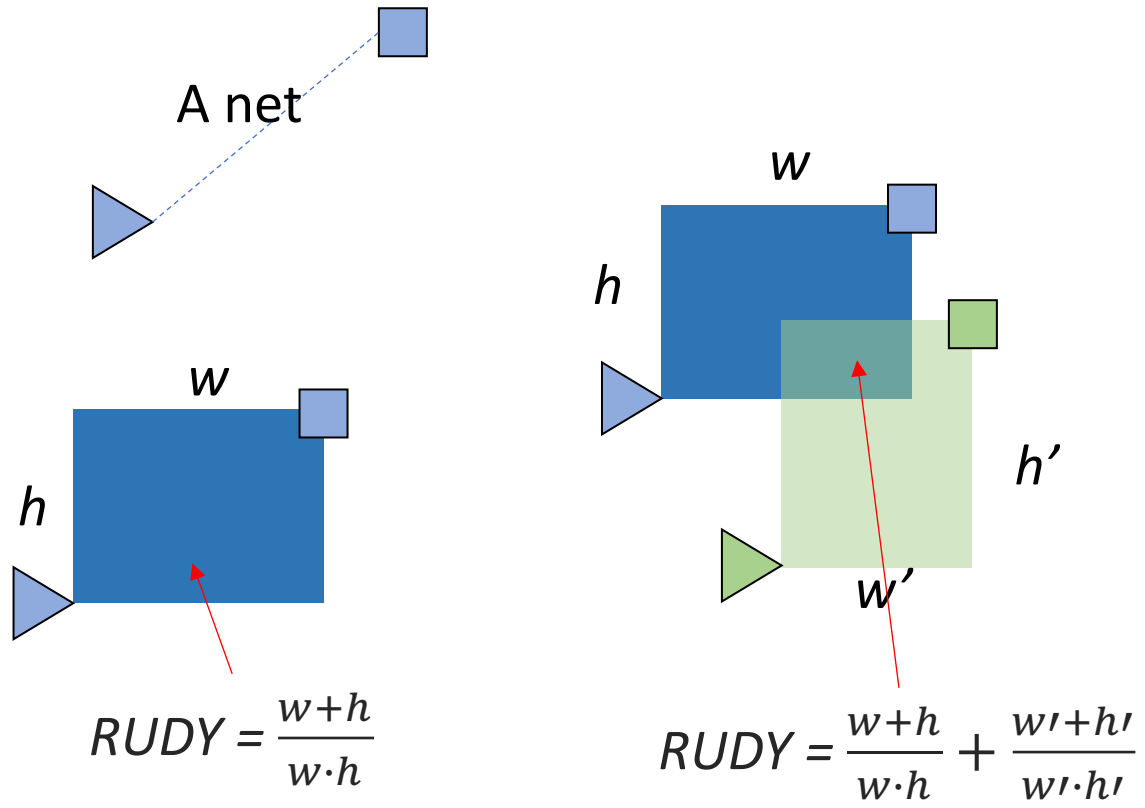

Customized FCN methods



**RouteNet [Z. Xie, et al., ICCAD'18]**

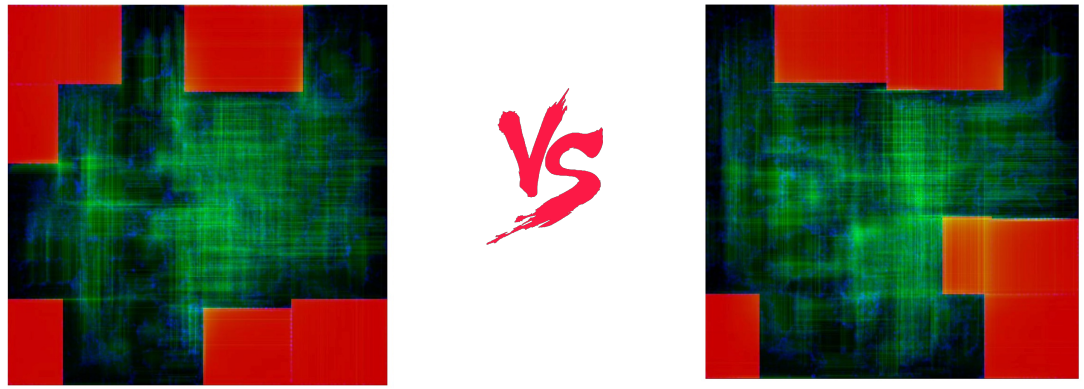
# Features for Rouability Prediction

- Macro:
  - Region occupied by macros
  - Density of macro pins in each layer
- Cell:
  - Density of cells
  - Density of cell pins
- RUDY features (wire density)
  - RUDY distributions
  - RUDY pins
- Congestion report:
  - Trial global routing (TR) congestion
  - Global routing (GR) congestion



# First Deep Learning Method for Routability Prediction

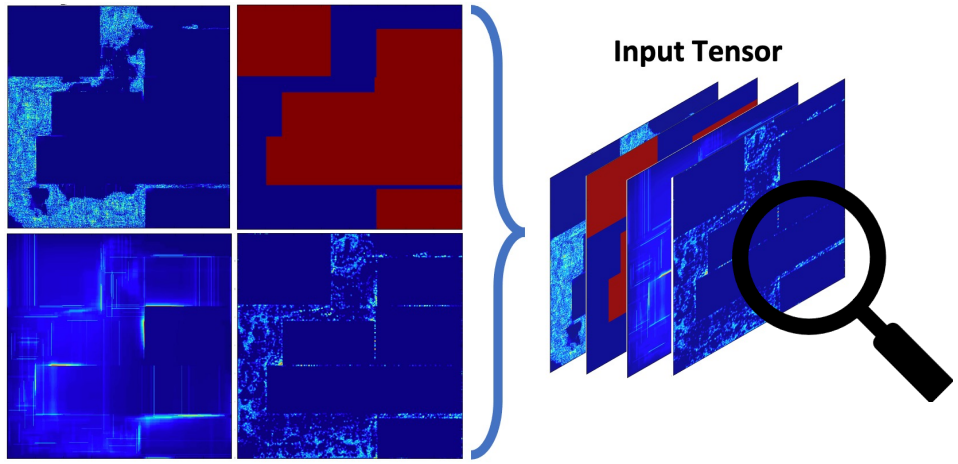
- Task 1: which one will result in less DRV count?




- Requires global routing:
  ~~Hours~~ * Number of Layouts

**In seconds, with similar accuracy**

- Task 2: where are DRC violations?




- Requires detailed routing
  ~~More hours~~ * Iterations

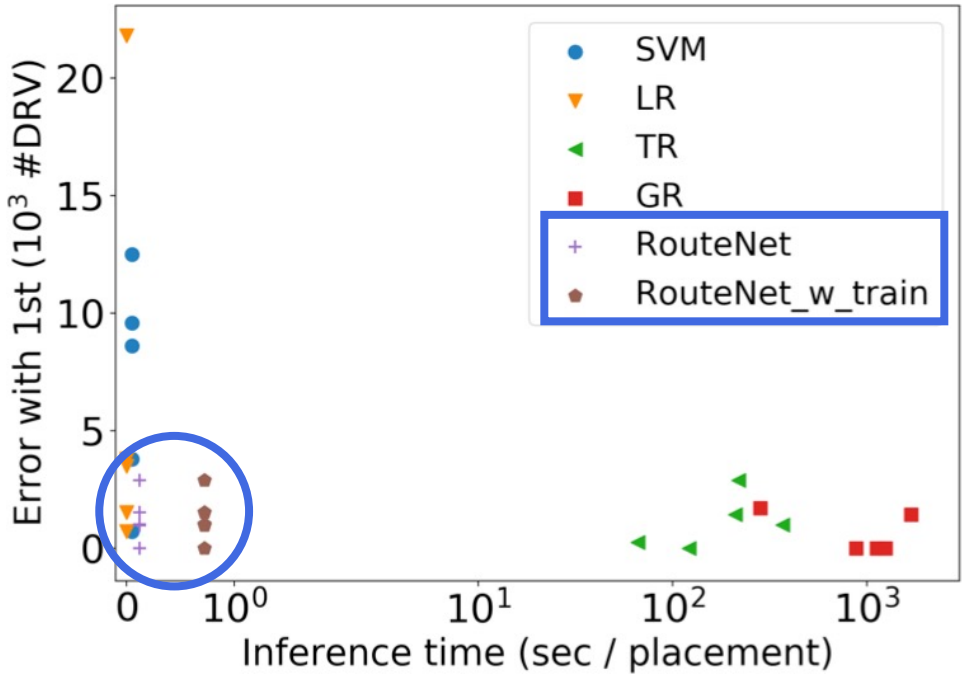
**In seconds, outperform previous works**



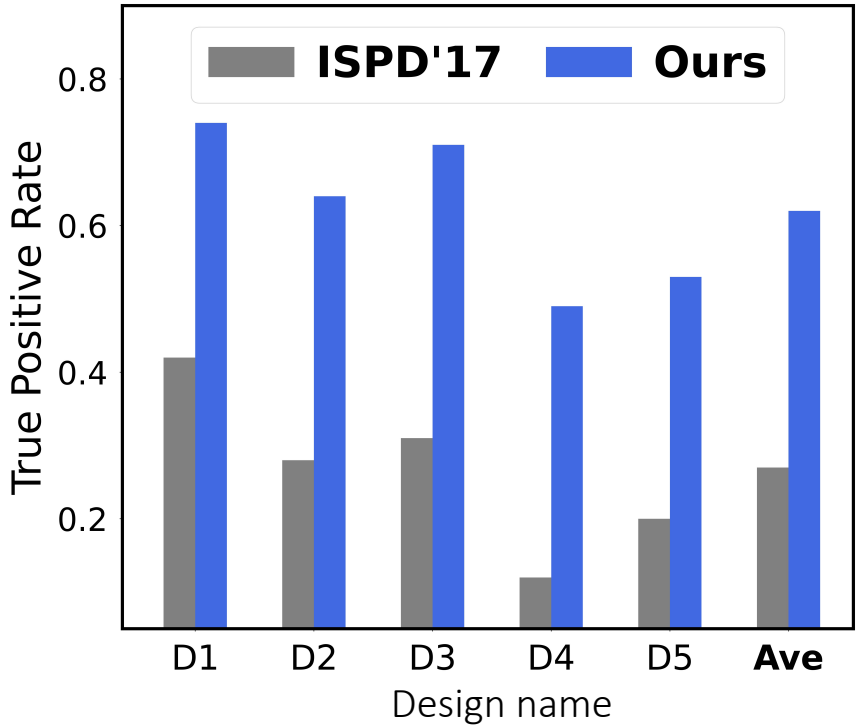
Duke UNIVERSITY

# Experimental Results on Routability

Task 1:




Task 2:



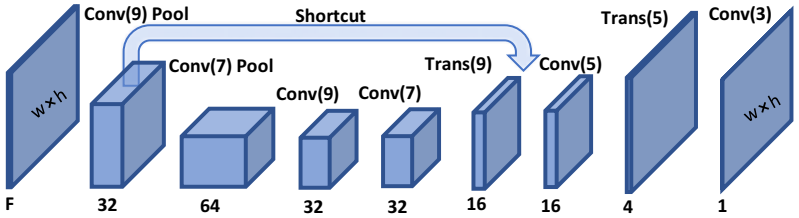

- **Fast** and **accurate** routability prediction at the same time
- **Superior** accuracy over previous work



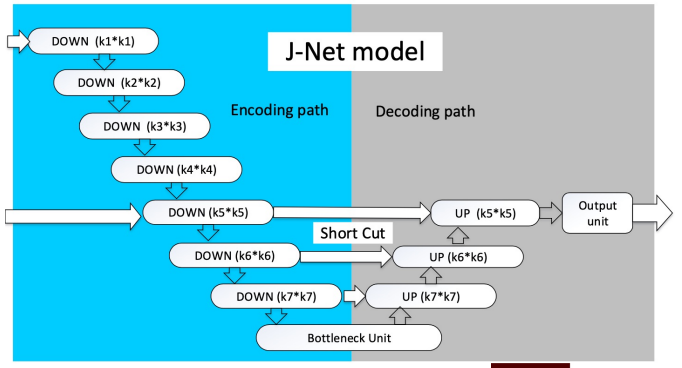
Baseline:
Chan et al. Routability Optimization for Industrial Designs at Sub-14nm Process Nodes Using Machine Learning. In ISPD'17.

Duke UNIVERSITY

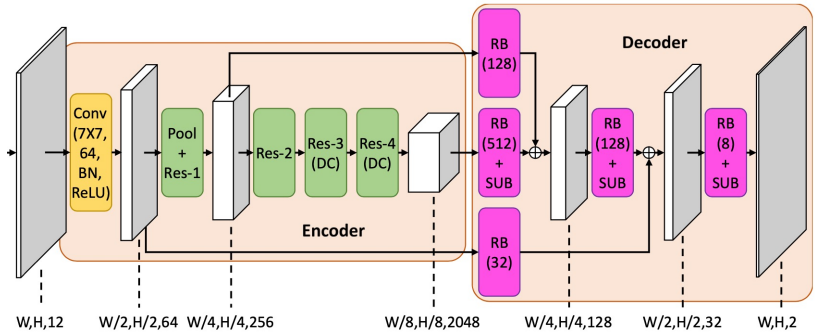
# Many Excellent Deep Learning Methods



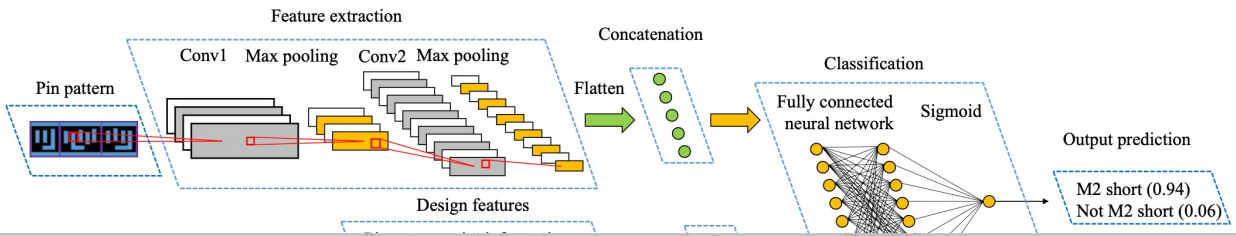

**RouteNet [Xie, et al., ICCAD'18]** Duke UNIVERSITY



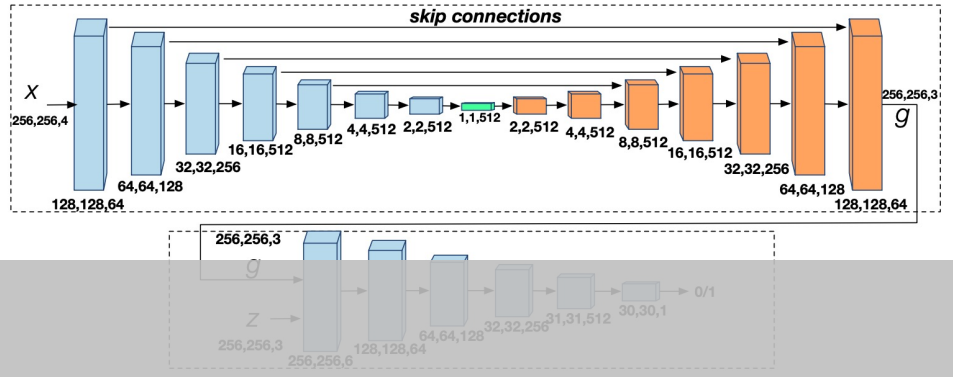

J-Net [Liang, et al., ISPD'20]




PROS [Chen, et al., ICCAD'20]

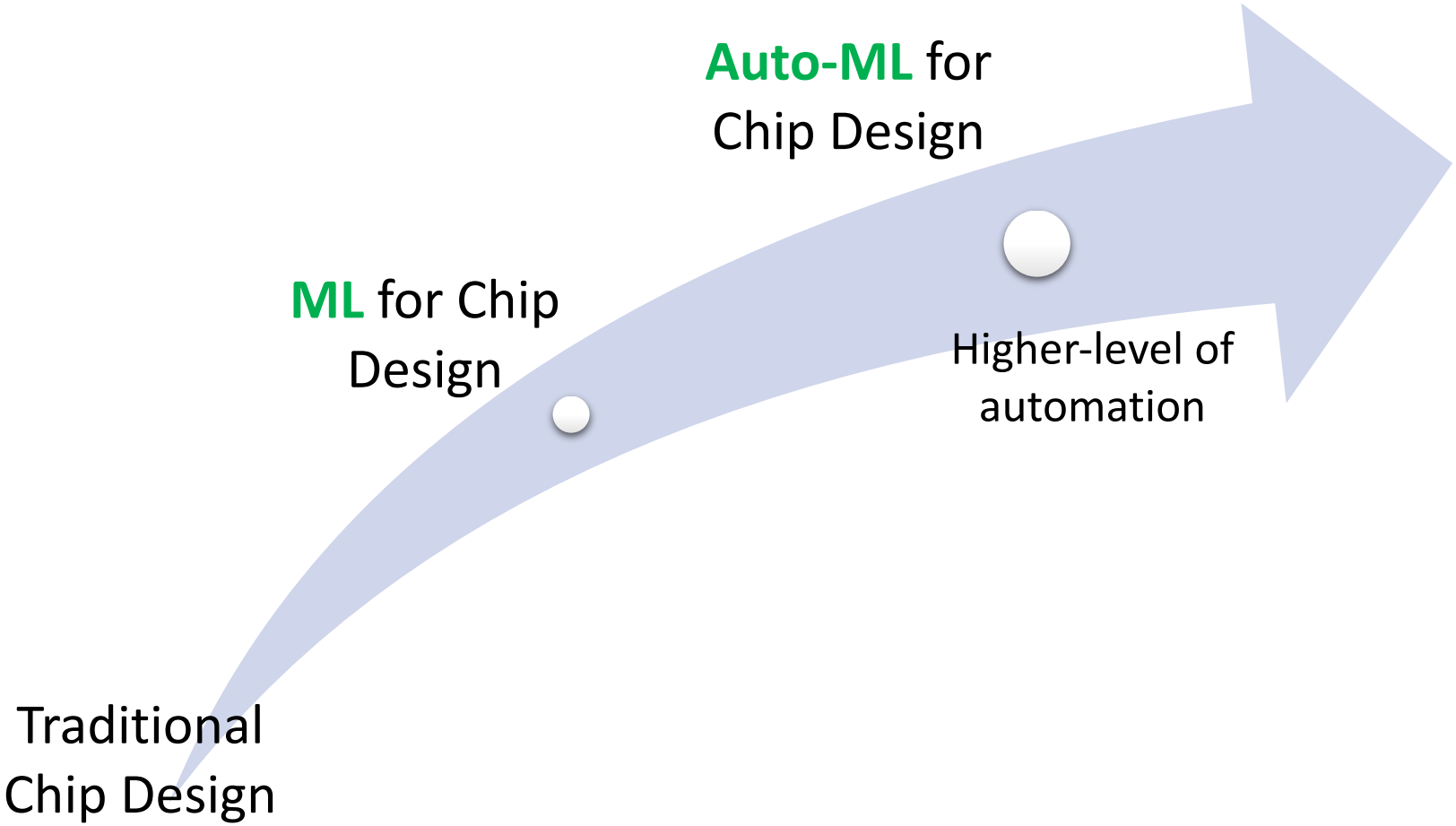



[Yu, et al., DAC'19]




# Tremendous Engineering Efforts Required!

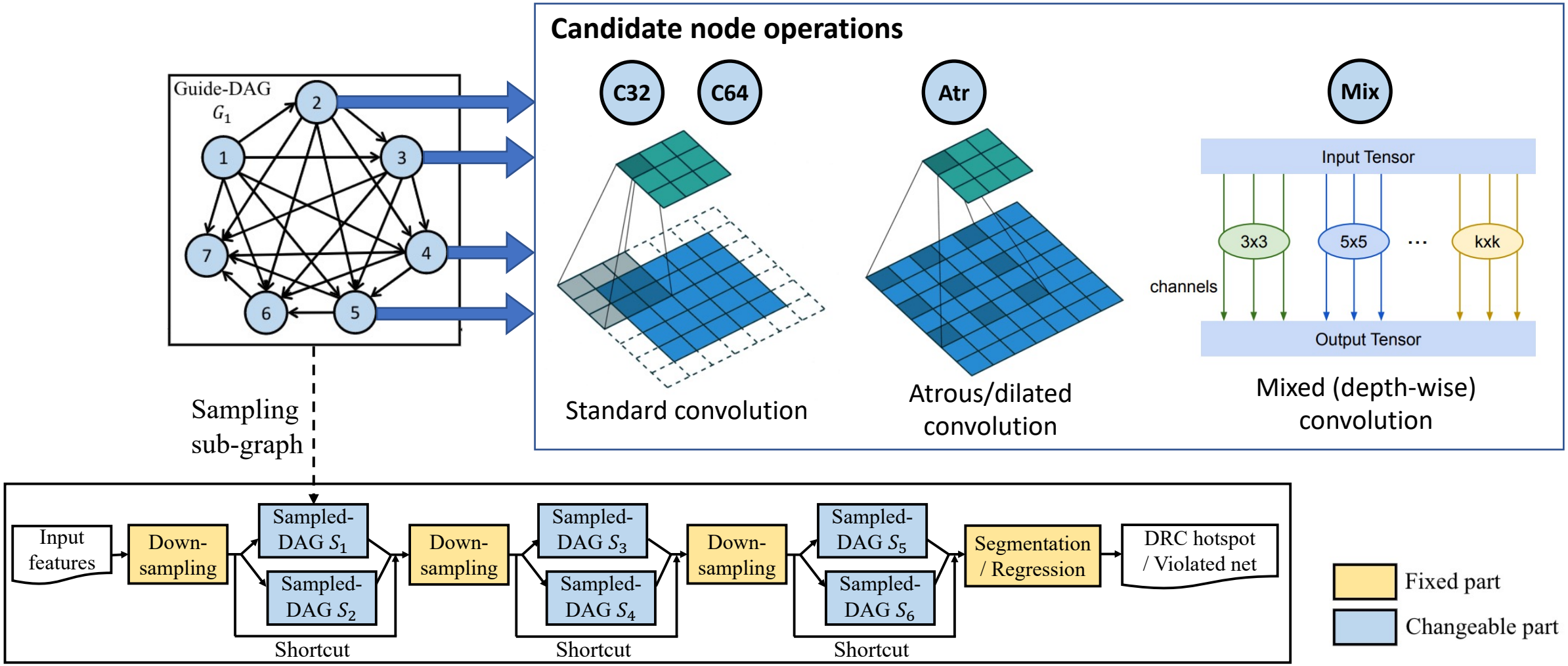
# What I Believe We Should Target





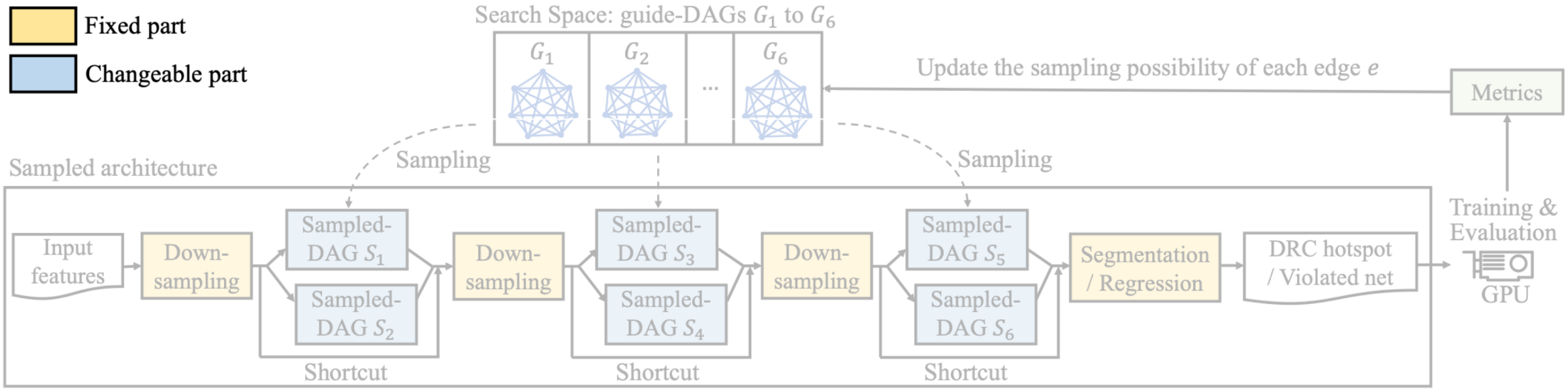

Duke
UNIVERSITY

# Automatic Estimator Development – Search Space




[C. Chang, et al. ICCAD'21]

# Automatic Estimator Development – Searching Algorithm

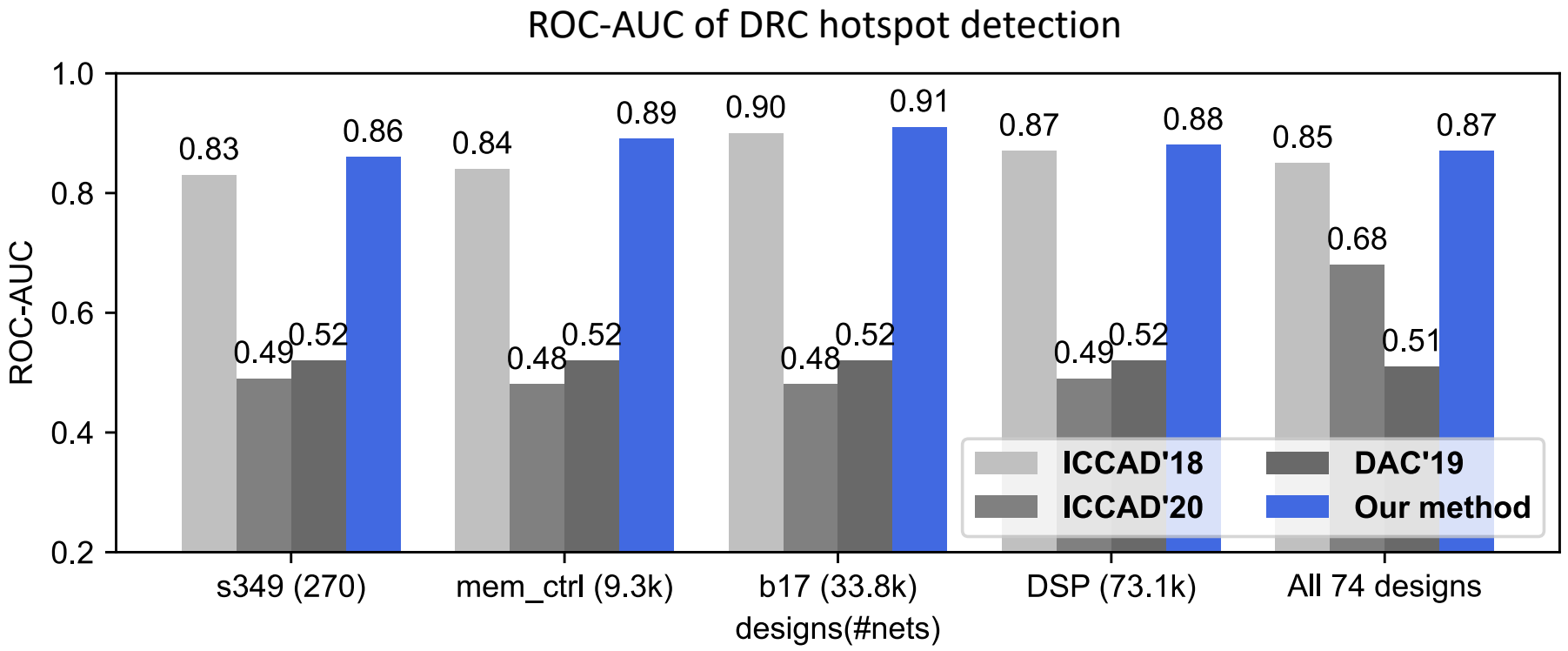



**In each iteration:**

1. Sample from the completely-ordered graph ($G_i$) to get ($S_i$)
2. Evaluate the sampled model by training and testing
3. Update the sampling probability by evaluation result

# Results from Automatically Developed Estimator

- Developed **without human** in **one day**
- **Outperforming** RouteNet [ICCAD'18], cGAN [DAC'19], PROS [ICCAD'20]




Baseline:


Xie et al. RouteNet: Routability prediction for mixed-size designs using convolutional neural network. In ICCAD'18.

