Truly Intelligent Circuit Design and Implementation

Zhiyao Xie

Dept. Electrical & Computer Engineering

Duke University

Outline of My Talk

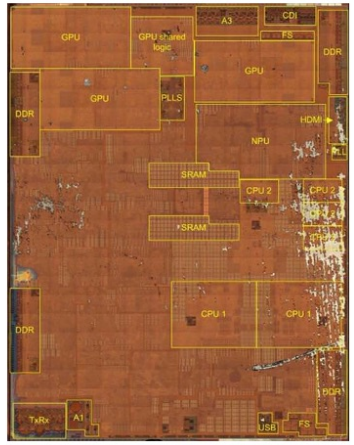
- **Part 1: My Ph.D. Works**
- **Part 2: My Future Plan**

Electronic Devices are Everywhere

*These images are found in public domain

Designers Try to Deliver Generational Gains

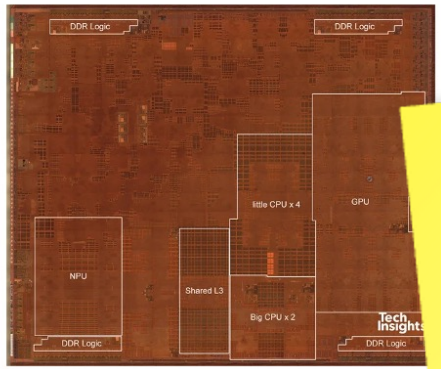
iPhone 8, X



Apple A11

10nm
4.3 B trasistors

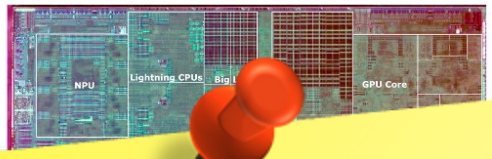
iPhone XS, XR



Apple A12

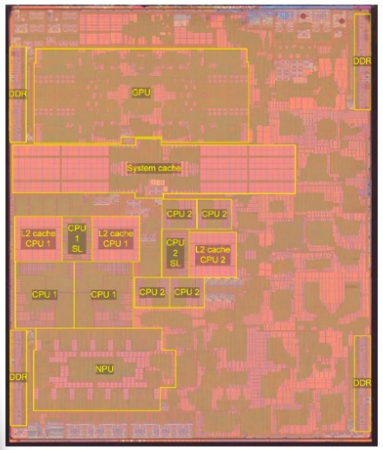
7nm
6.9 B trasistors

iPhone 11



Looks good!
Any challenges?