Yu et al. Painting on placement: Forecasting routing congestion using conditional generative adversarial nets. In DAC'19.

Chen et al. PROS: A plug-in for routability optimization applied in the state-of-the-art commercial EDA tool using deep learning. In ICCAD'20.



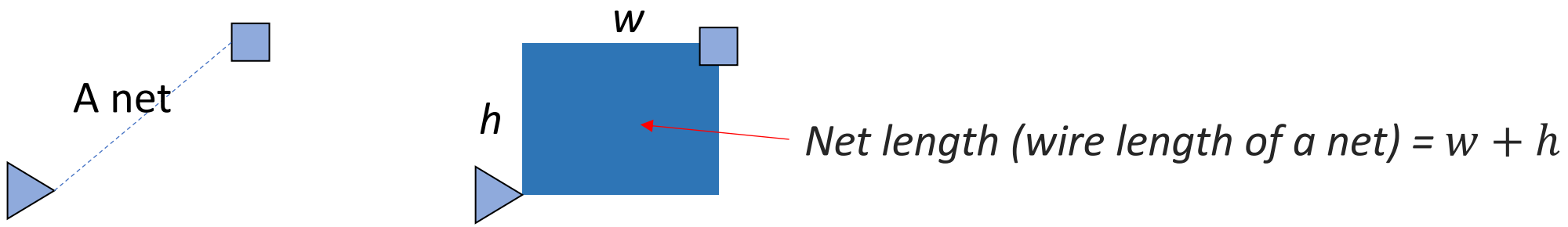

Duke UNIVERSITY

# Case Study 2:

# Timing & Interconnect Challenges

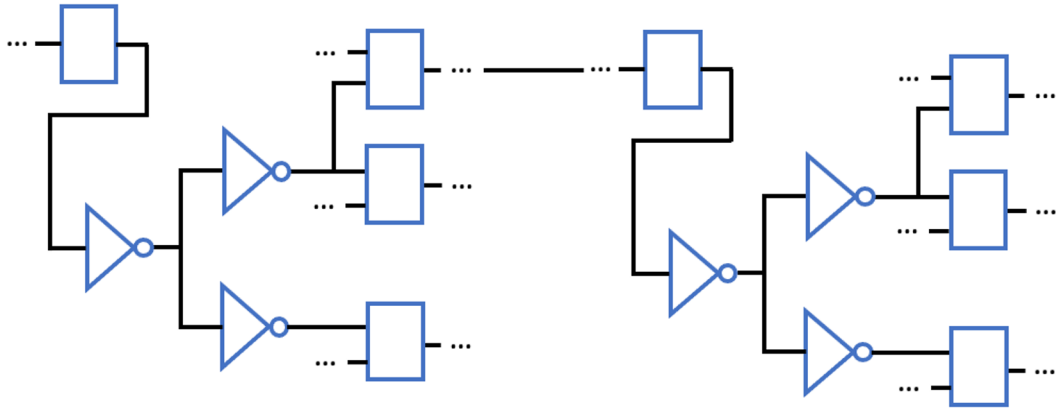
# Interconnect Estimation on Netlist is Inaccurate

- Net length prediction is desired at early stage
  - Interconnect is a dominating factor for power & performance
  - RC of metal wires proportional to net length
  - Net length not explicitly quantified or optimized until placement
- Trend in EDA industry: improve predictability at early stage
  - Physical-aware synthesis with consistent EDA engines in the flow
  - This is **still time consuming**



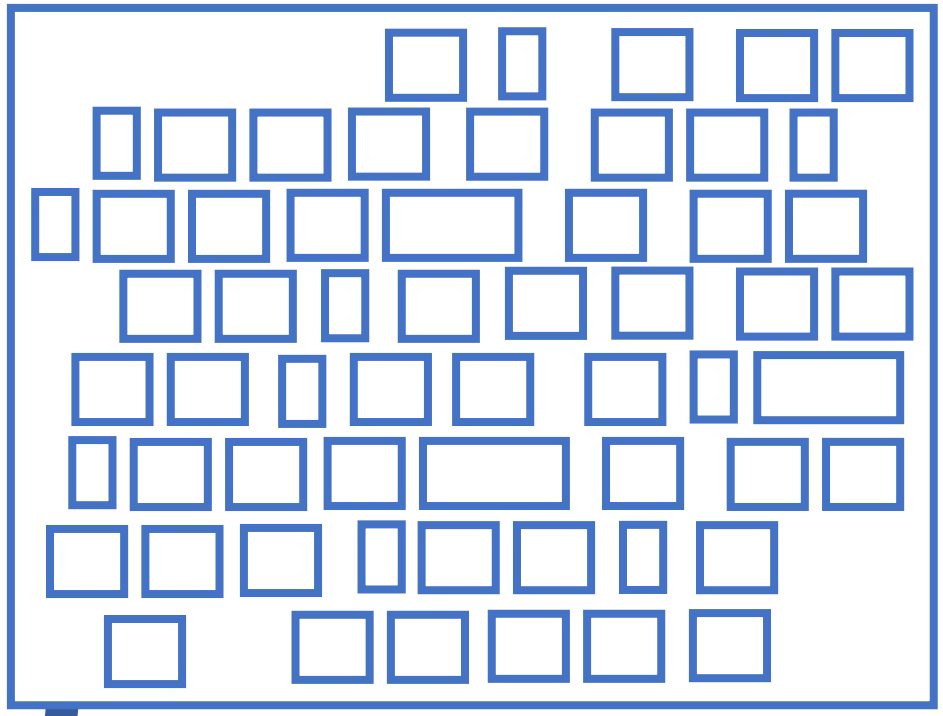
# Timing Estimation on Netlist is Inaccurate

**Gate-level Netlist**



**Placement**
**Hours-days**

**Circuit Layout**



**Absent** net length information
**Inaccurate** timing estimation

Edit netlist if necessary

**Better** net length information
**Better** timing estimation

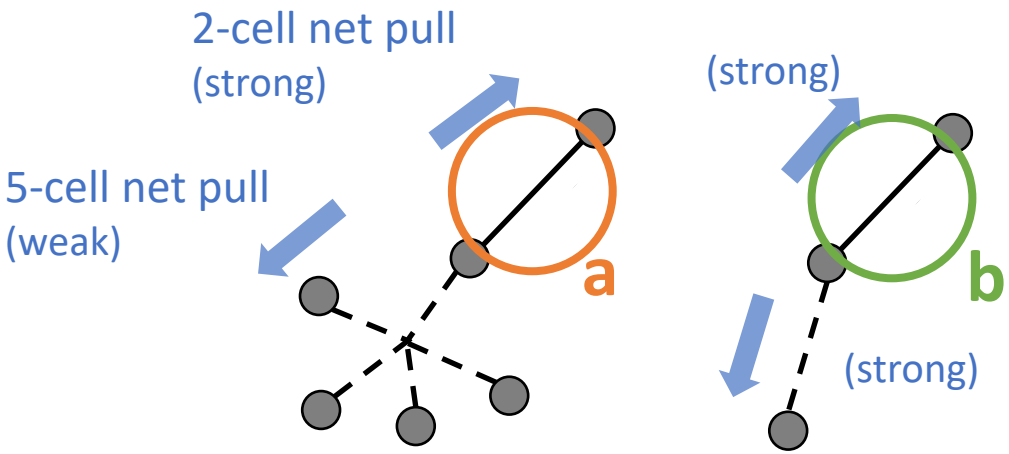
**[Z. Xie, et al. ASPDAC'21]**
**[Z. Xie, et al. TCAD'22]**

# Previous Works: Early Net Length Prediction

- Previous works **lack global information of the whole netlist**

Cell  Connection

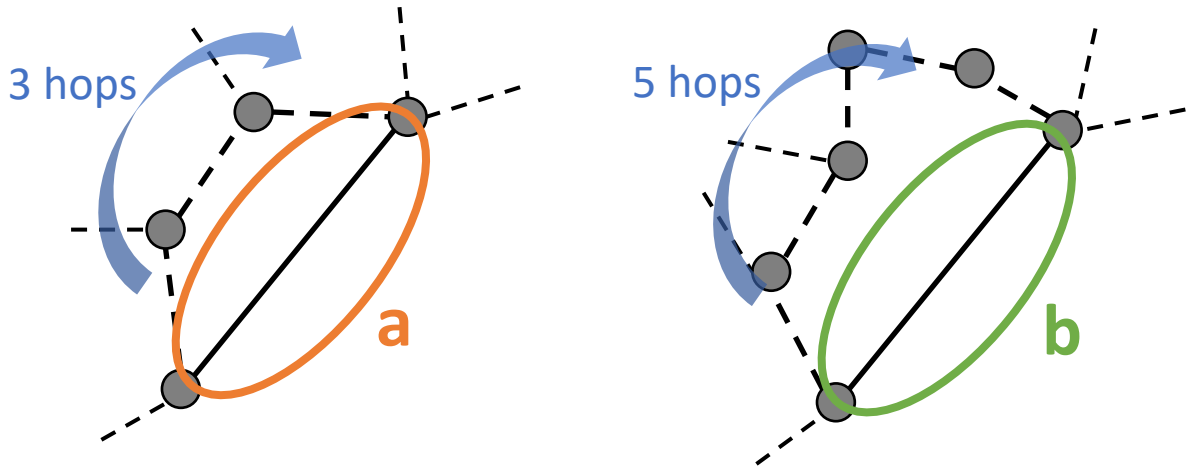
Prior work 1 [DAC'03]




Predict: Net-length **a** < Net-length **b**

**Compares 'pulling strength'**

Prior work 2 [ICCAD'05]

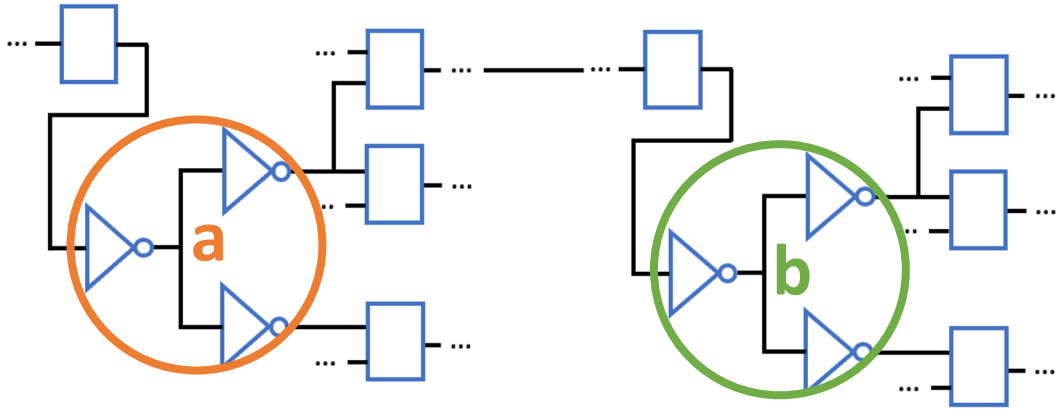



Predict: Net-length **a** < Net-length **b**

**Compares '#hops'**

# Example: Global Information in Net Length Prediction

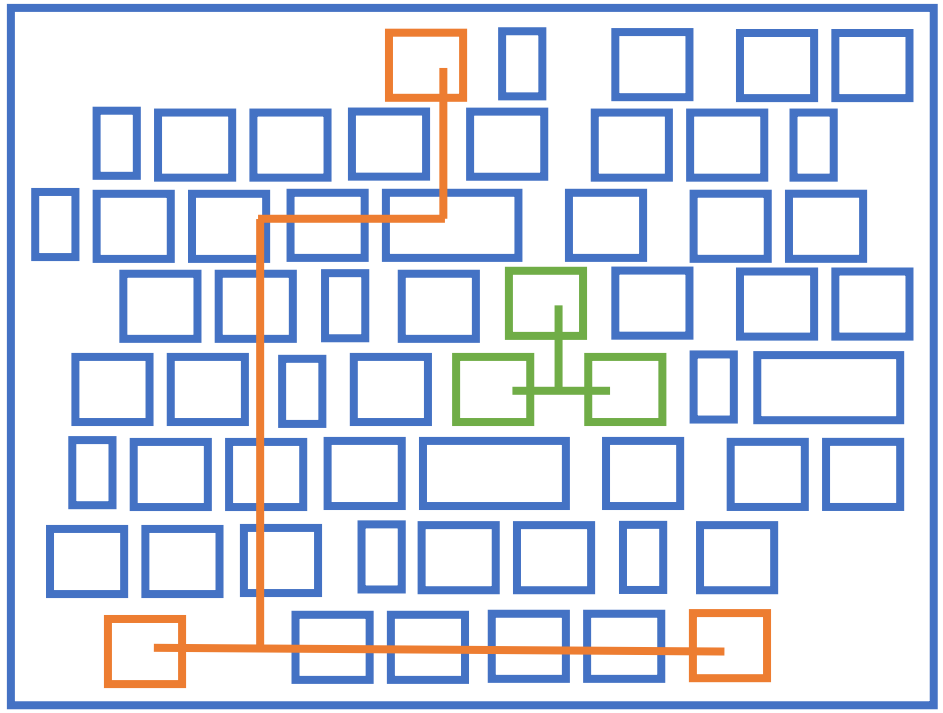
**Circuit Netlist**




Net-length?  Net-length?

**Placement**
**Hours-days**

**Circuit Layout**



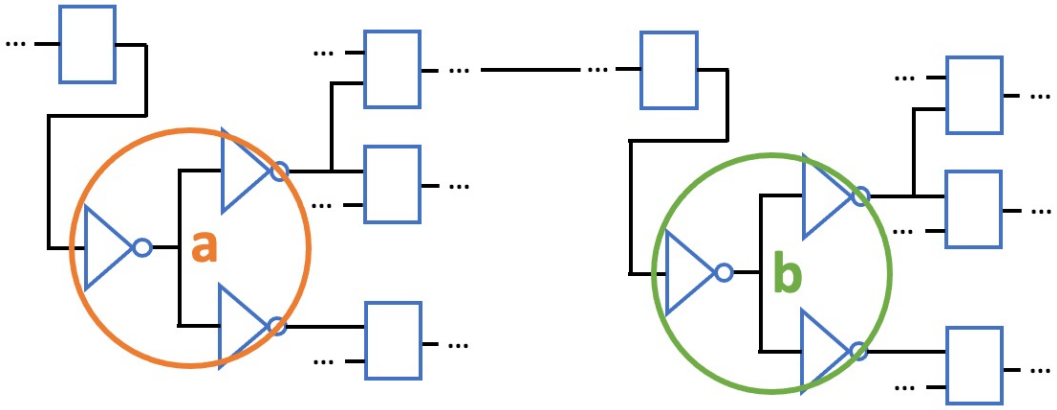
Traditional tools: **Same!**
Previous works: **Similar!**