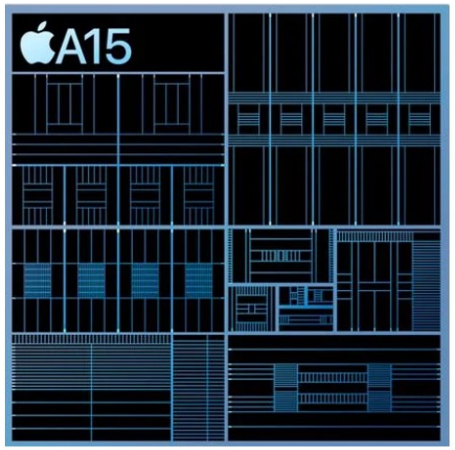
iPhone 12



Apple A14

5nm
11.8 B trasistors

iPhone 13



Apple A15

5nm
15 B trasistors

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

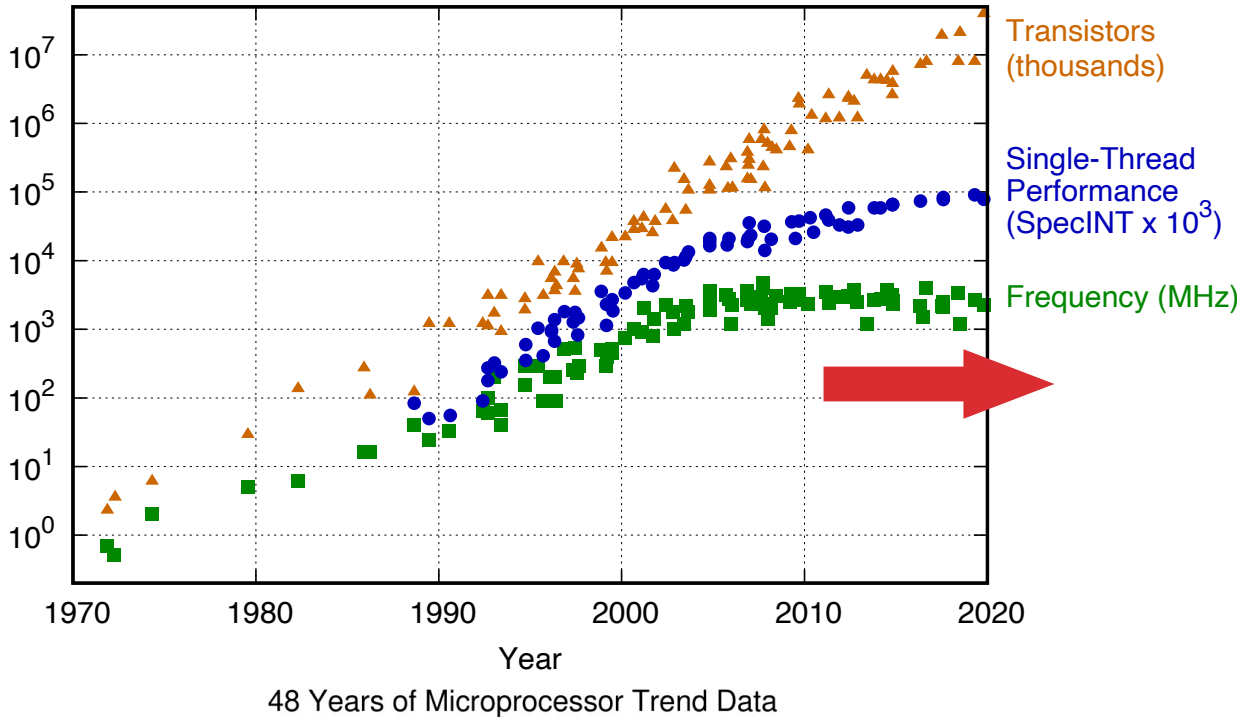
*Source: TechInsights Inc.

Duke UNIVERSITY

Chip Design Challenges

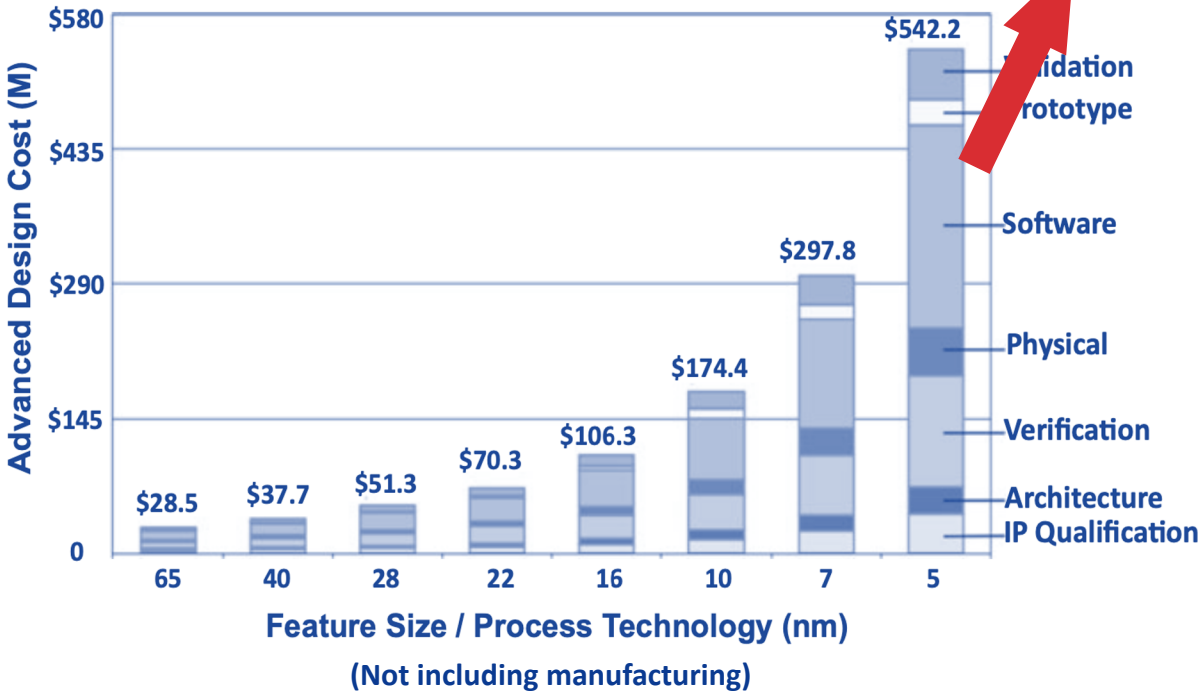
Diminishing performance gain and increasing design cost