Net-length **a** >> Net-length **b**
Delay **a** >> Delay **b**

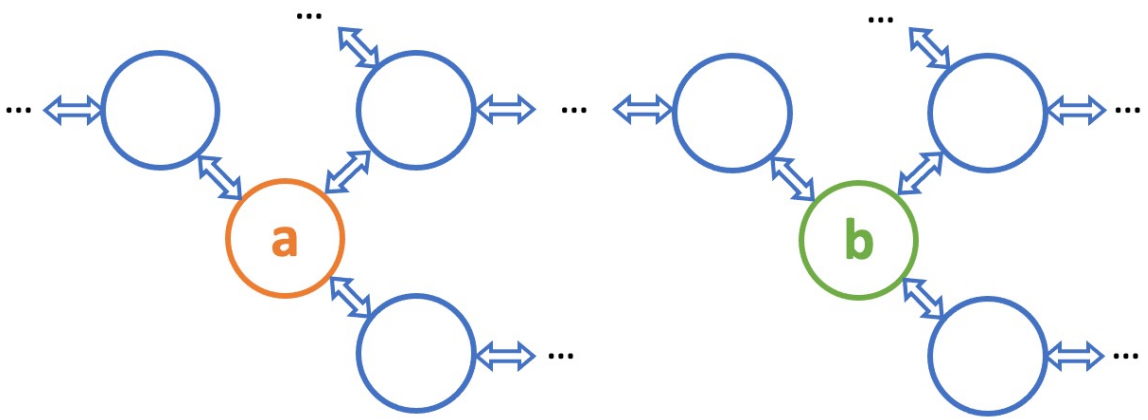


Duke UNIVERSITY

# Our (Fast) Solution to Capture Global Information




Process as Graph




Customized GNN

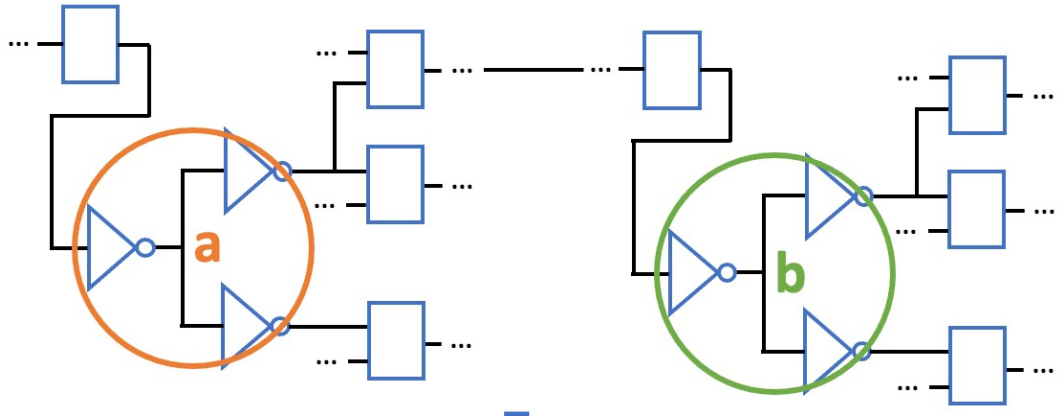
net-length a

net-length b

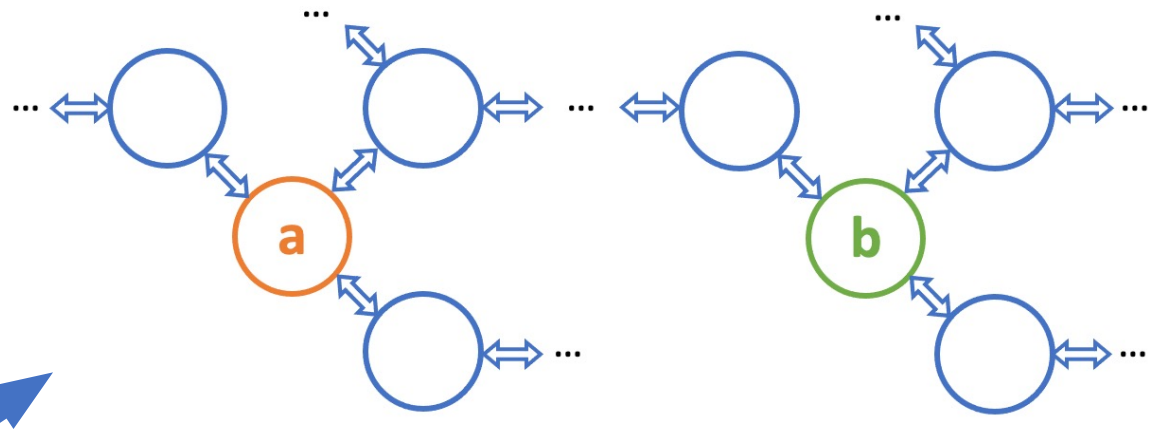
Timing model

**Reduces error** in commercial timing engine

# Our (Standard) Solution to Capture Global Information

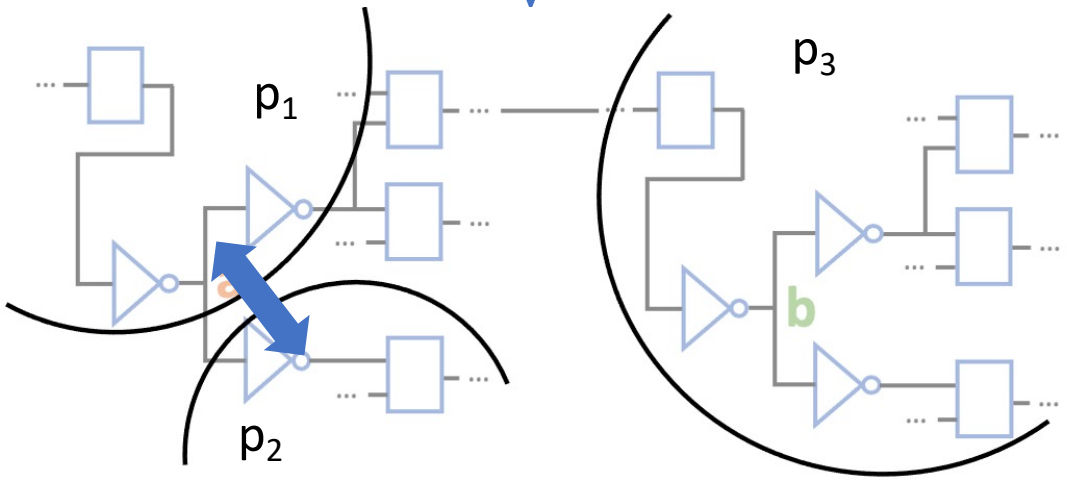



Process as Graph




Partitioning

Info on Edge



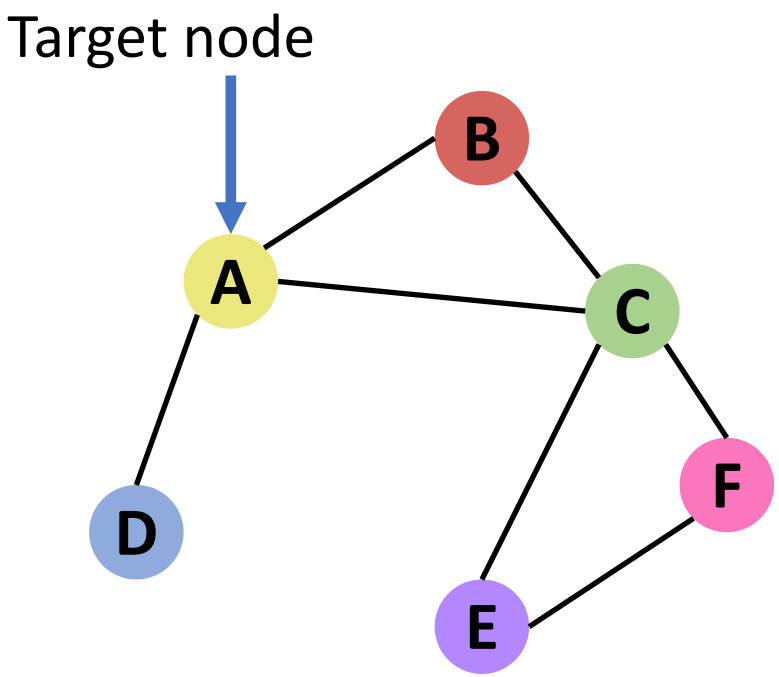

Customized GNN

**Recognize** net-length a >> net-length b

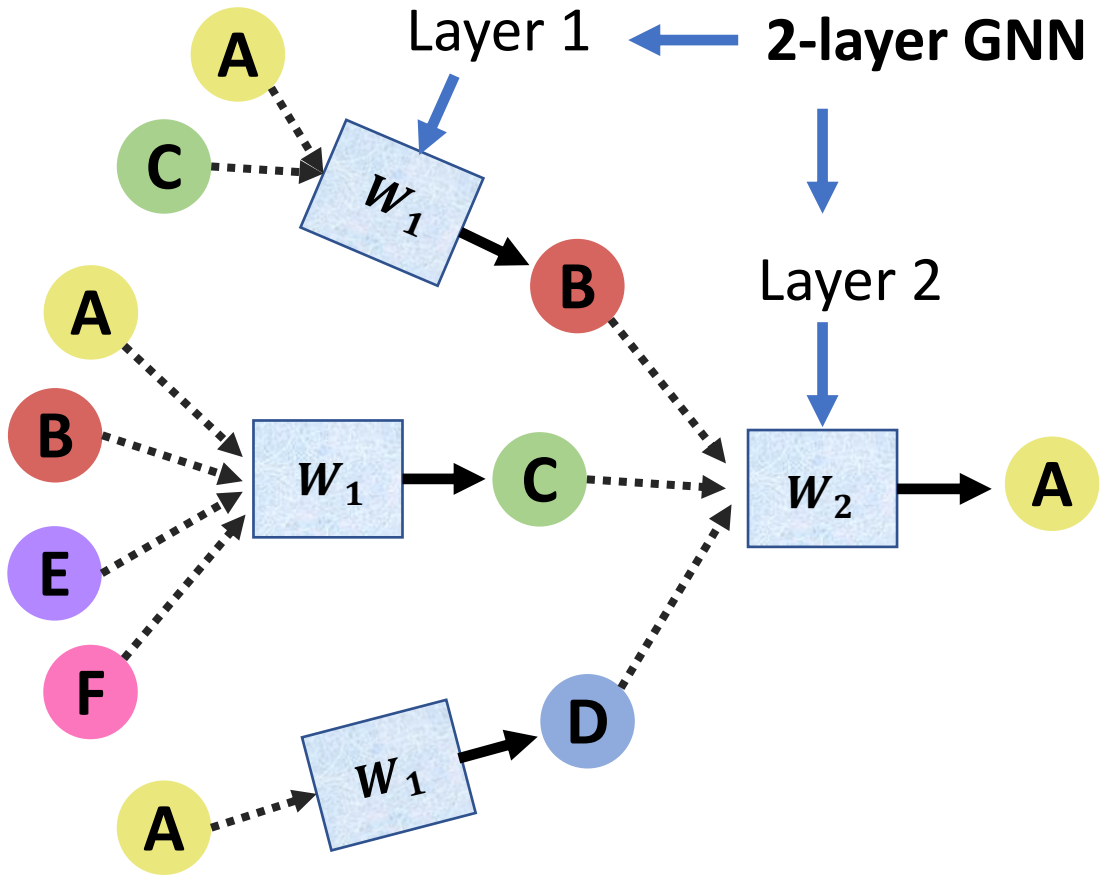
Timing model

**Reduces error** in commercial timing engine

# Graph Neural Network Background




An input Graph



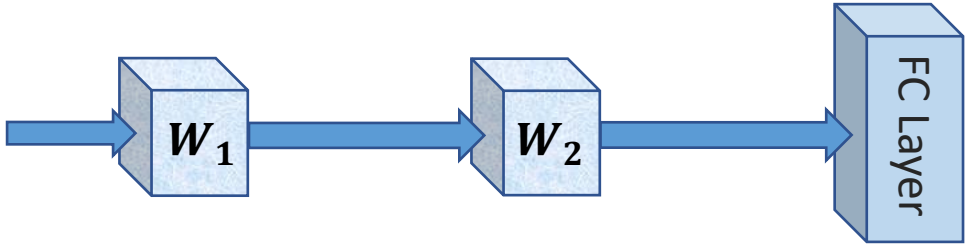

Processing the graph with 2-layer GNN



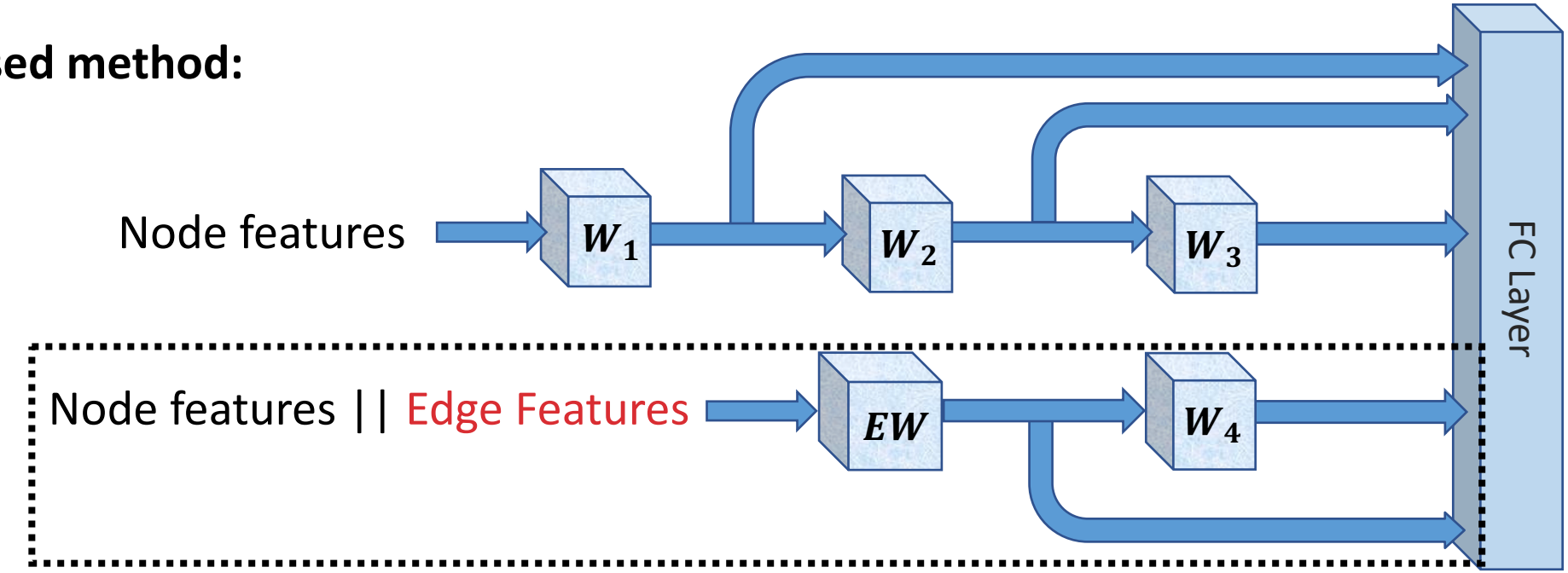
Duke UNIVERSITY

# Our Fast & Standard GNN Method

**Previous GNN example:**

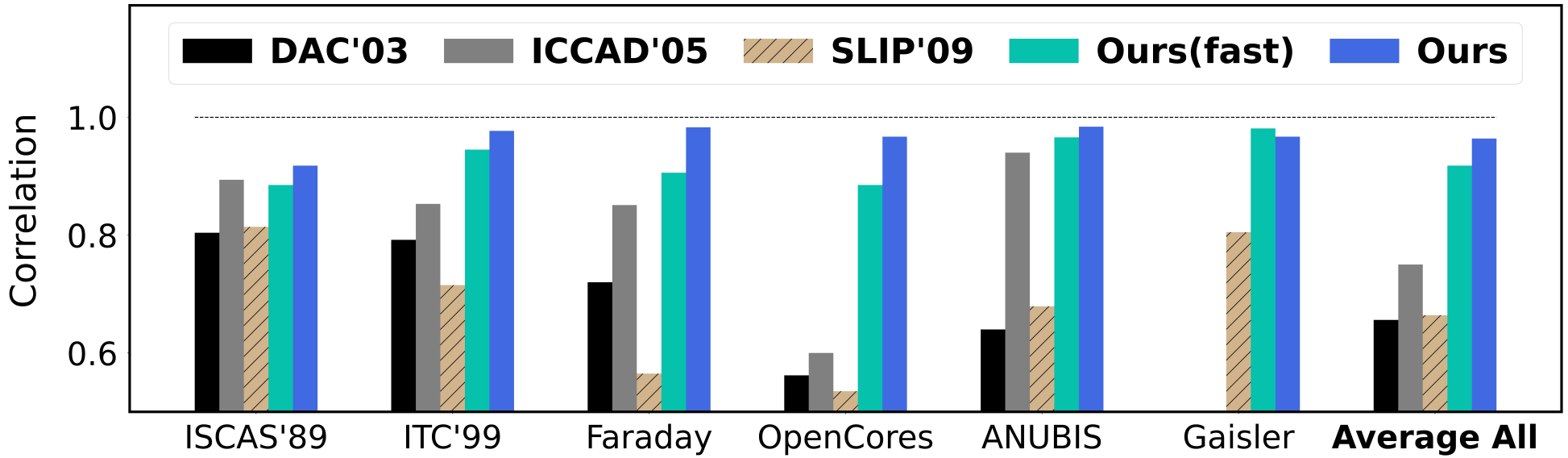



**Our GNN-based method:**




Structures in the standard version

# Experimental Results on Net Length




- Measured on 6 benchmarks with 37 different designs
- **High accuracy (correlation)** for individual net length prediction

Baseline:

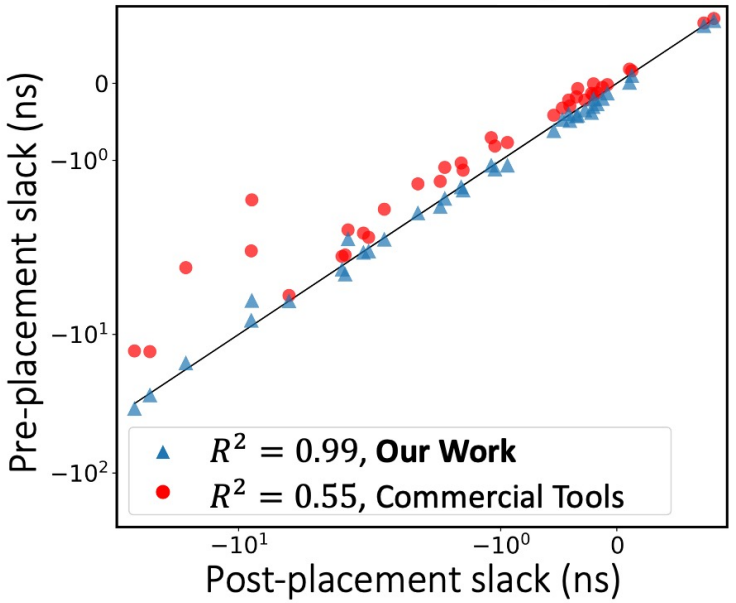
Hu et al. Wire length prediction based clustering and its application in placement, DAC'03.
Kahng et al. Intrinsic shortest path length: a new, accurate a priori wirelength estimator. ICCAD'05.

# Experimental Results on Timing

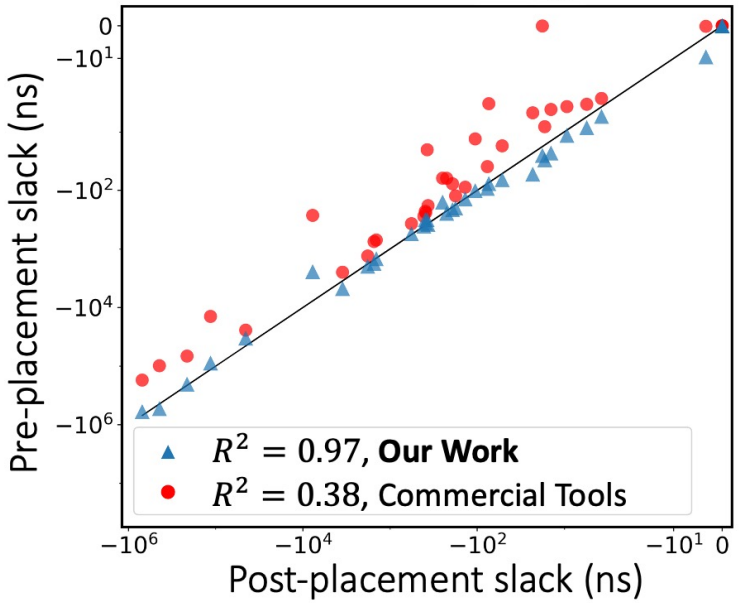
Delay of Timing Arcs

| | $R$ | $R^2$ |
|---|---|---|
| Timing Report | 0.86 | 0.70 |
| Model w/o Net Length | 0.89 | 0.78 |
| **Ours (Fast)** | **0.91** | **0.82** |
| **Ours** | **0.94** | **0.87** |

WNS on All Designs




TNS on All Designs




- **Improve** arc delays over commercial tools
- **Improve** slack estimations (WNS, TNS) over commercial tools

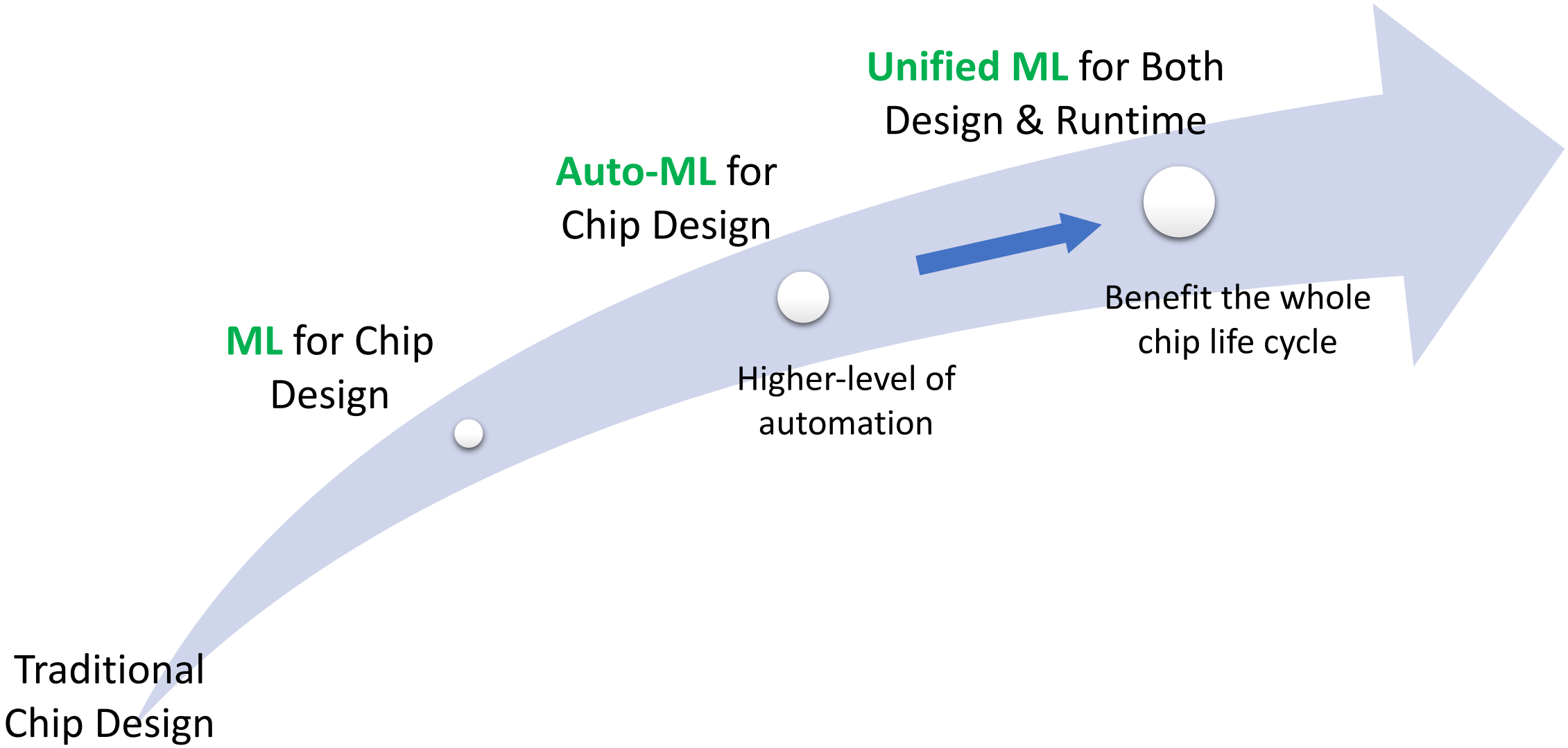


Duke UNIVERSITY

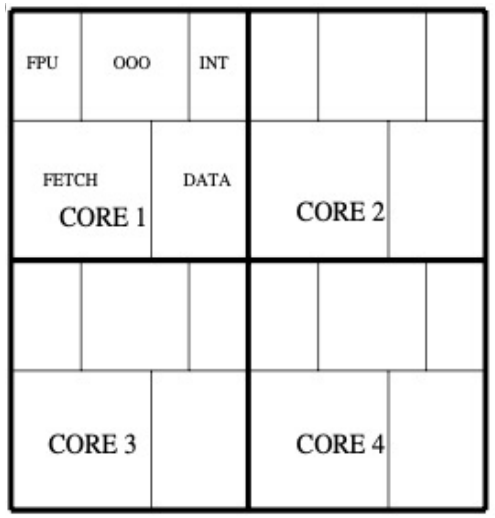
# Case Study 3:

# Power & Power Delivery Challenges

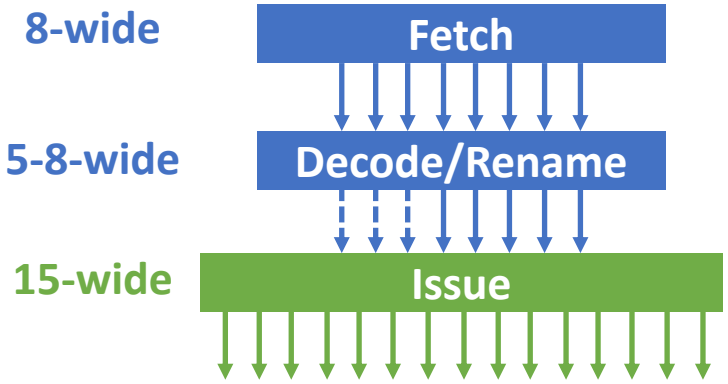
# What I Believe We Should Target



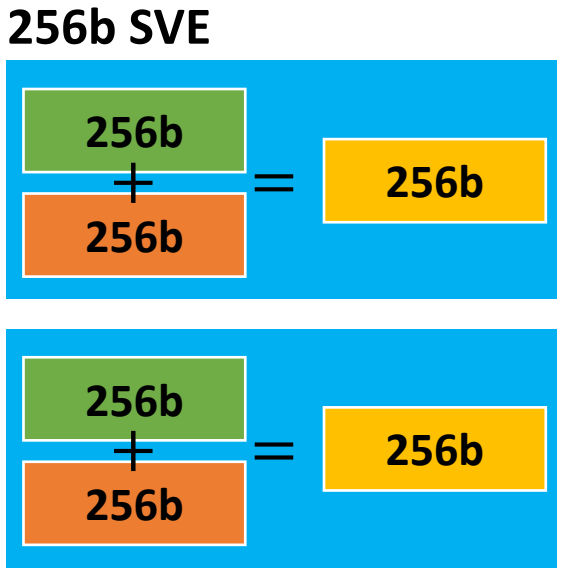
# Challenge 1 – Design-time Power Introspection




Many-core CPU with more transistors




Wider issue




More vectored execution

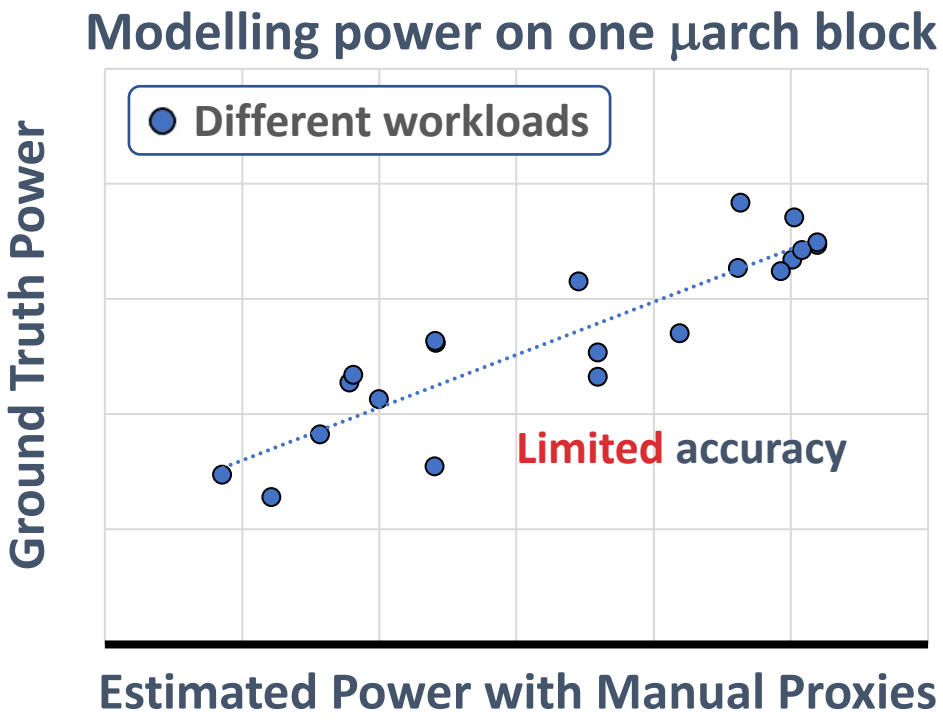
- Delivering generational performance gains **adversely impacts** CPU power
- Power-delivery resources **not keeping pace** with CPU power demands
- **Increasing power-sensitivity drives the need for design-time introspection**

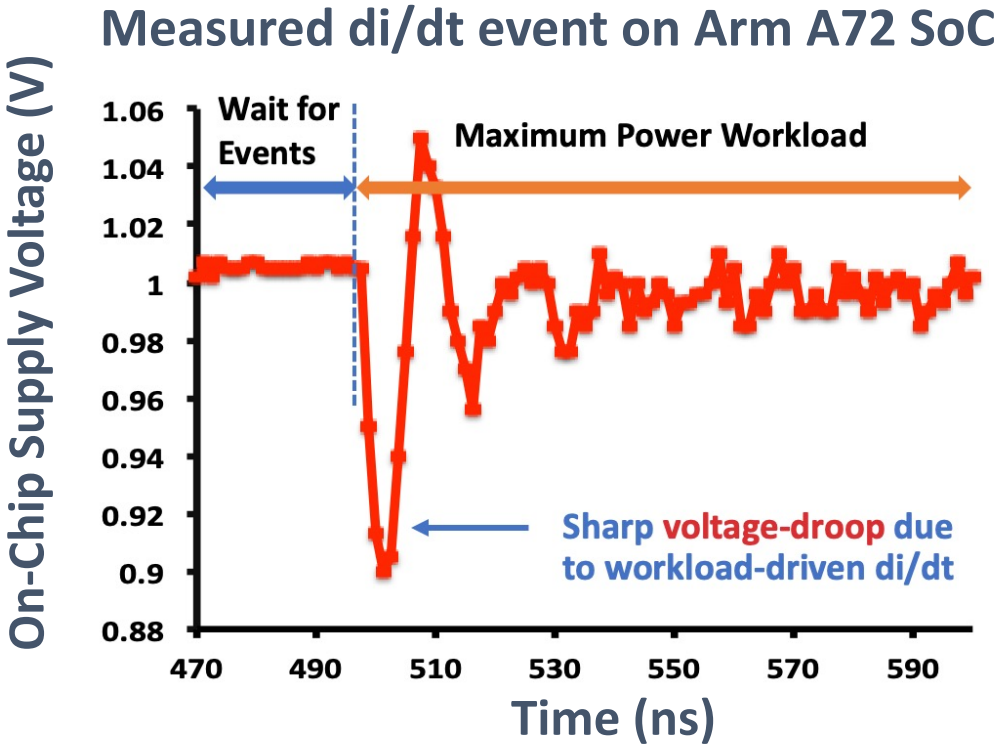


**APOLLO [Z. Xie, et al., MICRO'21] (Best Paper Award)**

Source: Arm Neoverse V1, 2021

# Challenge 2 – Run-time Power Introspection

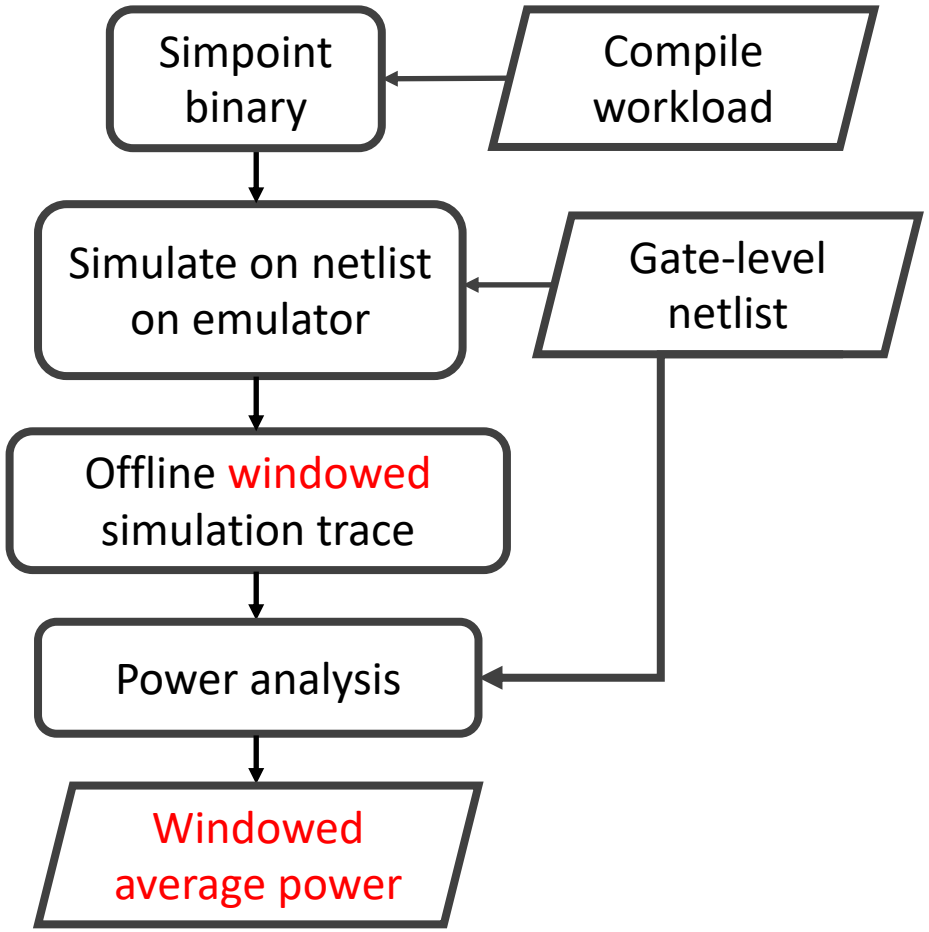



- **Peak-Power mitigation requires accurate power estimation to drive throttling**
  - Manually inferring proxies is very difficult in complex modern CPUs
- **Abrupt changes in CPU current-demand (di/dt event) leading to deep voltage-droop**

# Challenge 3 - Workload Power Characterization

- **Need power-characterization of real-world workloads**
  - Simple micro-benchmarks not longer sufficient
- **Single SPEC simpoint can take weeks on the expensive emulator**
  - Power measurement is expensive
- **Only average power consumption available**
  - Impossible to scale to di/dt event analysis




**Industry-Standard Emulator-Driven Power Flow**

# Challenges from Both Design-time and Runtime

A unified solution for both scenarios

## Runtime Challenges Summary

- Peak power mitigation
  - **Difficult to manually** infer proxies
- Voltage droop (Ldi/dt) mitigation
  - Require very **low** response latency

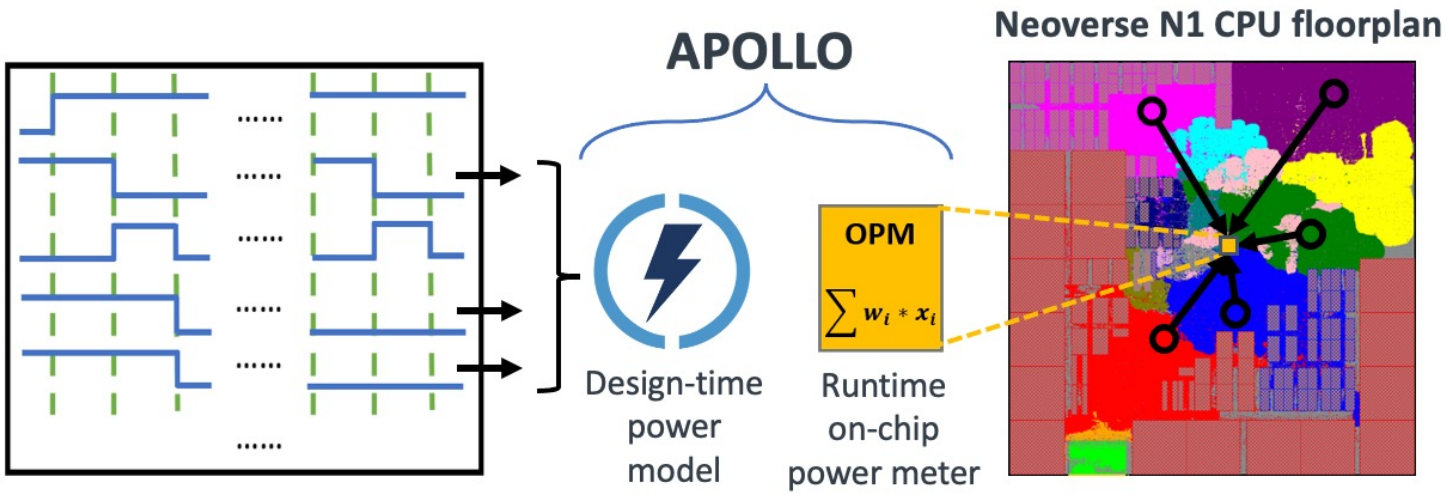
## Design-time Challenges Summary

- Simulation on realistic workloads
  - **Expensive** and **slow**
  - **Limited** temporal-resolution

## What is An "Ideal" Power Estimator?

1. **Accurate yet fast**
2. **Achieve high temporal resolution**
3. **Low runtime on-chip overheads**
4. **Easily extensible to diverse designs**

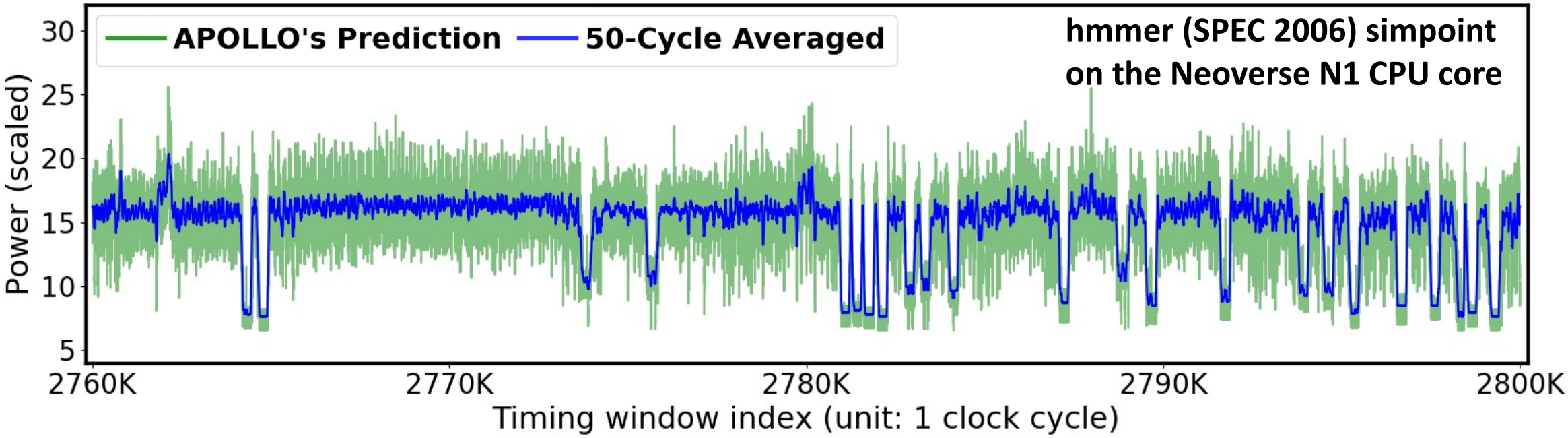
# APOLLO: A Unified Power Modeling Framework




- **Fast**, yet **accurate** design-time simulation
- **Single-cycle** temporal resolution
- **Low-cost**, yet **accurate** runtime monitoring
- Design-agnostic **automated** development

# A Workload Execution Preview of APOLLO

**40K cycles of APOLLO Power Estimation out of a trace of 17M cycles**

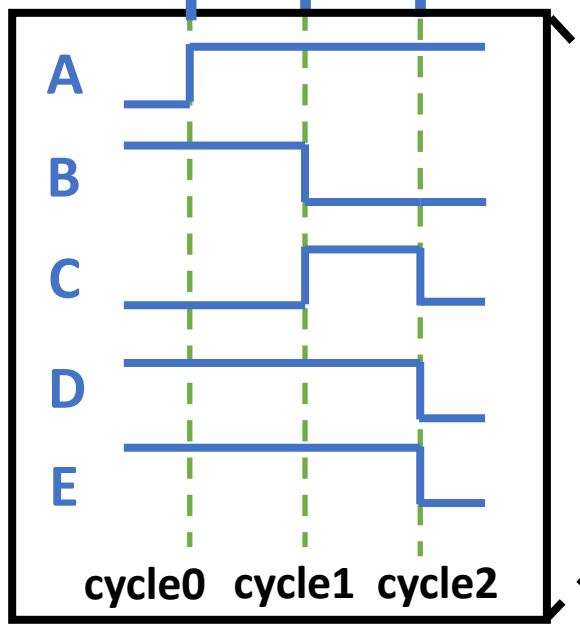



- **~2 weeks execution time** reduced to **few minutes** on the emulator
- Unprecedented power-introspection due to **single-cycle** temporal resolution



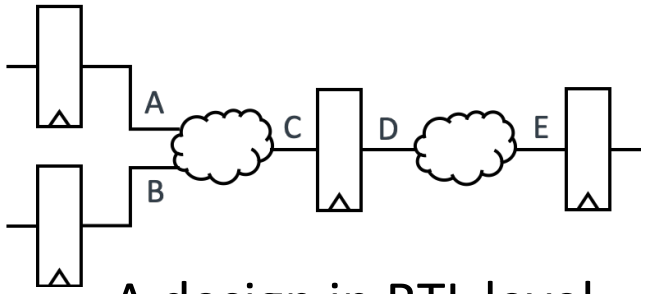
Duke UNIVERSITY

# APOLLO Feature Generation & Model Training




In .fsdb/.vcd file format

| | A | B | C | D | E | Power |
|---|---|---|---|---|---|---|
| cycle0 | 1 | 0 | 0 | 0 | 0 | P0 |
| cycle1 | 0 | 1 | 1 | 0 | 0 | P1 |
| cycle2 | 0 | 0 | 1 | 1 | 1 | P2 |




A design in RTL level.

$X$ $y$

$F$

| | A | B | C | D | E | |
|---|---|---|---|---|---|---|
| cycle0 | 1 | 0 | 0 | 0 | 0 | |
| cycle1 | 0 | 1 | 1 | 0 | 0 | … … |
| cycle2 | 0 | 0 | 1 | 1 | 1 | |
| ⋮ | | | | | | |

$M$ RTL signals

$=$

P0
P1
P2
⋮

Train the ML model: $F(X) = y$

**$M$ > 500,000** in Neoverse N1

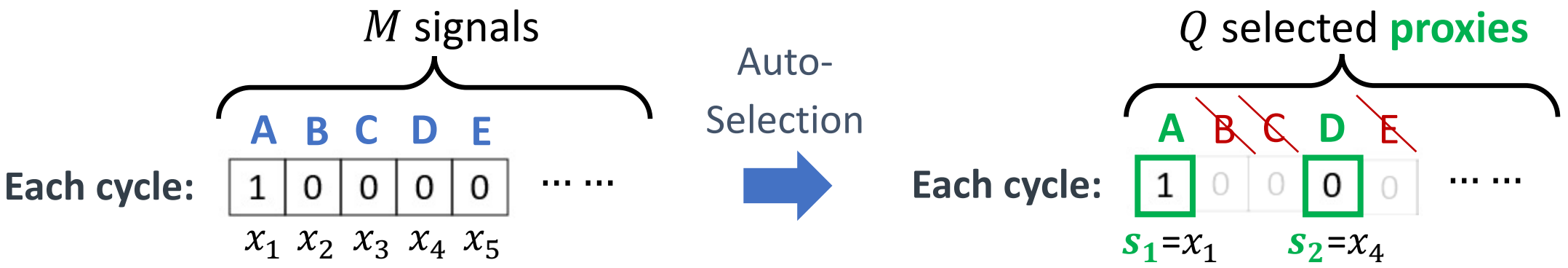
**$M$ > 1,000,000** in Cortex-A77



Duke UNIVERSITY

# Simple Key Ideas

- **Linear** model can estimate power accurately
- **Small** portion of signals (proxies) can provide enough information




Linear model with $M$ RTL signals

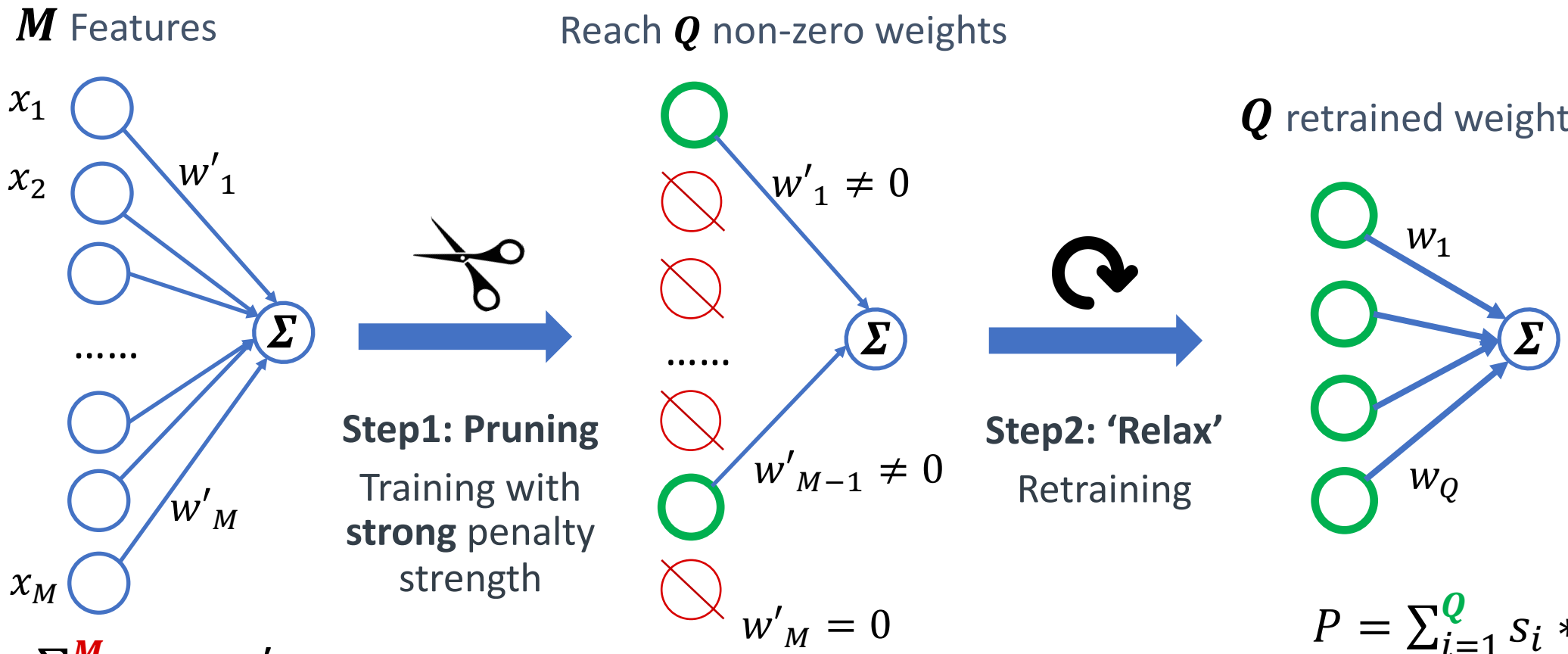
$$P = \sum_{i=1}^{M} x_i * w'_i$$

Linear model with $Q$ selected proxies

$$P = \sum_{i=1}^{Q} s_i * w_i$$

# ML-Based Power Proxies Selection

Model construction in two steps



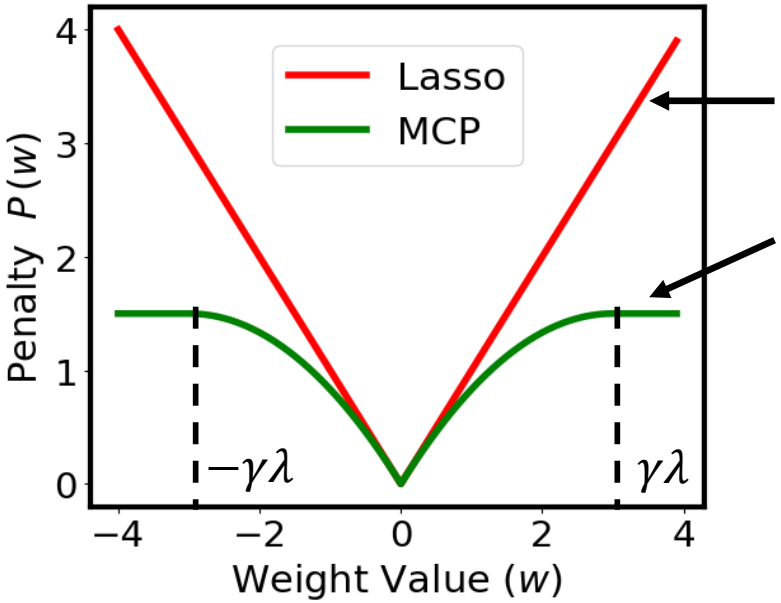

Minimax concave penalty (**MCP**) for pruning



Duke UNIVERSITY

# Why MCP for Pruning?

- To make $\boldsymbol{Q} \ll \boldsymbol{M}$, penalty is set to be very large.
- Lasso **degrades** model accuracy under large penalty
- MCP **protects** large weights thus **maintains** model accuracy




**Lasso penalize all weights**

**MCP protects large weights**

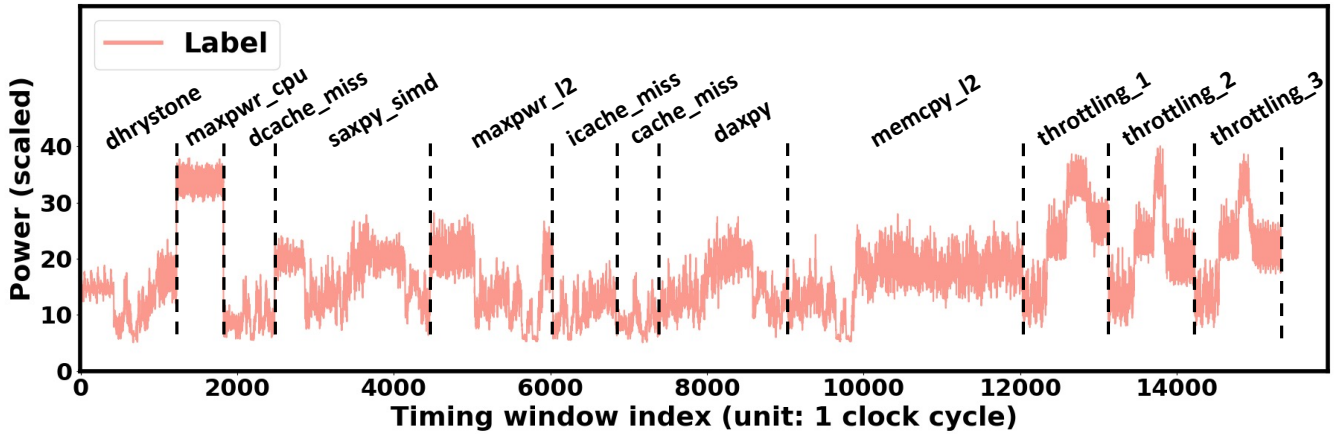


Duke UNIVERSITY

# Model Training and Testing


**Neoverse N1** (infra)
Deployed in AWS Graviton


**Cortex A77** (mobile)
Deployed In Snapdragon 865



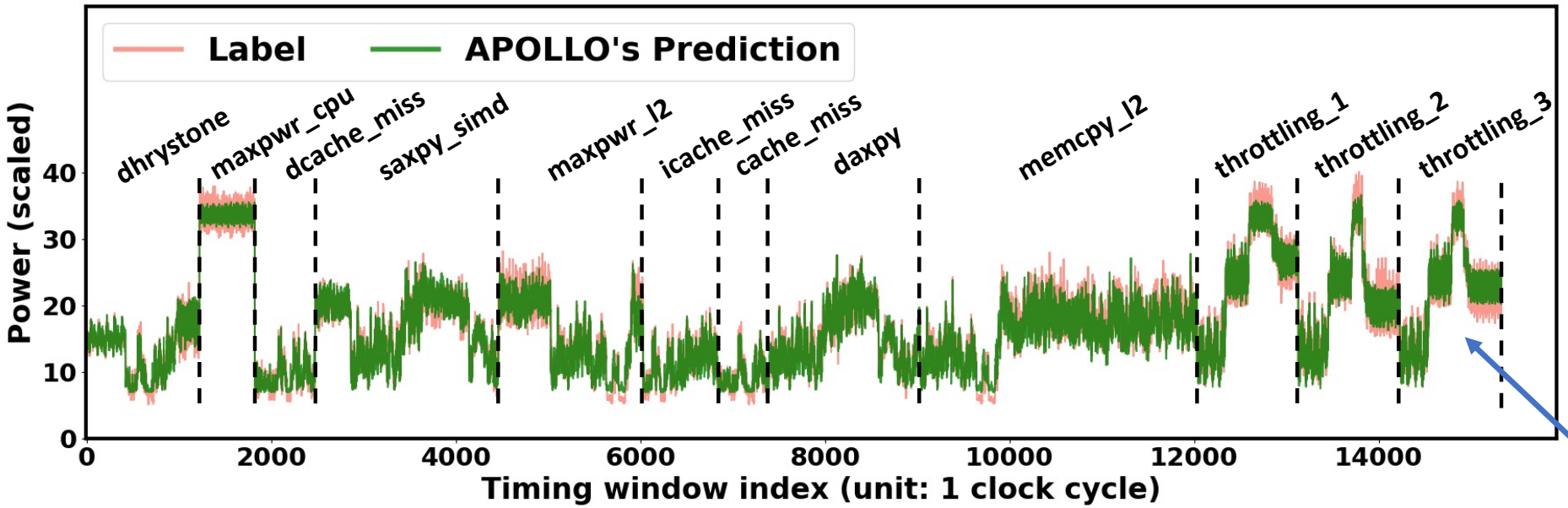

- Experiments on 3GHz 7nm Arm **commercial** microprocessors **Neoverse N1** and **Cortex A77**
- **Automatically** generate a “diverse” set of random micro-benchmarks for training
- Testing on **various** Arm power-indicative workloads