Per-Core Performance Gain is Diminishing



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

IBS Design Cost is Skyrocketing

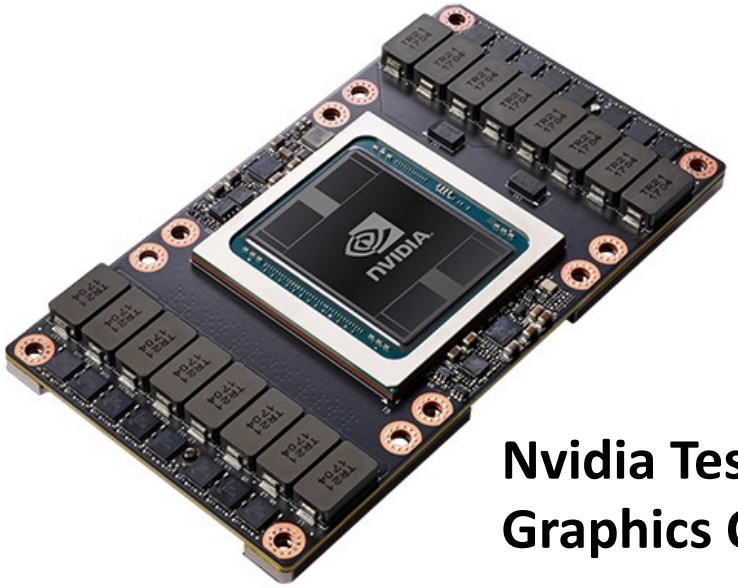


International Business Strategies, 2020

Duke UNIVERSITY

Chip Design Challenges

Not only costly, also long turn-around time



Nvidia Tesla V100 Graphics Card

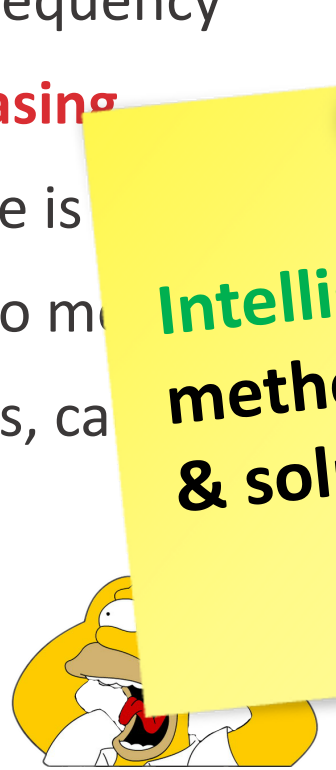
It took **several thousand** engineers **several** years to create, at an approximate development cost of **$3 billion**. – Jensen Huang, CEO of Nvidia

Nvidia GPU Technology Conference (GTC), 2017

This is Real Problem!