Duke UNIVERSITY

# Prediction Accuracy as Design-Time Power Model

Per-cycle prediction from APOLLO with $\boldsymbol{Q}$=159 proxies




- MAE = 7.19%
- $R^2$ = 0.953

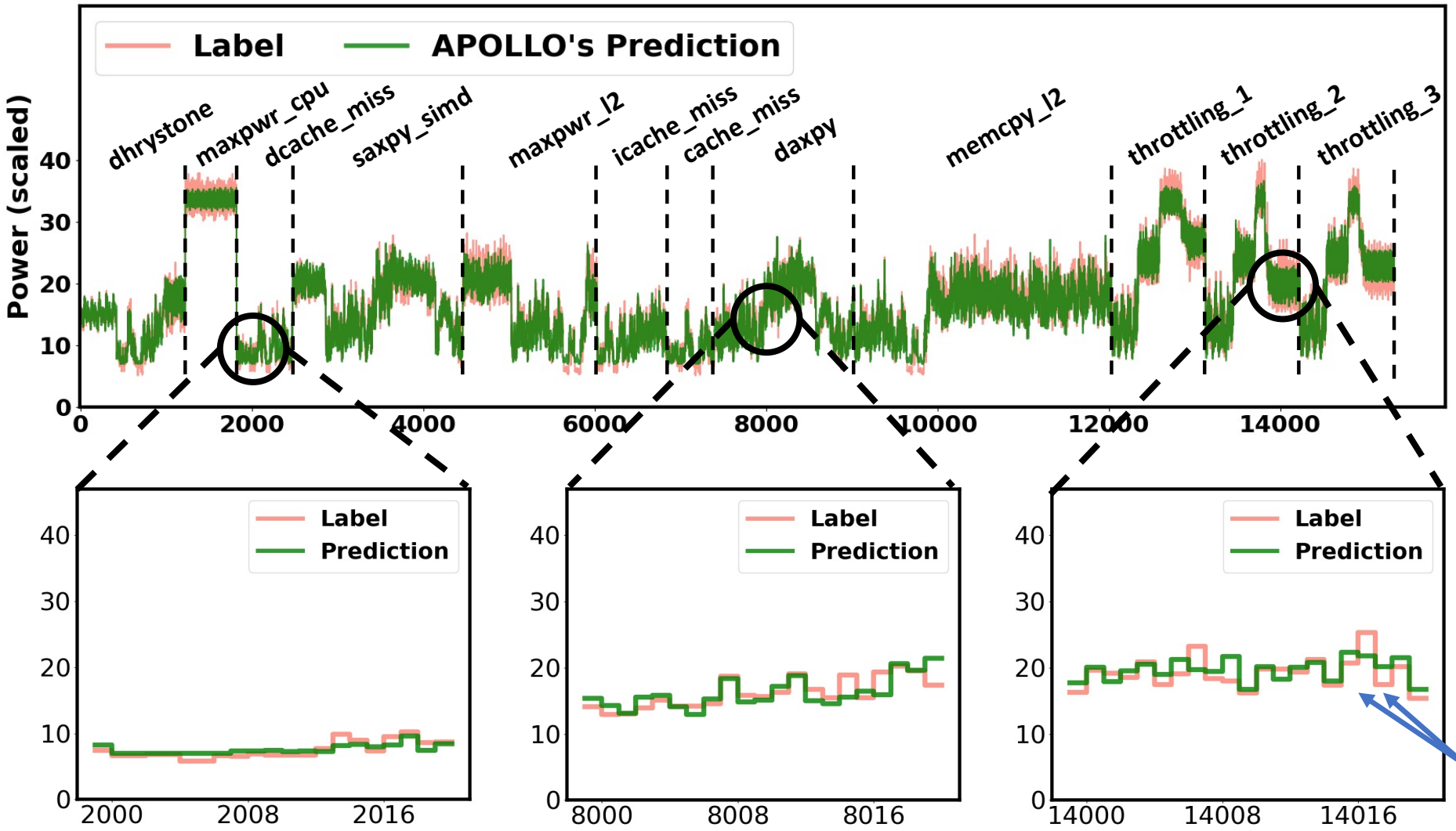
Prediction trace shows great agreement with ground-truth



Duke UNIVERSITY

# Prediction Accuracy as Design-Time Power Model

Per-cycle prediction from APOLLO with $Q$=159 proxies



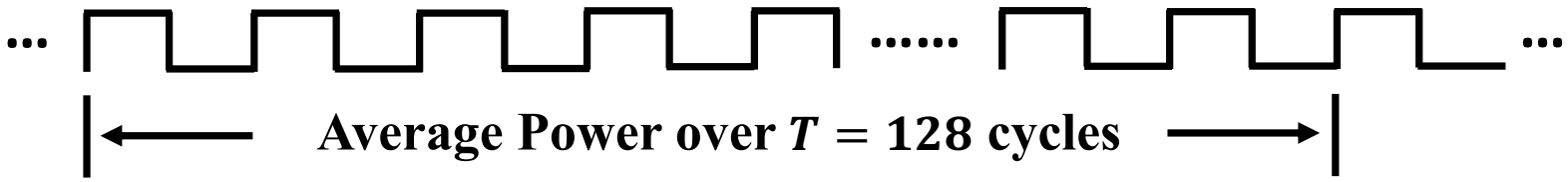

- MAE = 7.19%
- $R^2$ = 0.953

Per-cycle error can be averaged

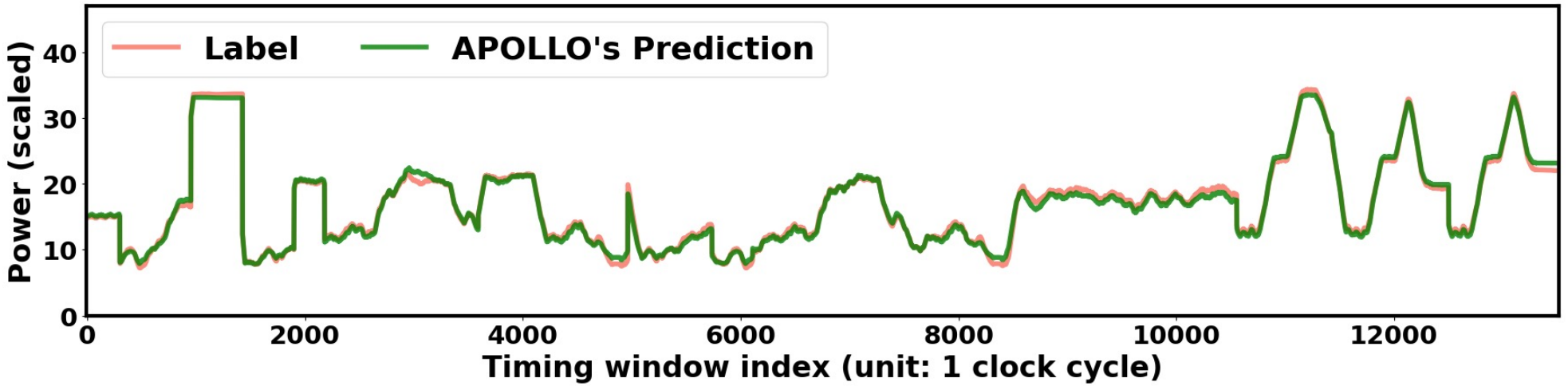


Duke UNIVERSITY

# Accuracy on Multi-Cycle Power Estimation




**128-cycle** prediction from APOLLO with $Q$=70 proxies



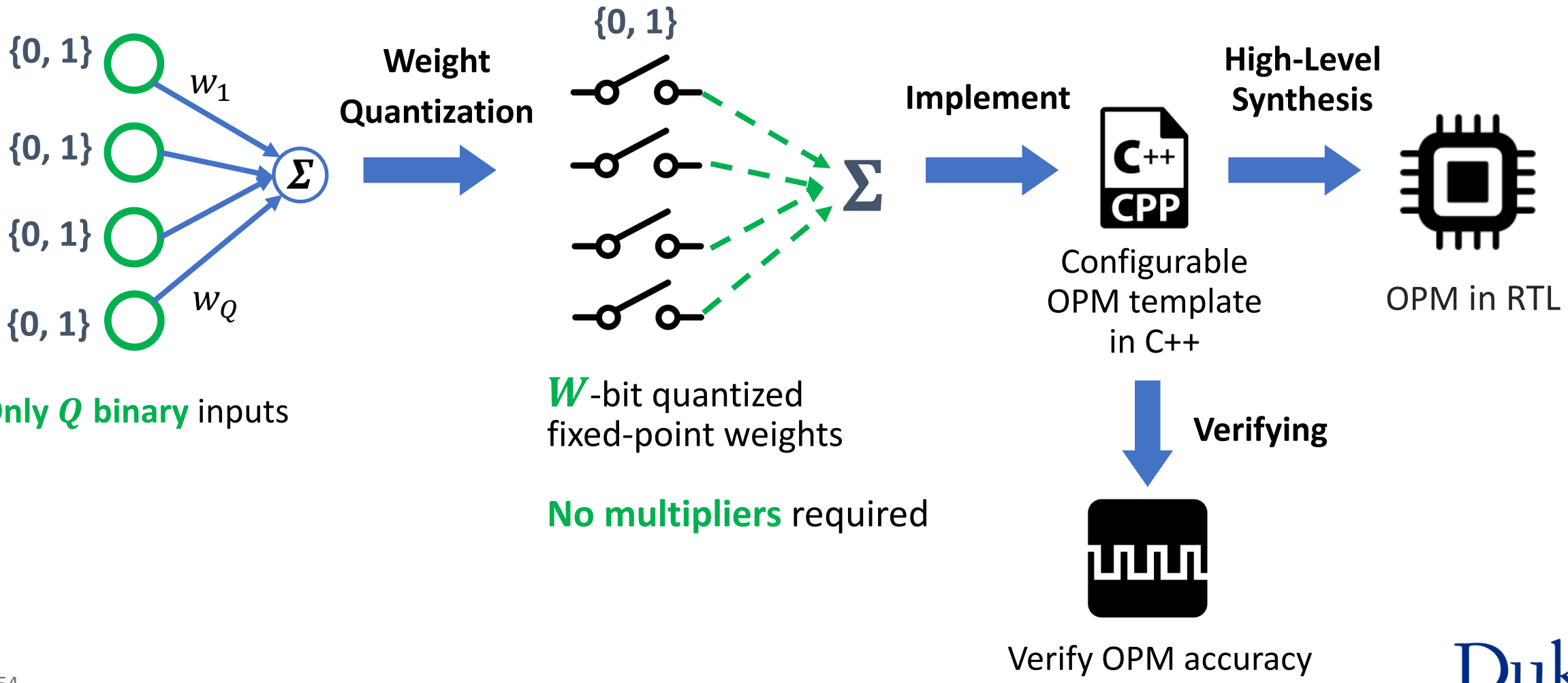

- MAE = 2.82%
- $R^2$ = 0.993
- Higher accuracy



Duke UNIVERSITY

# Automated Low-Cost Runtime OPM Implementation

APOLLO is designed to be hardware-friendly

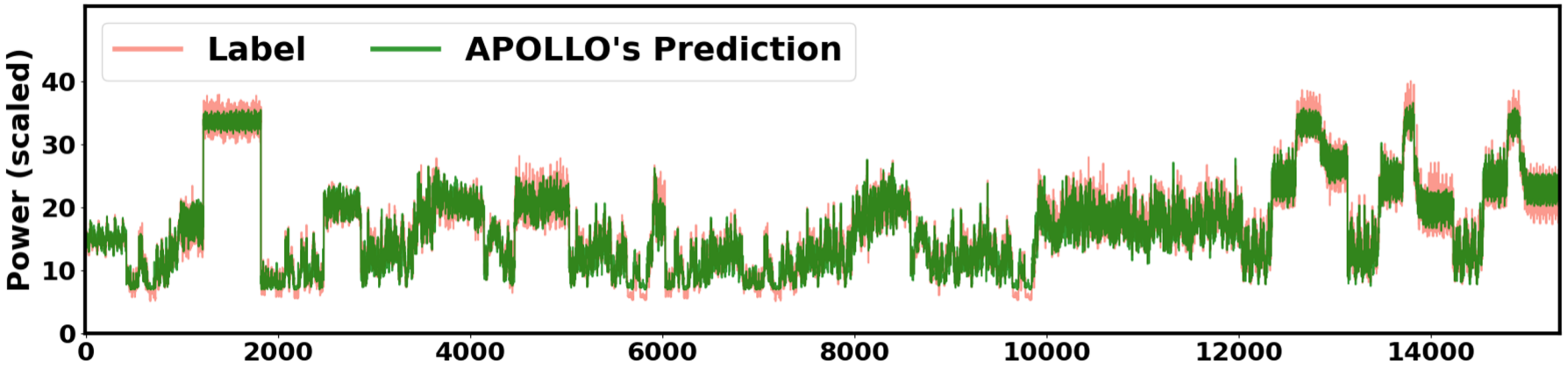





Duke UNIVERSITY

# Prediction Accuracy from Design-time Model & OPM

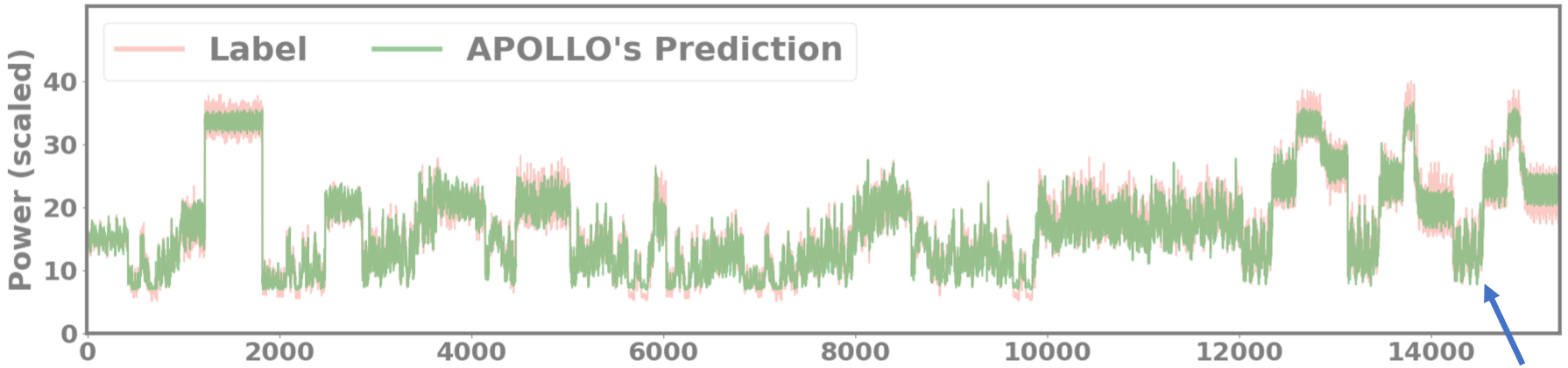
Per-cycle prediction from APOLLO with $\boldsymbol{Q}$=159 proxies




- MAE = 7.19%
- $R^2$ = 0.953

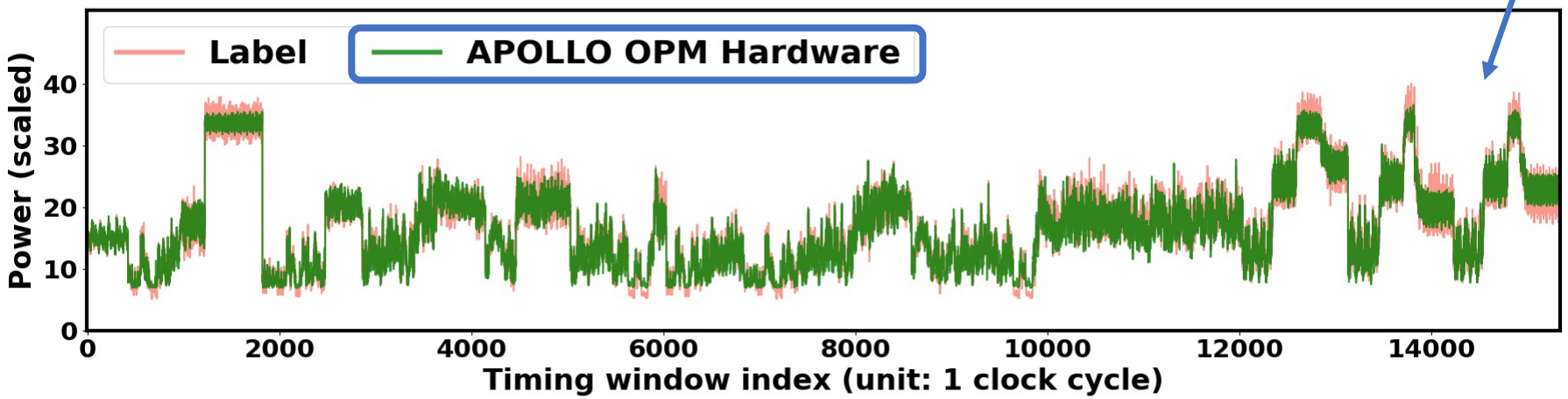
# Prediction Accuracy from Design-time Model & OPM

Per-cycle prediction from APOLLO with $\boldsymbol{Q}$=159 proxies




Negligible difference

Prediction from runtime OPM with $\boldsymbol{Q}$=159 proxies



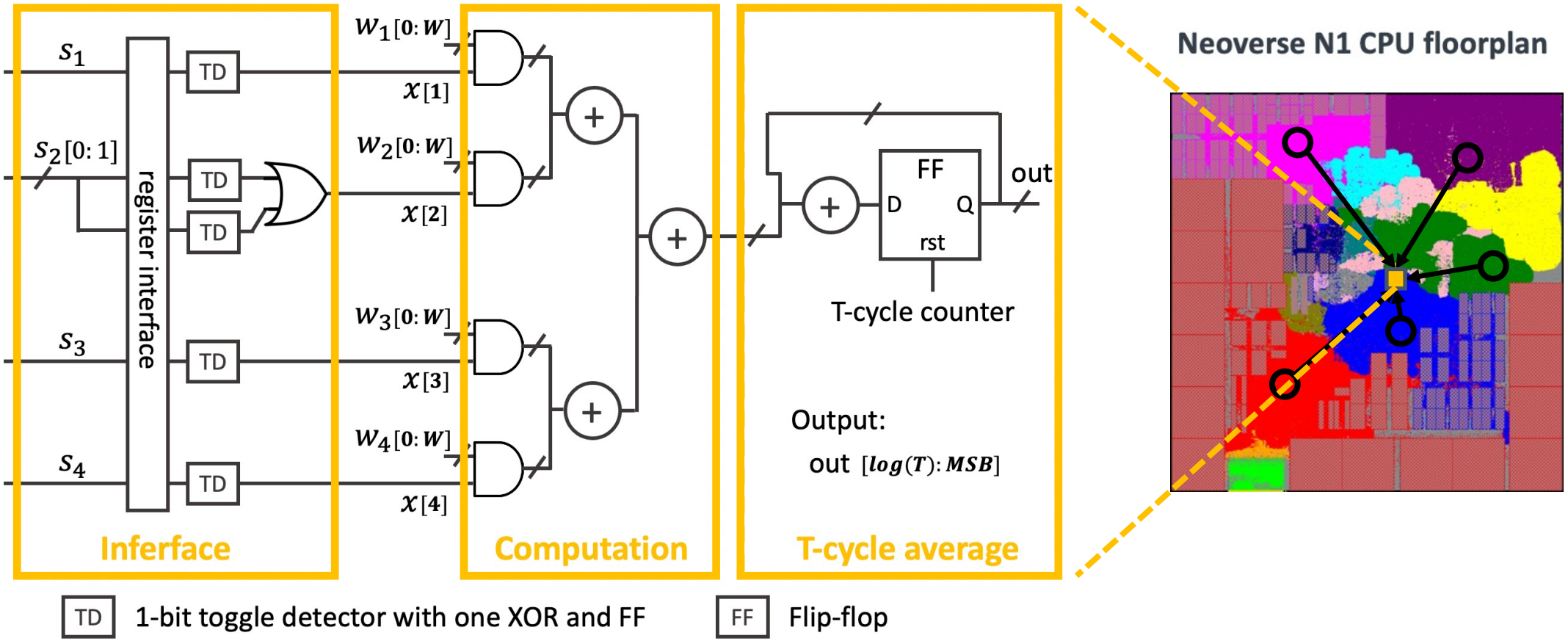

- MAE = 7.19%
- $R^2$ = 0.953

Negligible difference

- MAE = 7.19%
- $R^2$ = 0.953

- **$W$=11 bits after quantization**

# Overview of the OPM Hardware Design



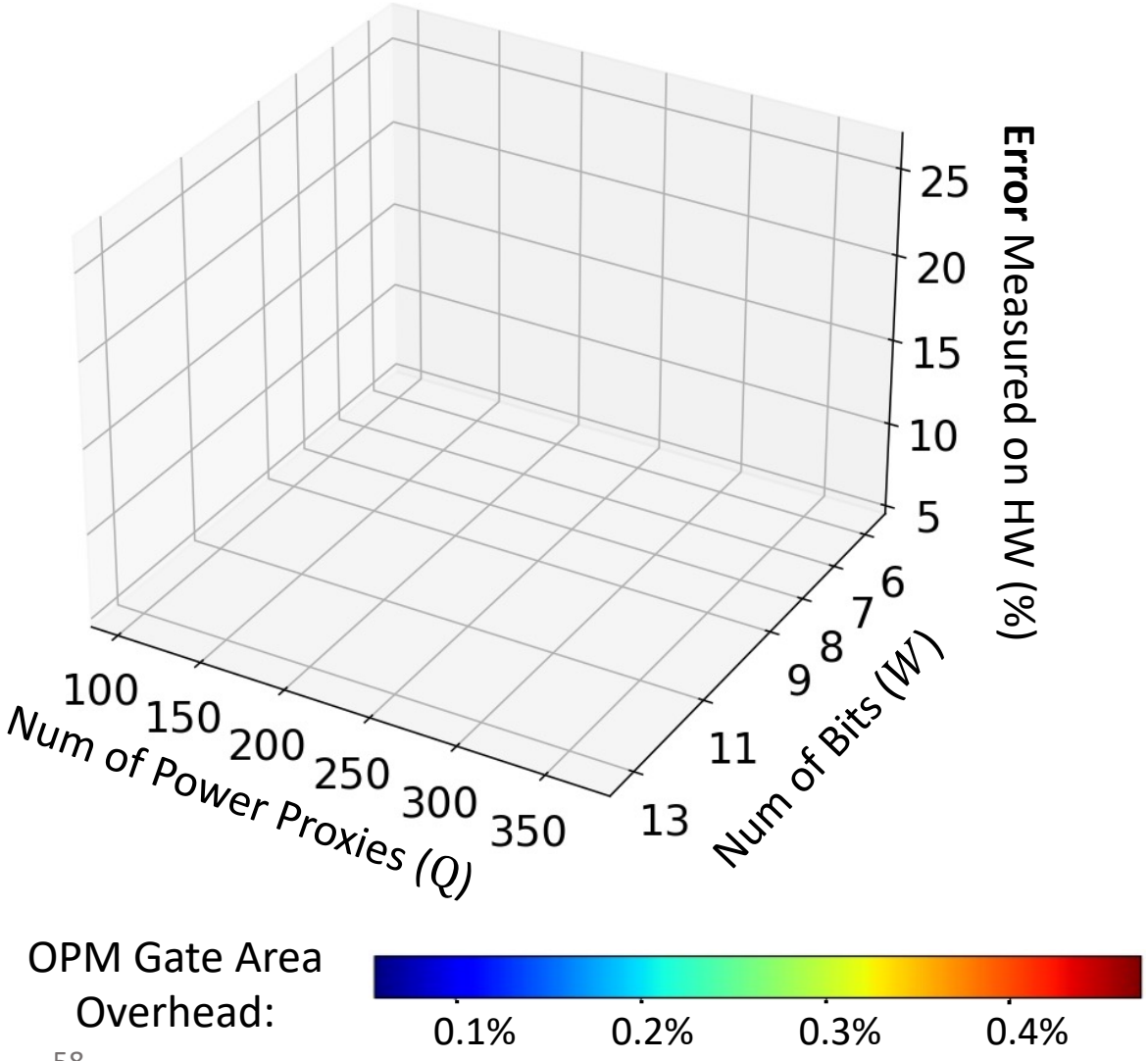

- **No** multipliers or dividers, only **Q** binary inputs and **W**-bit quantized weights



Duke UNIVERSITY

# Accuracy vs. Hardware Cost (Area Overhead) of the OPM

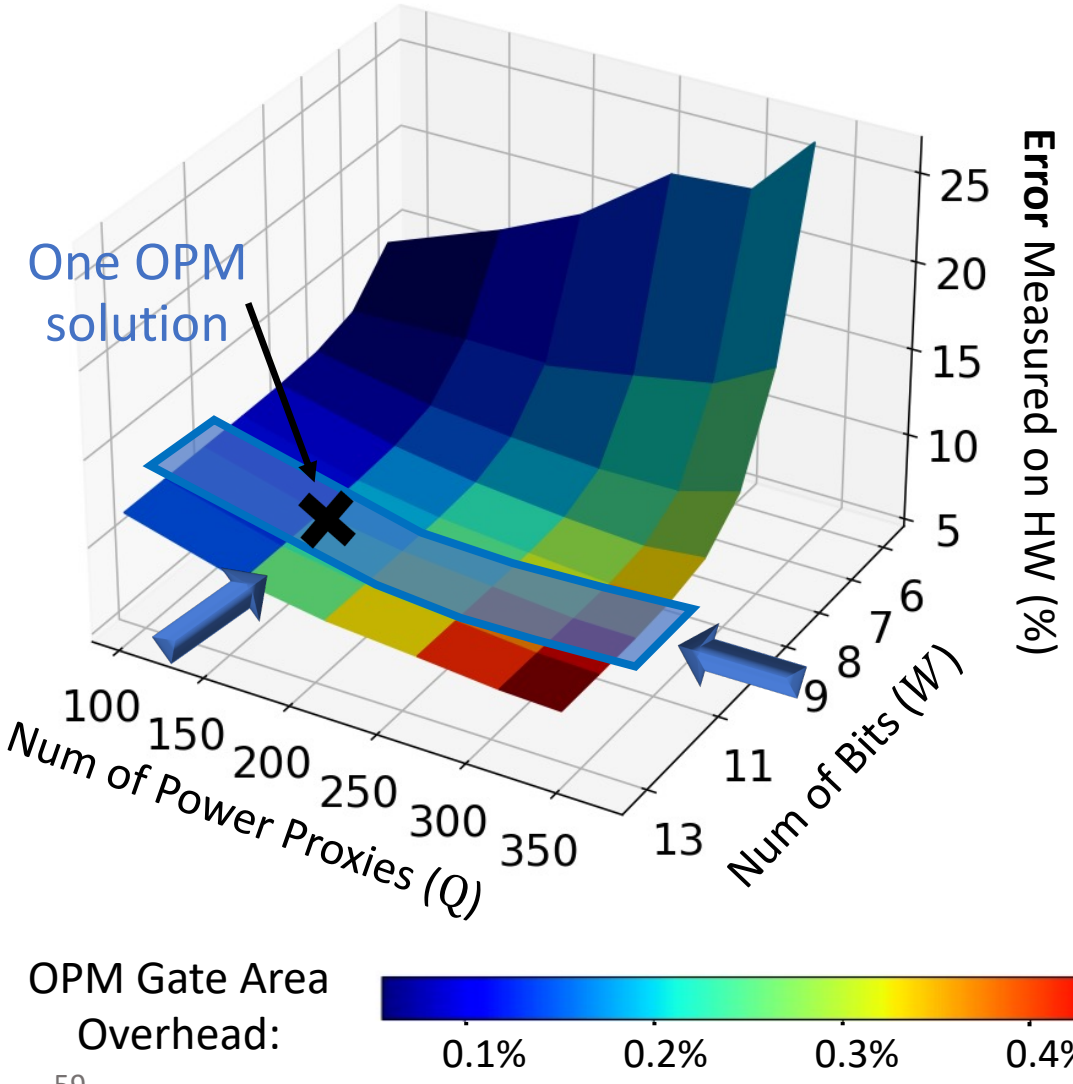
Runtime OPM implementation on Neoverse N1




- Trade-off accuracy and hardware cost
- Sweep proxy num $\boldsymbol{Q}$ and quantization bits $\boldsymbol{W}$

# Accuracy vs. Hardware Cost (Area Overhead) of the OPM

Runtime OPM implementation on Neoverse N1

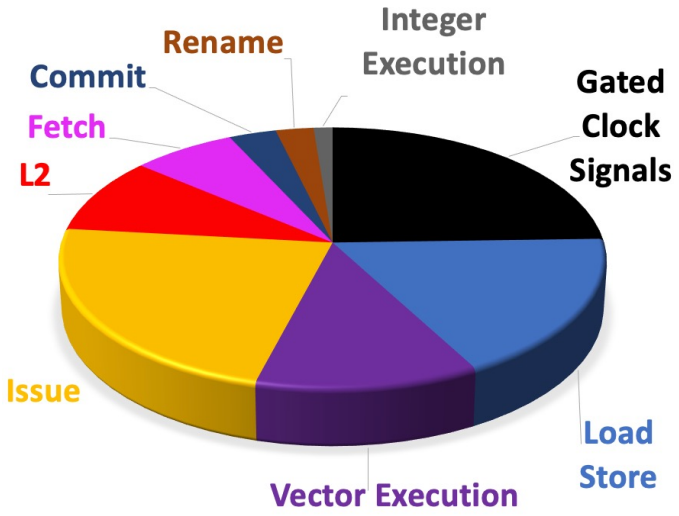



- Trade-off accuracy and hardware cost
- Sweep proxy num $\boldsymbol{Q}$ and quantization bits $\boldsymbol{W}$
- **Strategy**
  - Keep quantization $\boldsymbol{W}$= 10 to 12 bits
  - Vary $\boldsymbol{Q}$ for different solutions
- **For an OPM with $\boldsymbol{Q}$=159, $\boldsymbol{W}$=11**
  - **< 0.2%** area overhead of Neoverse N1
  - **< 10%** in the error



Duke UNIVERSITY

# Potential Application: Design-time Power Introspection



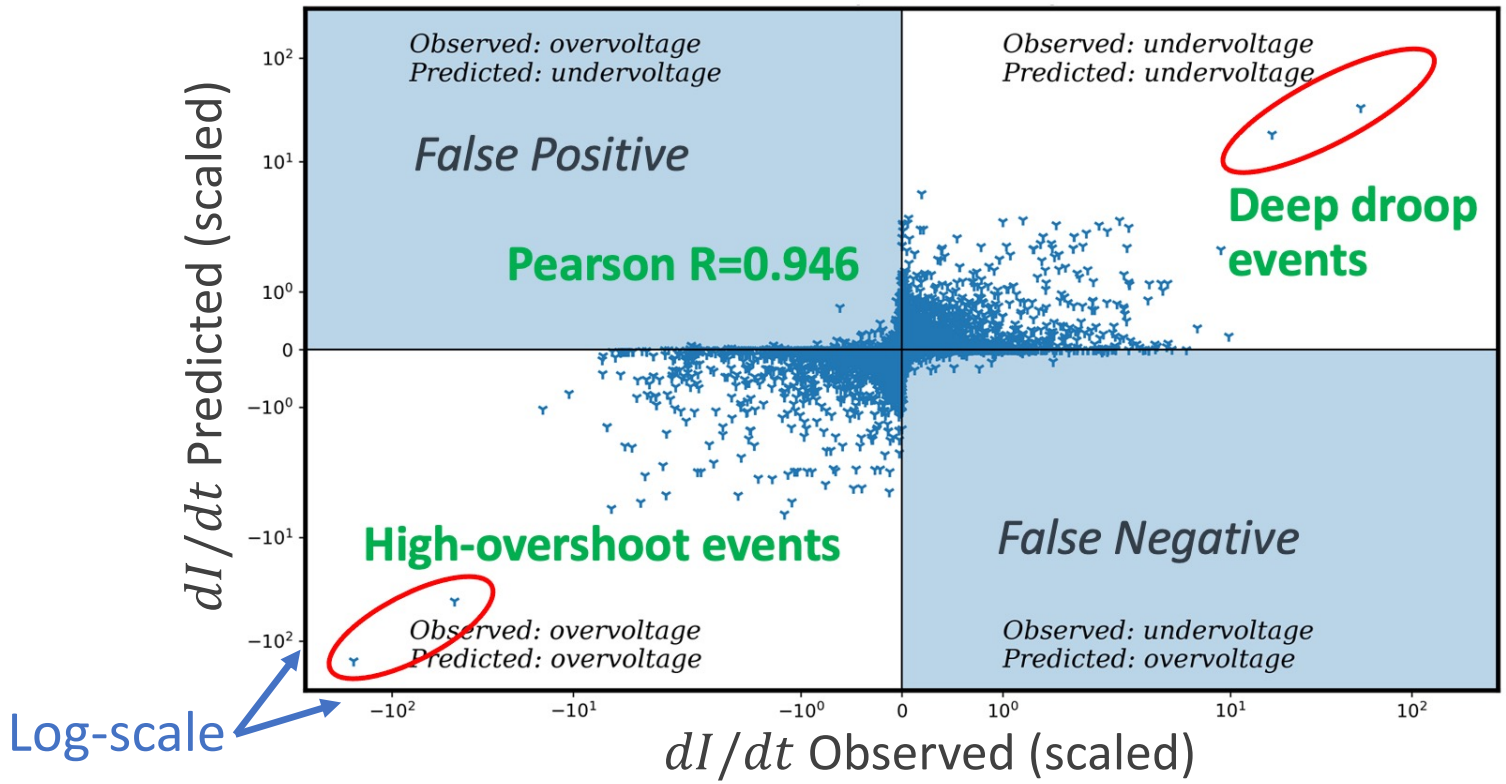

Distribution of power proxies on Neoverse N1


Trained **only** with **more meaningful signals** as initial feature candidates

Better **interpretability**

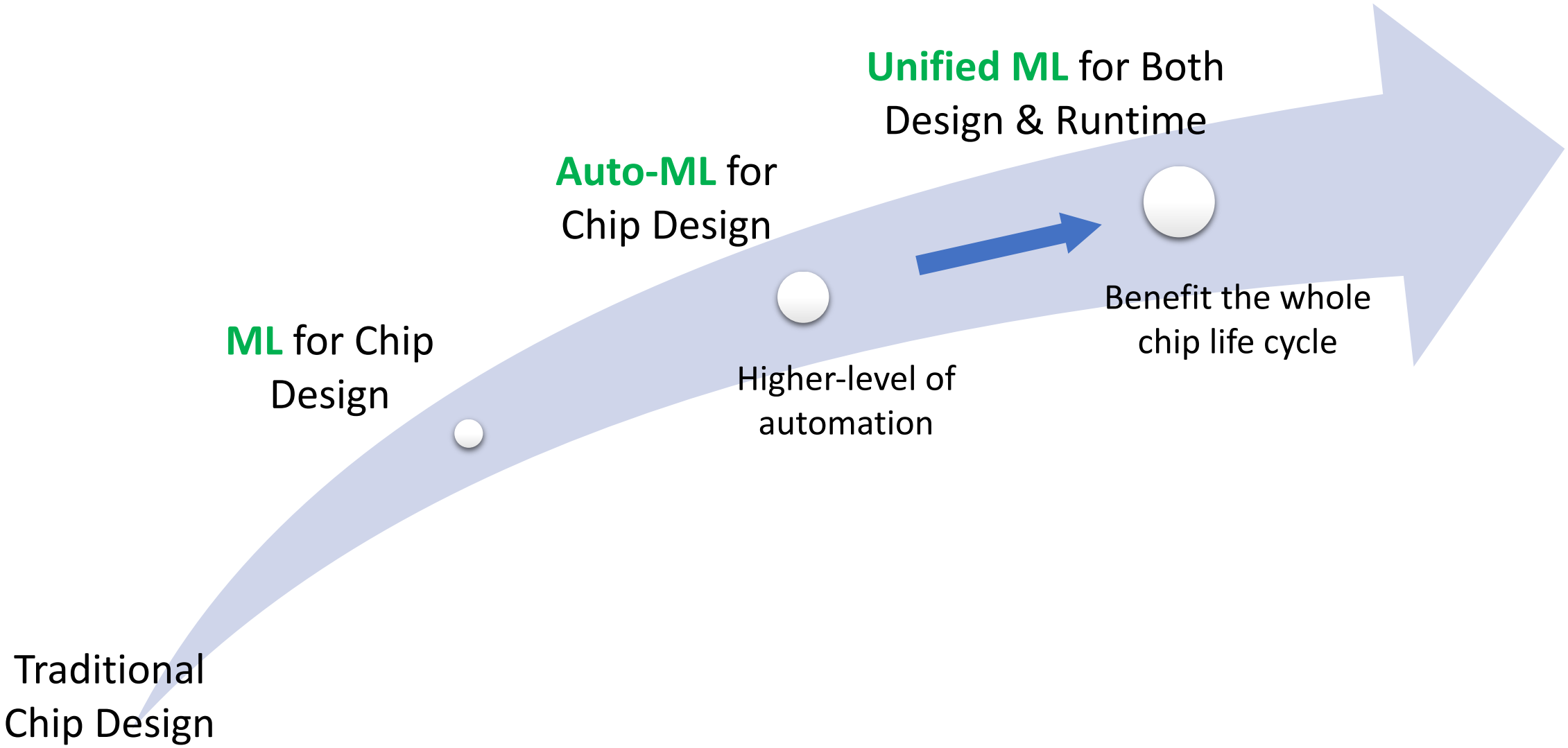
## Identify power contributors for designers!

# Potential Application: Runtime dI/dt Mitigation



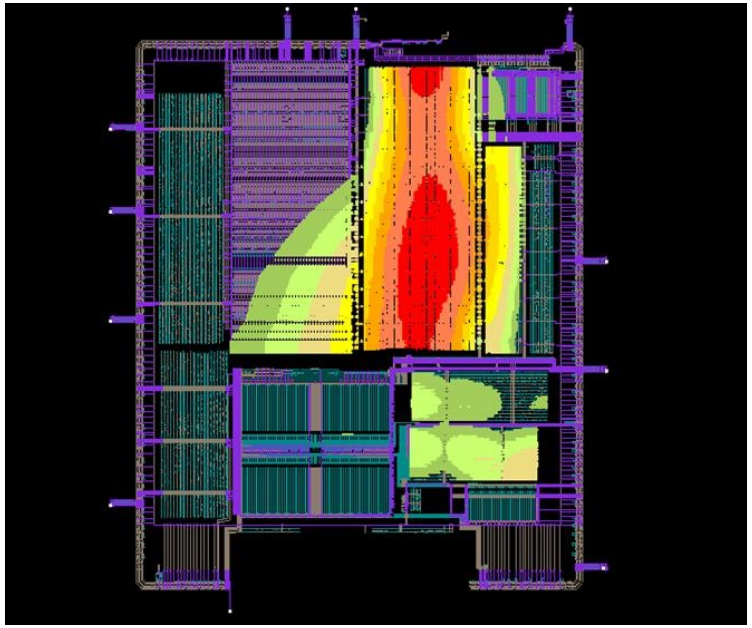

## Enable CPU-driven Proactive $dI/dt$ Mitigation!

# What I Believe We Should Target

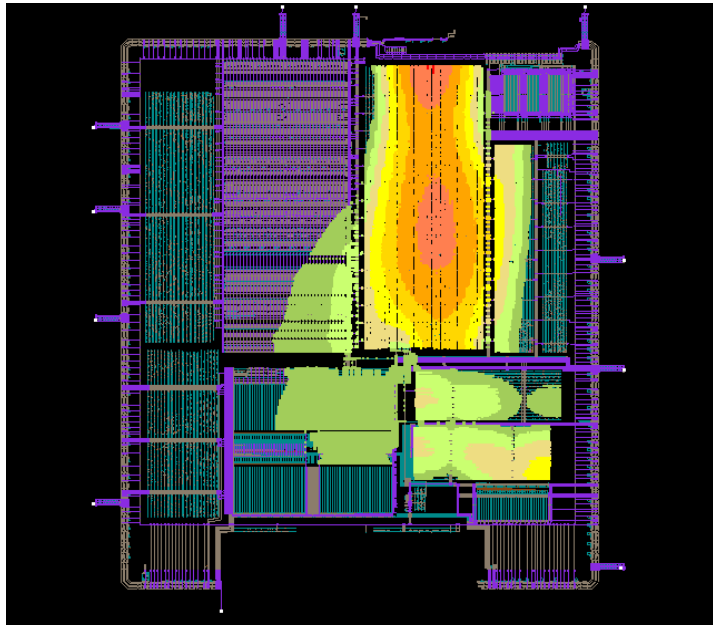


# How does the voltage noise distribute on the circuit layout?

# Fine-grained IR Drop Hotspots on Layout



Initial IR drop distribution



After adding wires to PDN

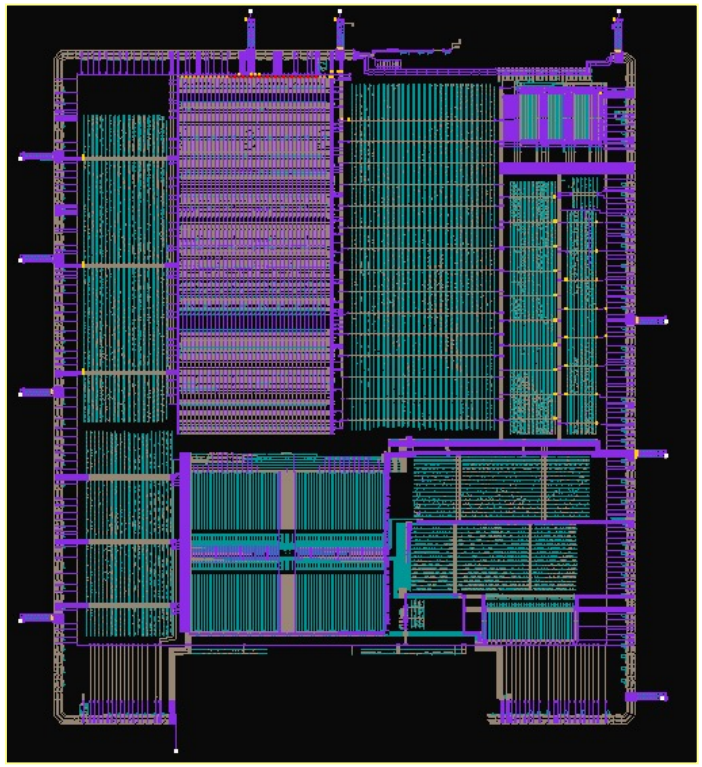
- IR drop unevenly distribute on a layout
- Need identification of IR drop hotspot before mitigation



Source: Cadence

# Cost on IR Drop Identification

Time-consuming simulations performed repeatedly



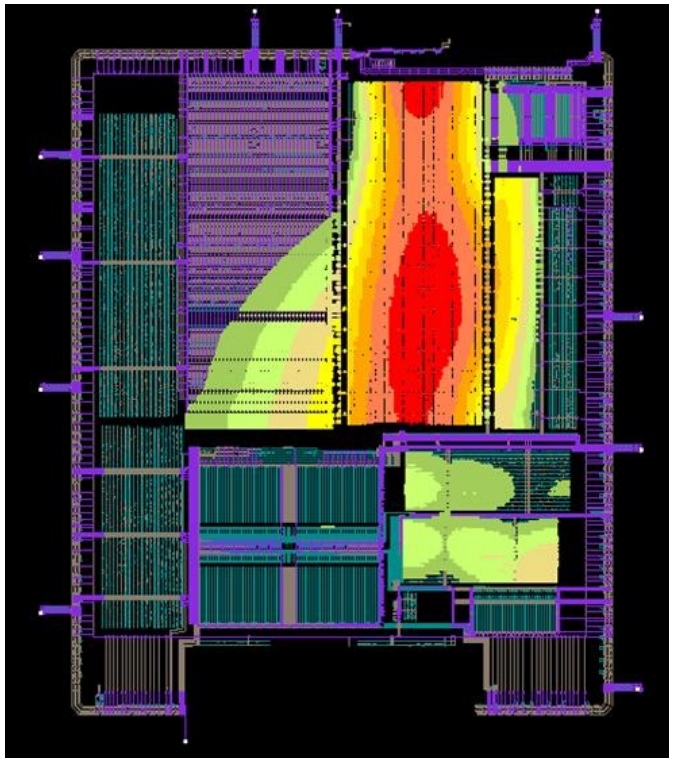
Provide feedback
Edit the design

Simulation to get IR drop hotspots

Ansys ~4 hours on 1M cells

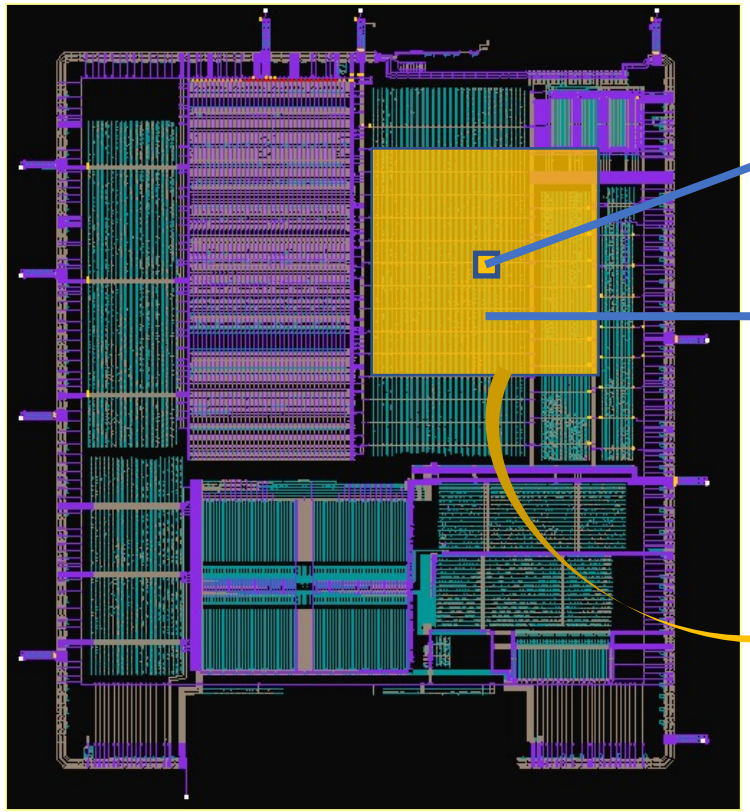
ML-based algorithm:
Not only once!
5 mins, >90% accuracy

~~4-hour simulation again~~



[Z. Xie, et al. ASPDAC'20]
[Z. Xie, et al. ICCAD'20] (Invited)

# Dynamic IR Drop Estimation Methodology



The location to check IR drop

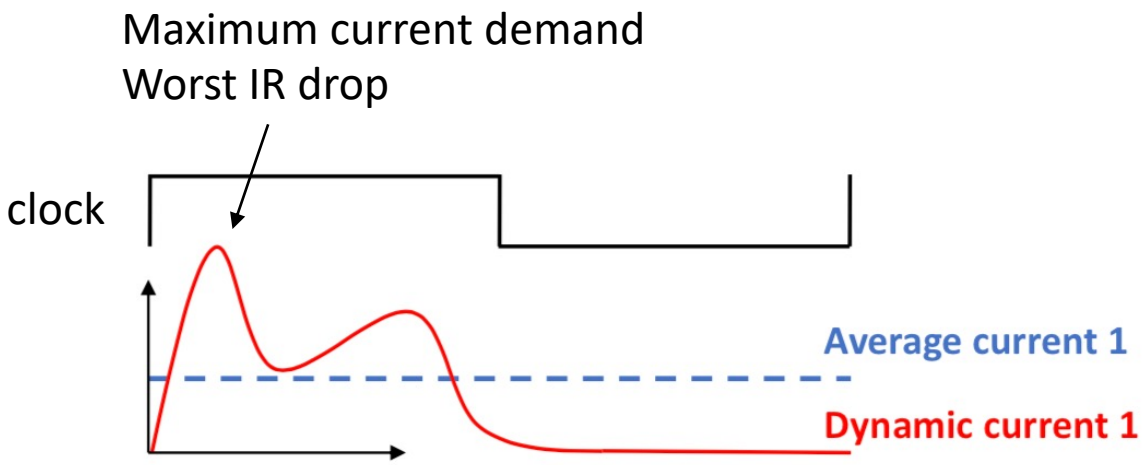
rger region

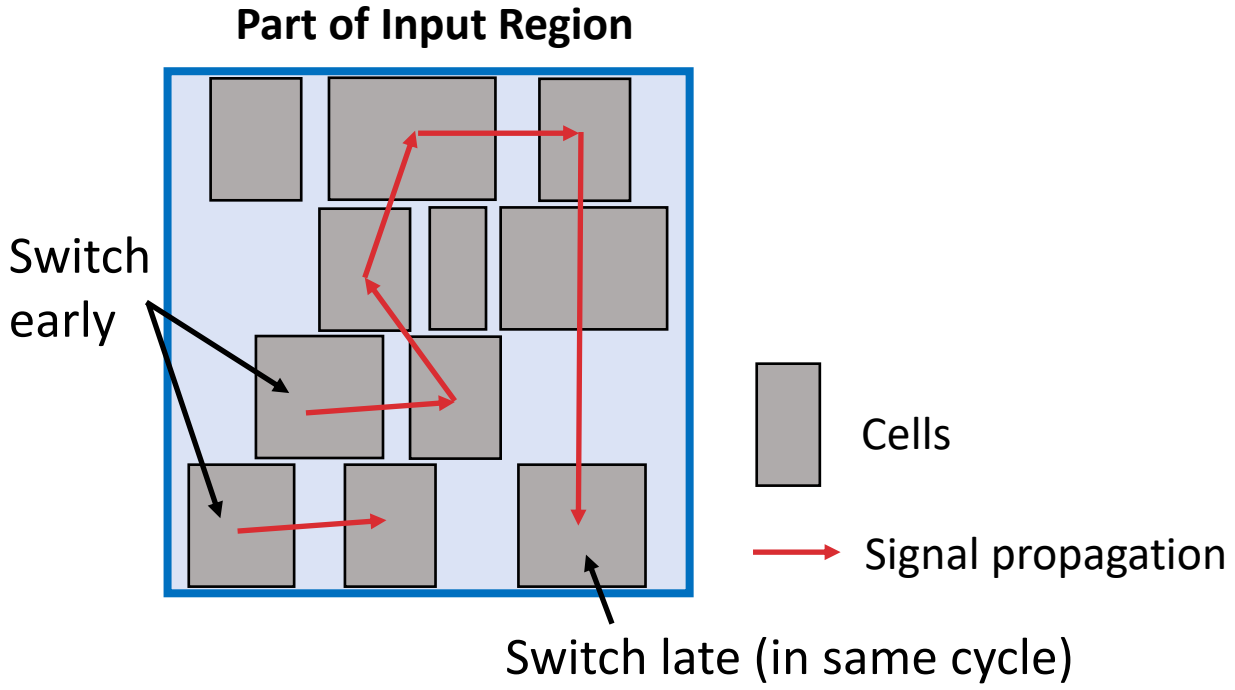
**One important factor is missing!**

IR drop value

- Neighbors all contribute to power demand at a local region

# Incorporate Timing with Customization

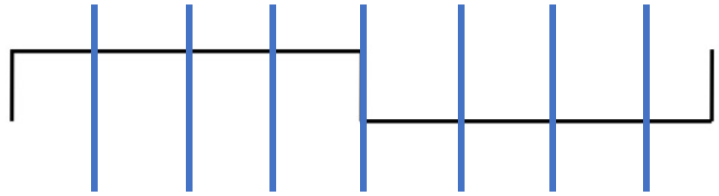


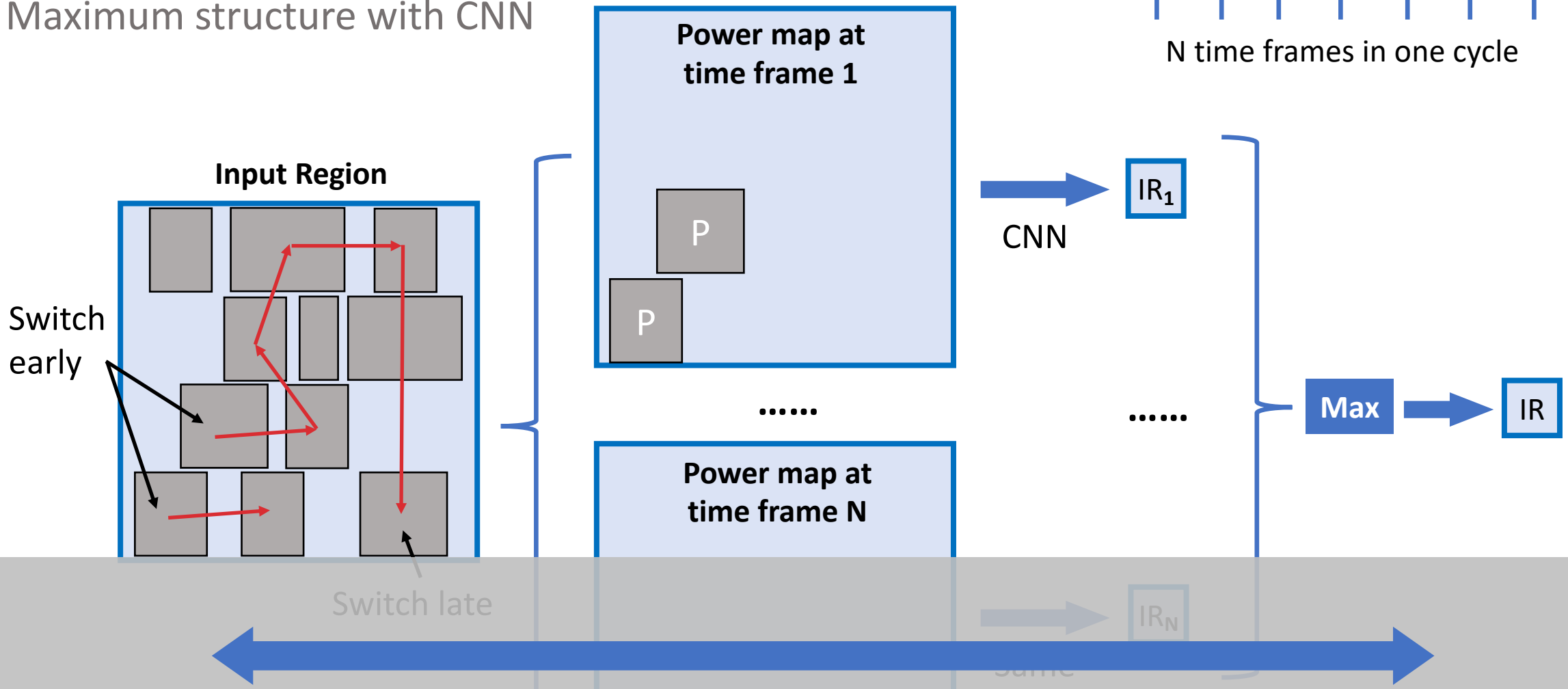

- Power/current consumption depends on switching activities
- Gates switch at different time frame/instant!

# Our Customized Method

Maximum structure with CNN

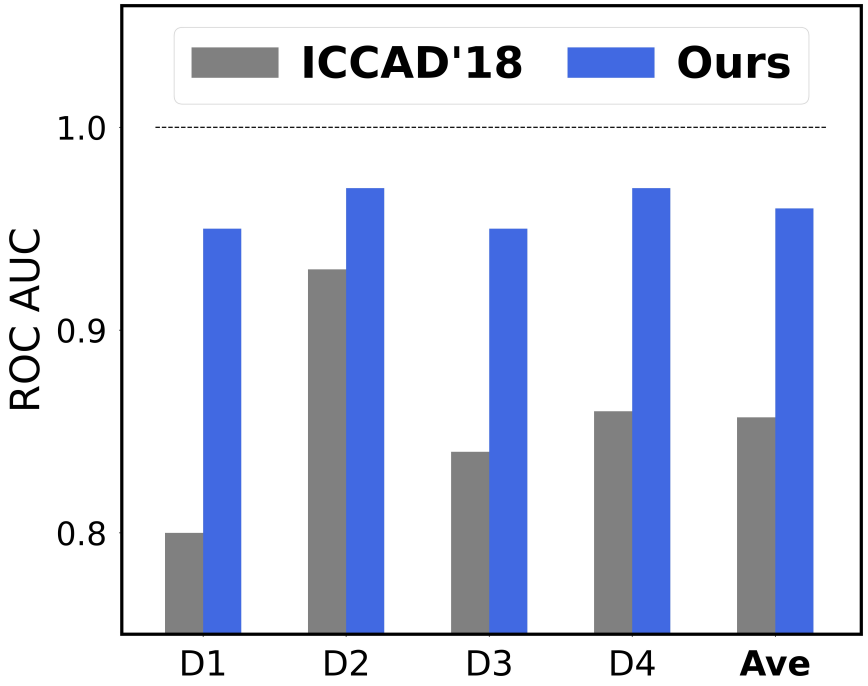


N time frames in one cycle




**Performs end-to-end training**

# Experimental Results on IR Drop



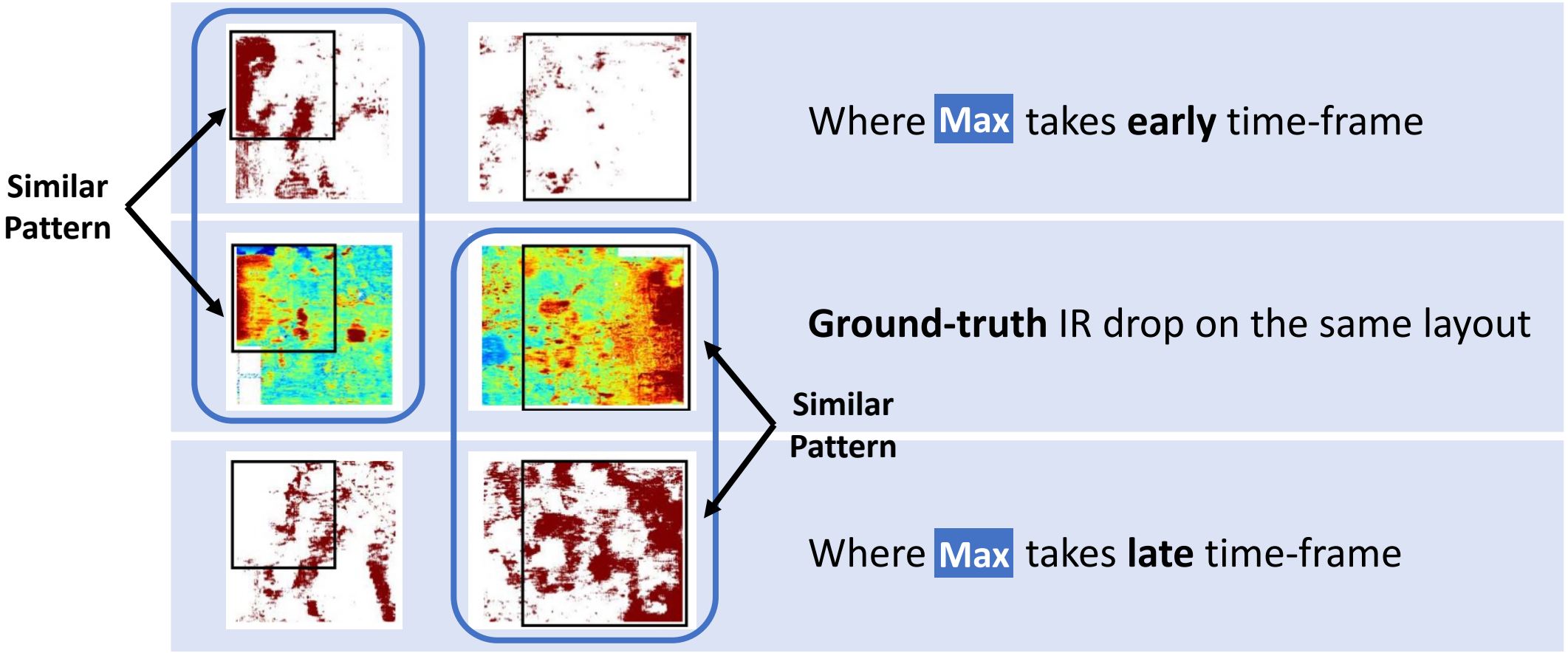

| Design MD1 | Violated Cell | # Hotspots |
|---|---|---|
| Before Mitigate | 22185 | 5092 |
| After Mitigate | 17052 | 3778 |
| Improvement | 23% | 26% |

| Design MD2 | Violated Cell | # Hotspots |
|---|---|---|
| Before Mitigate | 31097 | 3627 |
| After Mitigate | 23941 | 2489 |
| Improvement | 23% | 31% |

- Superior accuracy over previous work
- Integrated to guide mitigation flow to reduce IR violations by **20-30%**

Baseline:
Fang et al. Machine-learning-based dynamic IR drop prediction for eco. ICCAD'18.

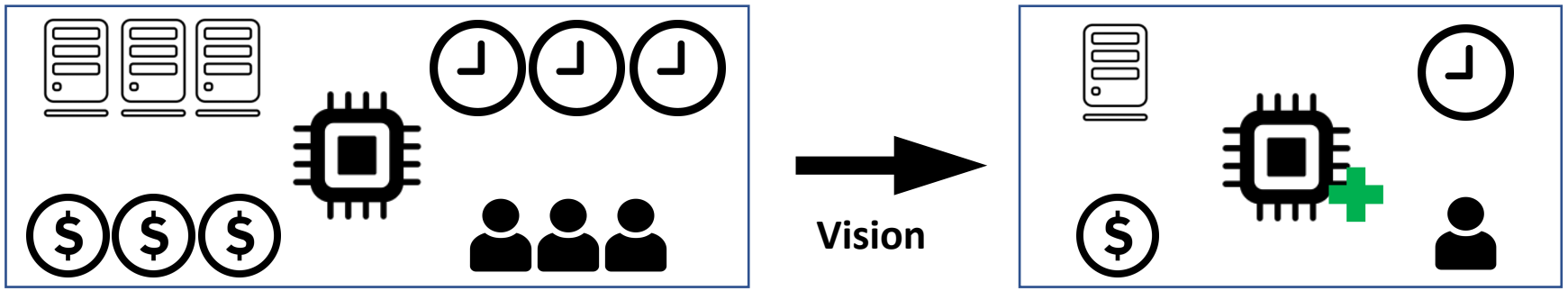
# Look into the IR Drop Model




- Hotspots on **same** layout are triggered at **different** time frames

# Summary and Takeaway

- Problem: Increasing Challenges in Chip Design
  - Cost, time-to-market, reliance on designers, diminishing performance return, ……
- **ML** in chip design
  - Targets routability, timing and interconnect, power and power delivery challenges
  - Less simulation time, faster feedback, less designer effort
- **Automated** and **unified ML** in both design & runtime
  - Reduces months of model development to hours, no developers
  - Benefit the entire chip life cycle



# Publication List (1/2)

- **Zhiyao Xie**, et al. Pre-Placement Net Length and Timing Estimation by Customized Graph Neural Network. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*. Accepted.
- **Zhiyao Xie**, et al. APOLLO: An Automated Power Modeling Framework for Runtime Power Introspection in High-Volume Commercial Microprocessors. *In International Symposium on Microarchitecture (MICRO)*, 2021. (**Best Paper Award**).
- Chen-Chia Chang, Jingyu Pan, Tunhou Zhang, **Zhiyao Xie**, et al. Automatic Routability Predictor Development Using Neural Architecture Search. In *International Conference on Computer Aided Design (ICCAD)*, 2021.
- **Zhiyao Xie**, et al. Net$^2$: A Graph Attention Network Method Customized for Pre-Placement Net Length Estimation. In *Asia and South Pacific Design Automation Conference (ASPDAC)*, 2021.
- **Zhiyao Xie**, et al. Fast IR Drop Estimation with Machine Learning (Invited). In *International Conference on Computer Aided Design (ICCAD)*, 2020.

# Publication List (2/2)

- Rongjian Liang, **Zhiyao Xie**, et al. Routing-Free Crosstalk Prediction. In *International Conference on Computer Aided Design (ICCAD)*, 2020.
- **Zhiyao Xie**, et al. FIST: A Feature-Importance Sampling and Tree- Based Method for Automatic Design Flow Parameter Tuning. In *Asia and South Pacific Design Automation Conference (ASPDAC)*, 2020.
- **Zhiyao Xie**, et al. PowerNet: Transferable Dynamic IR Drop Estimation via Maximum Convolutional Neural Network. In *Asia and South Pacific Design Automation Conference (ASPDAC)*, 2020.
- Yu-Hung Huang, **Zhiyao Xie**, et al. Routability-Driven Macro Placement with Embedded CNN-Based Prediction Model. In *Design, Automation and Test in Europe Conference (DATE)*, 2019.
- **Zhiyao Xie**, et al. RouteNet: Routability Prediction for Mixed-Size Designs Using Convolutional Neural Network. In *International Conference on Computer Aided Design (ICCAD)*, 2018.

# Thank you! Questions?

**Committee:**

Prof. Yiran Chen (Chair)

Prof. Hai Li (Co-Chair)

Prof. Jiang Hu

Prof. James Morizio

Prof. Jeffrey Derby

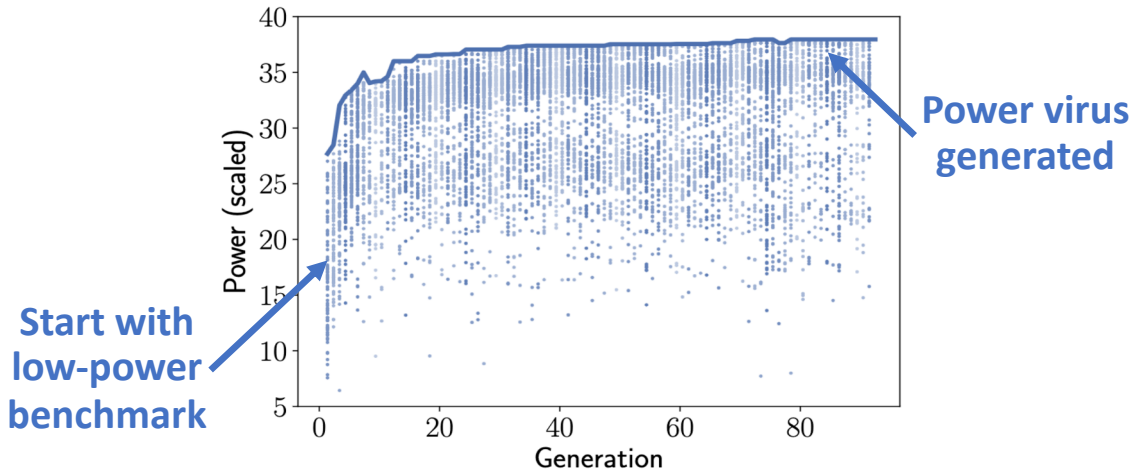
Duke

# Our Proposed Power Modeling Approach

A “diverse” set of random (micro-)benchmarks is critical

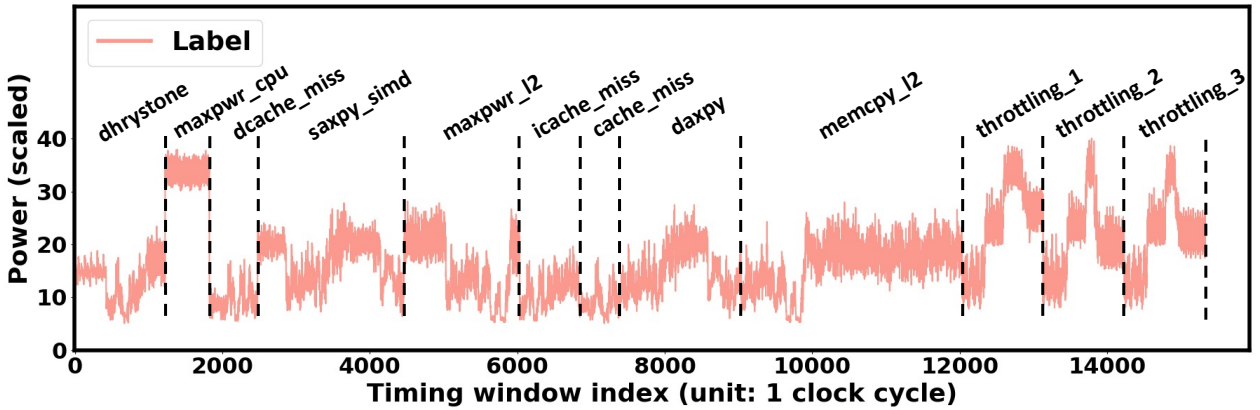
## Training data automatically generated

- Micro-architecture agnostic **genetic algorithm** to automatically generate max-power virus
- A “diverse” set is generated: lower-power in early generations and higher-power in later generations




## Model training & testing

- Experiments on 3GHz 7nm microprocessors **Neoverse N1** and **Cortex A77**
- Testing on Arm power-indicative workloads
  - Steady-state, transient, and throttling regions
  - High- and low-power-consumption regions






Duke UNIVERSITY

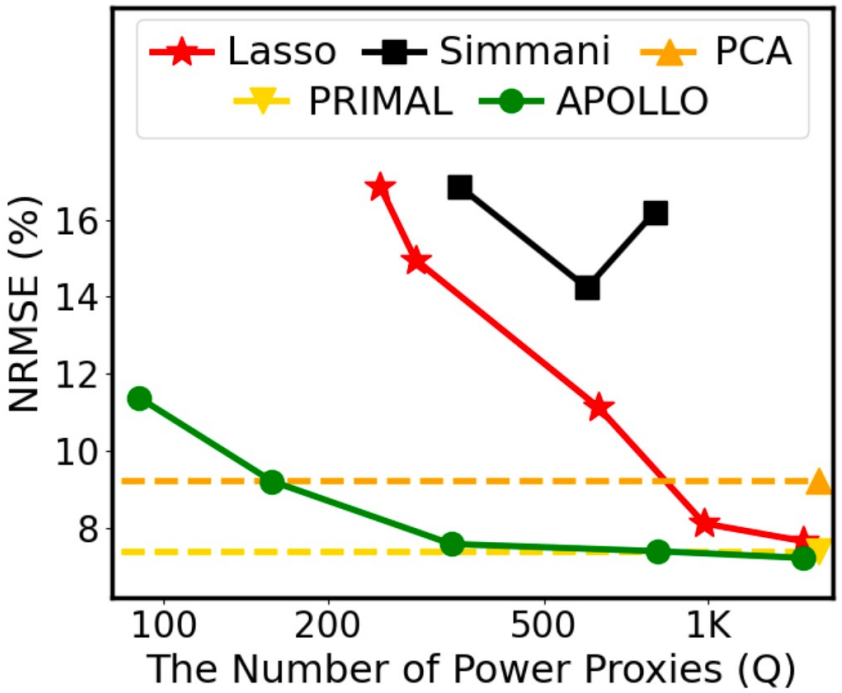
# FAQ: Prior Power Modeling Works

| Methods (Hardware Overhead in Area %) | Demonstrated Application | Model Type | Temporal Resolution | PC / Proxy Selection | Cost or Overhead |
|---|---|---|---|---|---|
| [20, 35, 43, 48, 61] | Design-time software model | Analytical | >1K cycles | N/A | Low |
| [78] | | Proxies | >1K cycles | Automatic or no selection | High |
| [17, 64] | | | | | Medium |
| [79] | | | Per-cycle | | High |
| [19, 42, 44, 72, 76] | | | | | Medium |
| [22] **(300% overhead)** | Design-time FPGA emulation | Proxies | Per-cycle | Automatic | High |
| [75] **(16% overhead)** | | | | | Medium |
| [40] | | | ~100s cycles | | |
| [66] | | | Per-cycle | Hybrid manual/auto | |
| [10, 11, 16, 24, 26, 33, 34, 36, 52, 58, 62, 63, 65, 68] | Runtime monitor | Event Counters | >1K cycles | Manual | Low |
| [38] | | | ~100s cycles | | |
| [23] **(2-20%)**, [51] **(1.5-4%)**, [53] **(7%)** | | Proxies | >1K cycles | Automatic | Medium |
| [80] **(4-10%)**, [81] **(7%)** | | | ~100s cycles | | |
| APOLLO **(0.2% overhead)** | Design-time model Runtime monitor | Proxies | Per-cycle | Automatic | Low |

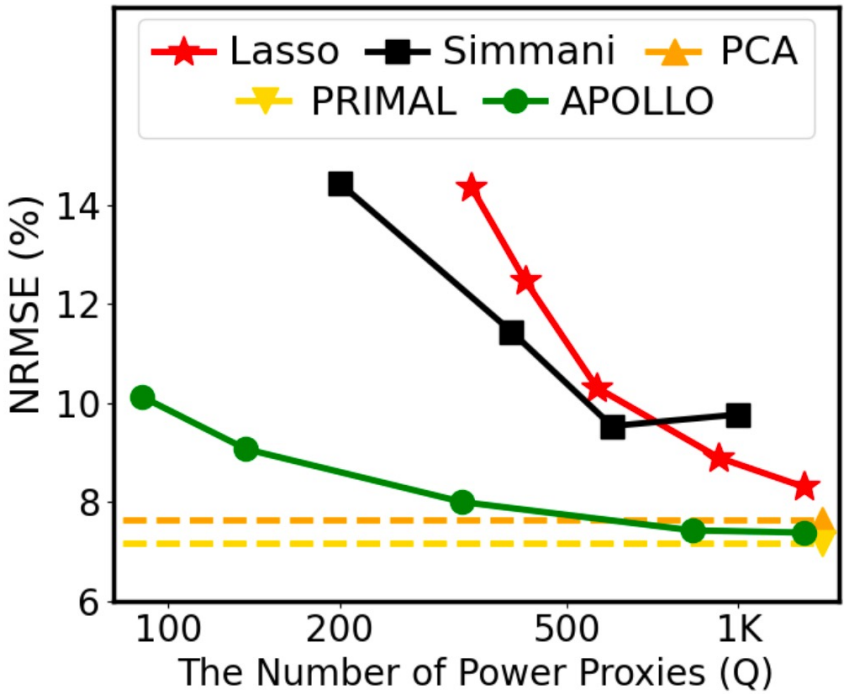
**Comparison among various power modeling approaches. The percentage numbers are area overheads.**

Duke UNIVERSITY

# FAQ: Accuracy Comparison



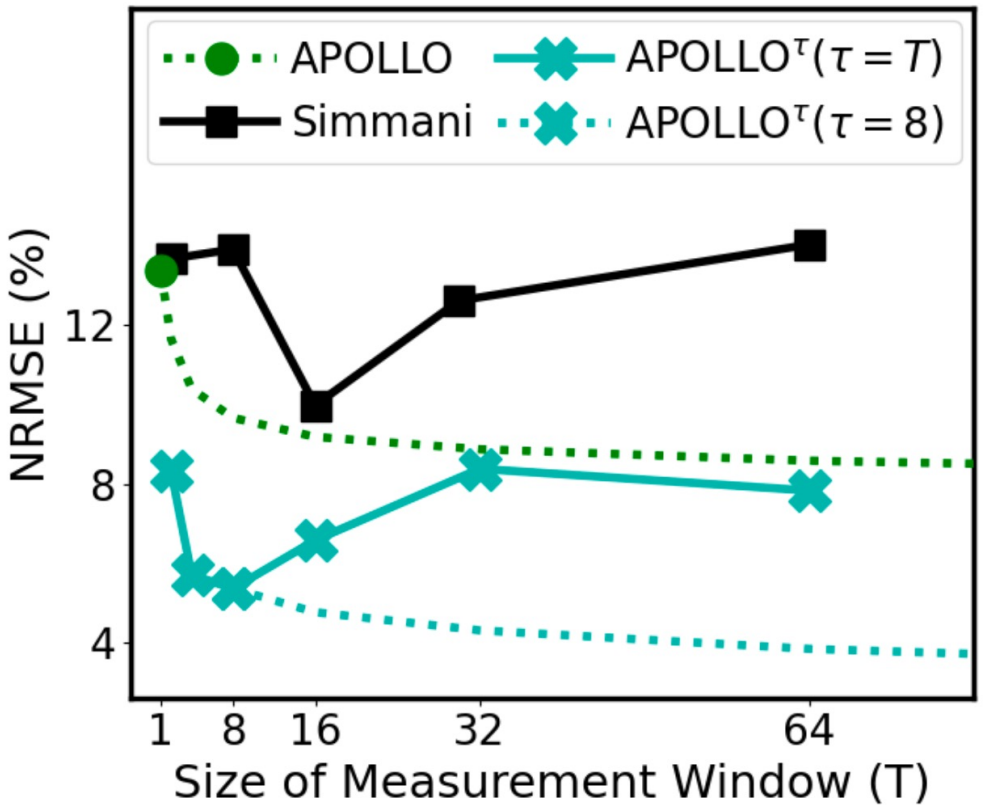

**Neoverse N1 CPU core**




**Cortex-A77 CPU core**

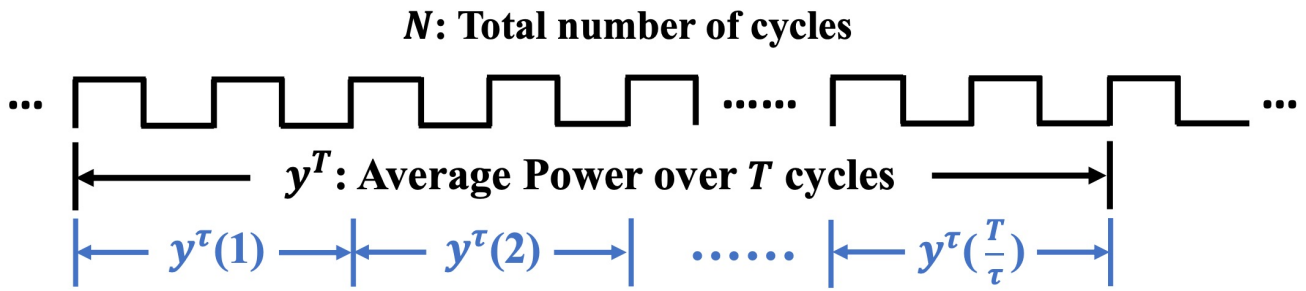
Duke UNIVERSITY

# FAQ: Accuracy with Multi-cycle Measurement Window




**Q = 200** for Simmani

**Q = 70** for APOLLO



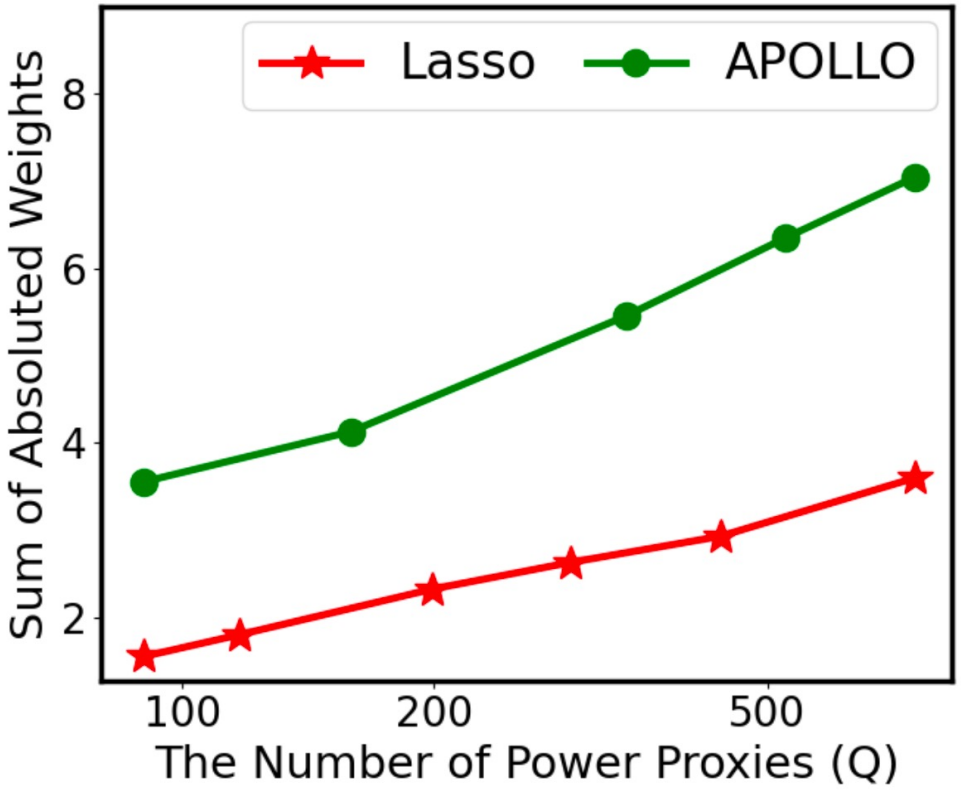

**During inference:**

$$p^{\tau}(1) = \sum_{j=1}^{Q} \omega_j \cdot x_j^{\tau}(1) = \sum_{j=1}^{Q} \omega_j \cdot \frac{1}{\tau} \sum_{i=1}^{\tau} x_j[i] = \frac{1}{\tau} \sum_{i=1}^{\tau} \sum_{j=1}^{Q} \omega_j \cdot x_j[i]$$

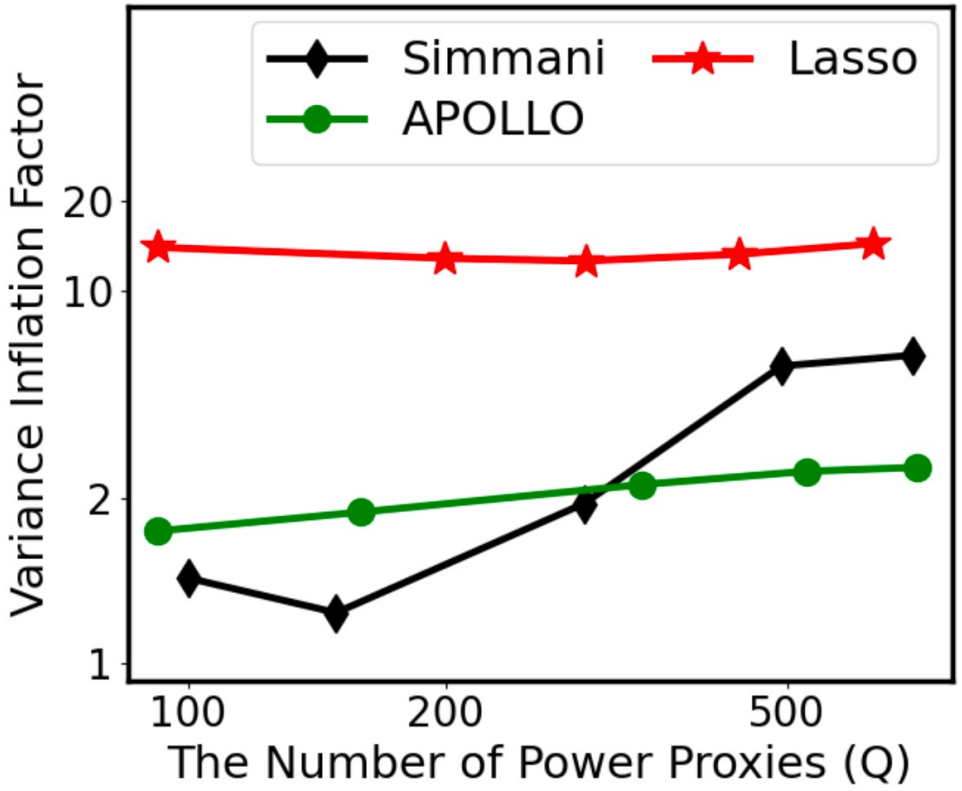
$$\text{Thus,} \quad p^{T} = \frac{1}{T/\tau} \sum_{k=1}^{T/\tau} p^{\tau}(k) = \frac{1}{T} \sum_{i=1}^{T} \sum_{j=1}^{Q} \omega_j \cdot x_j[i] \qquad (9)$$

Duke UNIVERSITY

# FAQ: Model Discussion




**Sum of all absolute weights**

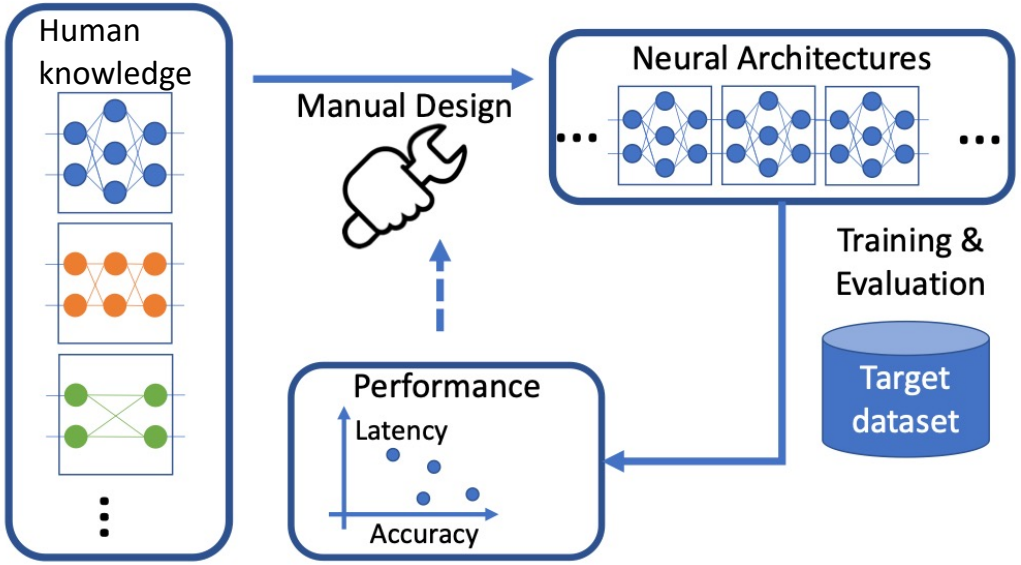



**Variance inflation factors (VIF)**

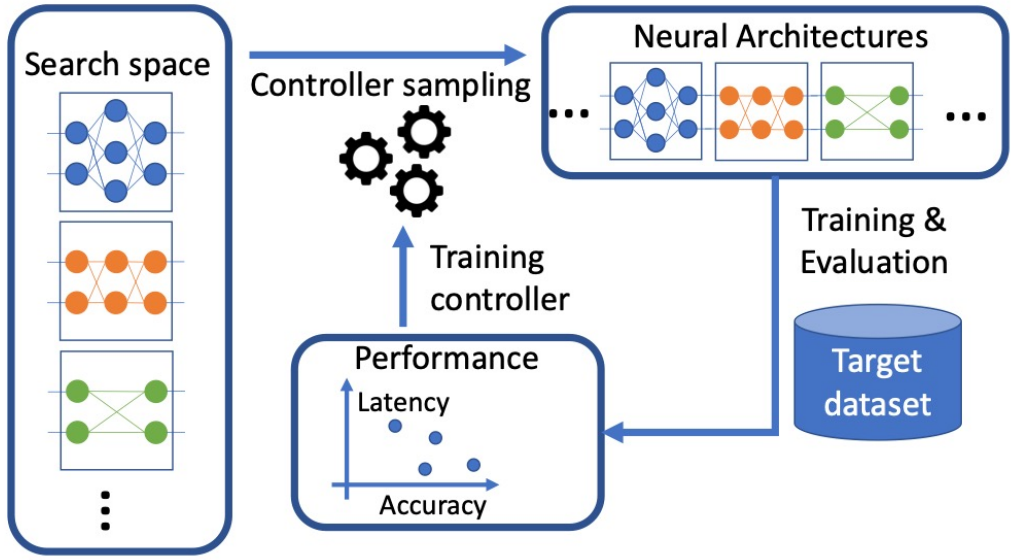
Duke
UNIVERSITY

# Neural Architecture Search

- Neural Architecture Search (NAS) enables design automation of ML models efficiently without human interventions




Traditional DNN Design



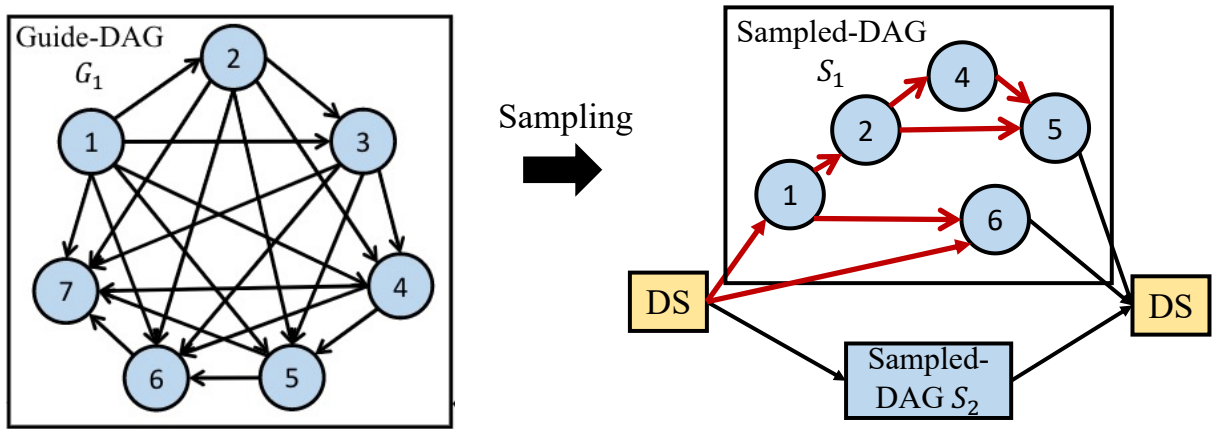

**Neural Architecture Search**

Wu, Bichen, et al. "Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

Duke UNIVERSITY

# Search Strategy

- Define a weight on each edge to control its sampling probability
- Sample edges from the guide-DAG with probabilities
- Updating flow
  - Preprocessing
  - Loop: while the performance of model not converges
    - **Subsample guide-DAGs according to the weights**
    - Train the model on our dataset to get performance metrics
    - Update connection weights
      - Higher performance leads to higher weights on edges




Duke UNIVERSITY

# Experimental Results – Accuracy

Comparison of the violated net count prediction

| Models | Kendall's $\tau$ on designs (#nets) | | | | Kendall's $\tau$ on all 74 designs | Pearson's correlation on all 74 designs |
|---|---|---|---|---|---|---|
| | s349 (270) | mem_ctrl (9.3k) | b17 (33.8k) | DSP (73.1k) | | |
| RouteNet [3] | 0.3620 | 0.1547 | 0.1779 | 0.4414 | 0.5264 | 0.7224 |
| **NAS-crafted model** | **0.6369** | **0.4657** | **0.2683** | **0.7302** | **0.5572** | **0.7930** |

Comparison of the DRC hotspot detection

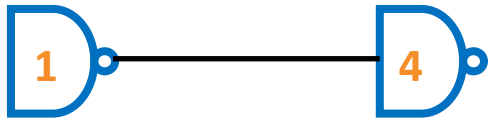
| Models | ROC-AUC on designs (#nets) | | | | ROC-AUC on all 74 designs |
|---|---|---|---|---|---|
| | s349 (270) | mem_ctrl (9.3k) | b17 (33.8k) | DSP (73.1k) | |
| RouteNet [3] | 0.829 | 0.844 | 0.902 | 0.866 | 0.847 |
| PROS [6] | 0.487 | 0.483 | 0.478 | 0.489 | 0.676 |
| cGAN [4] | 0.516 | 0.515 | 0.521 | 0.517 | 0.510 |
| **NAS-crafted model** | **0.865** | **0.891** | **0.911** | **0.884** | **0.865** |

# Method: Capture the Global Information

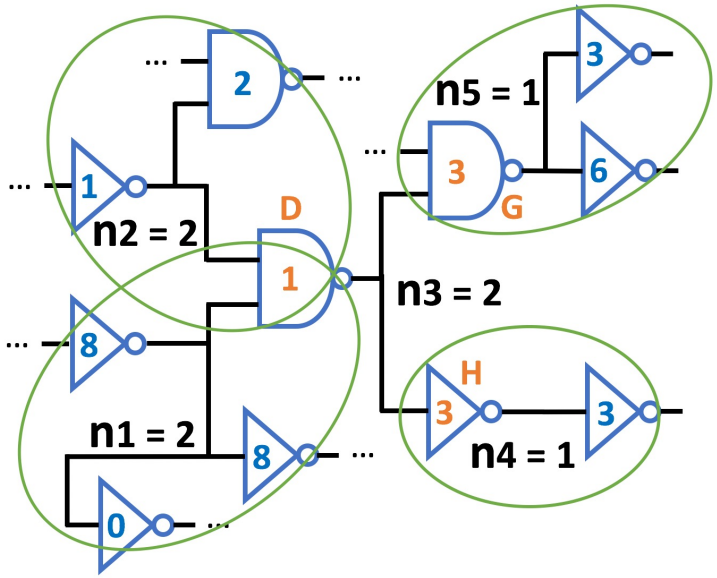
- To build the accurate version Net$^{2a}$:
  - Capture more global information (the topology of the whole netlist)
  - Capture the information by fast clustering (partitioning) on each netlist using h-metis
  - 1. clustering when nets viewed as node; 2. clustering when cells viewed as node
  - **Different cluster IDs indicates larger distance after placement**

Numbers on cell are cluster IDs:

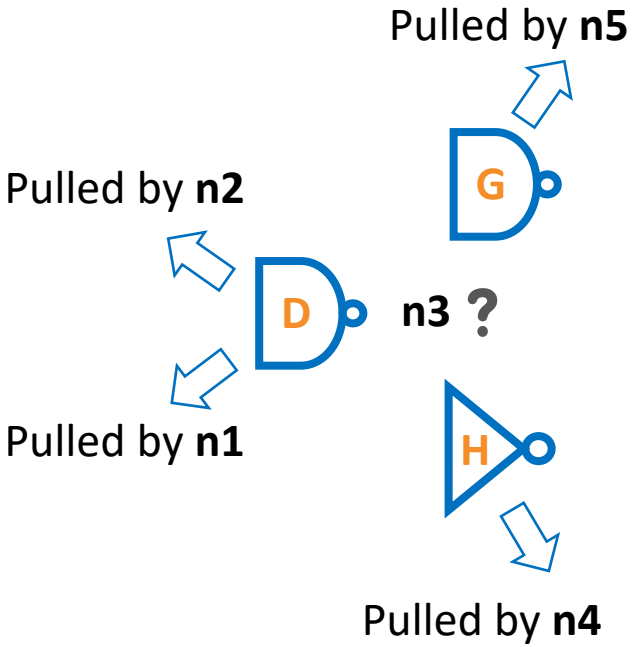

The same cluster IDs (1 == 1).
It indicates shorter distance.



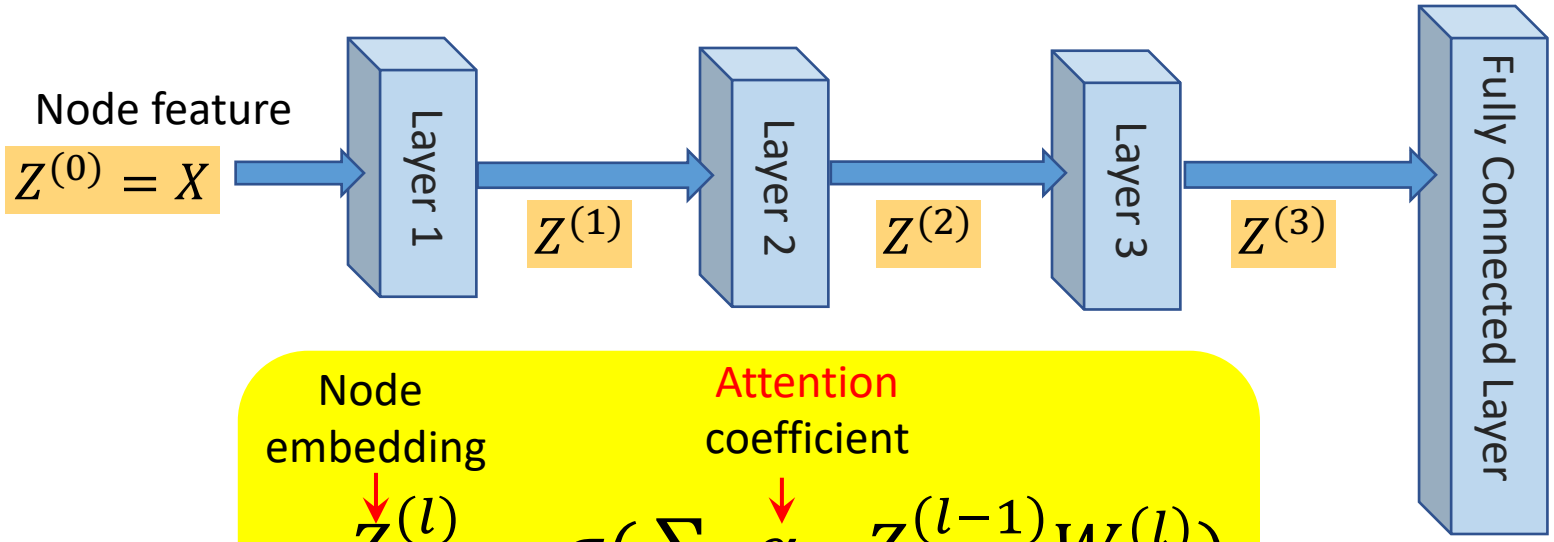

The different cluster IDs (1 != 4).
It indicates longer distance.




Example: study the size of **n3**.
Numbers on cell & net are cluster IDs.



# Graph Attention Network (GAT)




Node embedding

Attention coefficient

$$Z_i^{(l)} = \sigma(\sum_j \alpha_{ij} Z_j^{(l-1)} W^{(l)})$$

Activation function

Trainable weight

For node *i* and all its neighbors *j* :

$$\alpha_{ij} = \text{softmax}(\tau_{ij})$$

$$\tau_{ij} = \text{LeakyReLu}(a^{(l)}[W^{(l)} Z_i^{(l-1)} || W^{(l)} Z_j^{(l-1)}])$$

Trainable vector

# Experimental Results on Timing - Delay

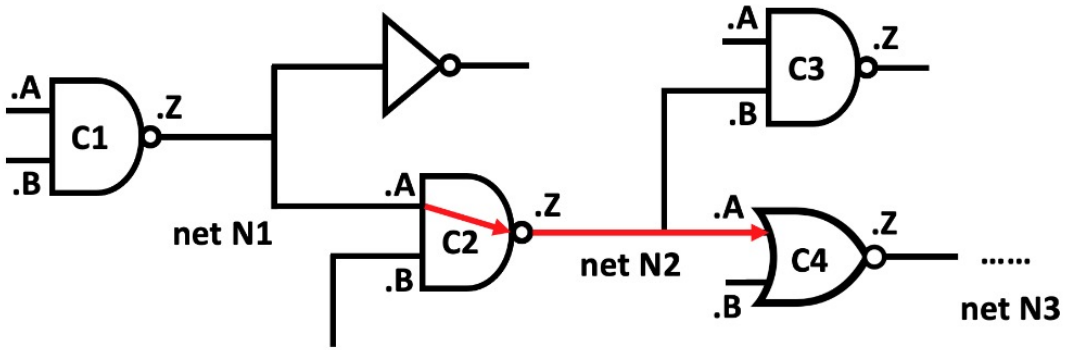



TABLE V: Pre-placement Path Slack Prediction Accuracy.

| | report_timing | | Time$^{f}$ | | Time$^{a}$ | |
|---|---|---|---|---|---|---|
| | Error | $R^2$ | Error | $R^2$ | Error | $R^2$ |
| Mean | 0.38 ns | 0.39 | 0.16 ns | 0.86 | 0.11 ns | 0.91 |
| Median | 0.18 ns | 0.77 | 0.07 ns | 0.95 | 0.05 ns | 0.97 |