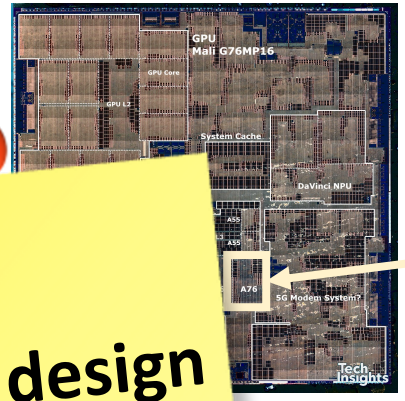
Challenges at advanced node

- **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
- ……



Intelligent design methodologies & solutions!

Inefficient chip design methodologies

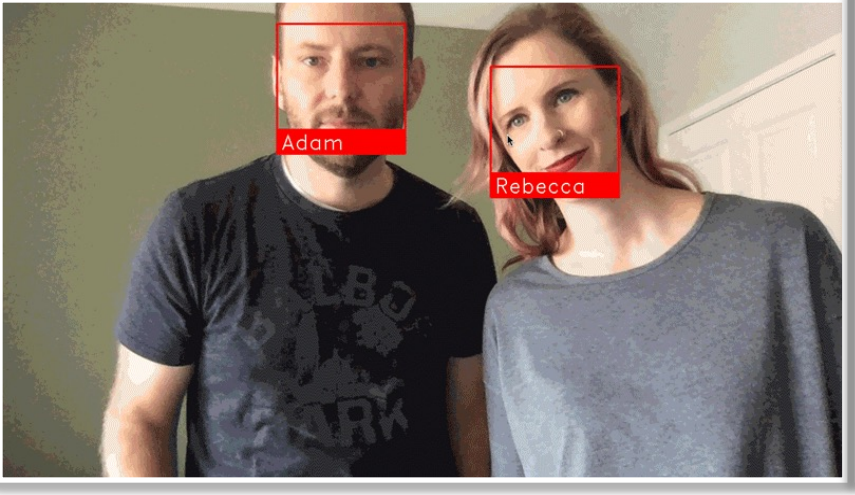


For one Arm CPU core with ~3 million gates

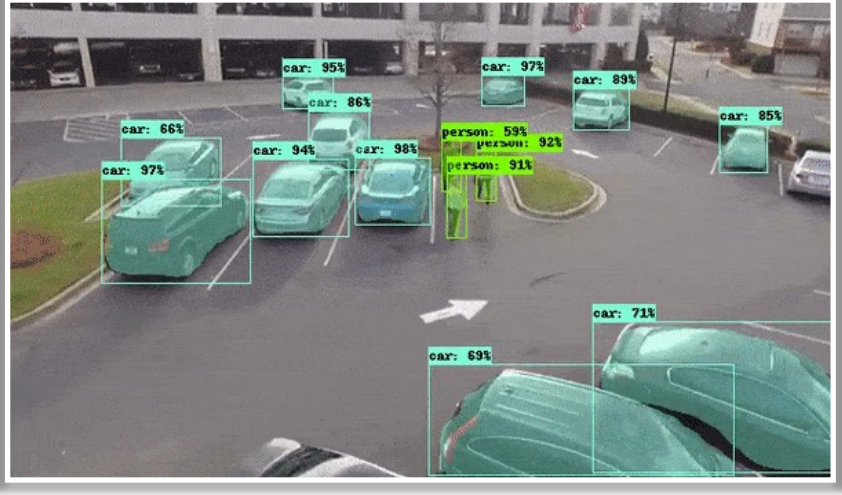
- e power simulation takes **~2 weeks**
- ation in physical design take **~1 week**
- s **repeatedly** constructed from scratch
- olutions rely on designer **intuition**
- ……

Source: The Kirin 990 SoC. TechInsights Inc.

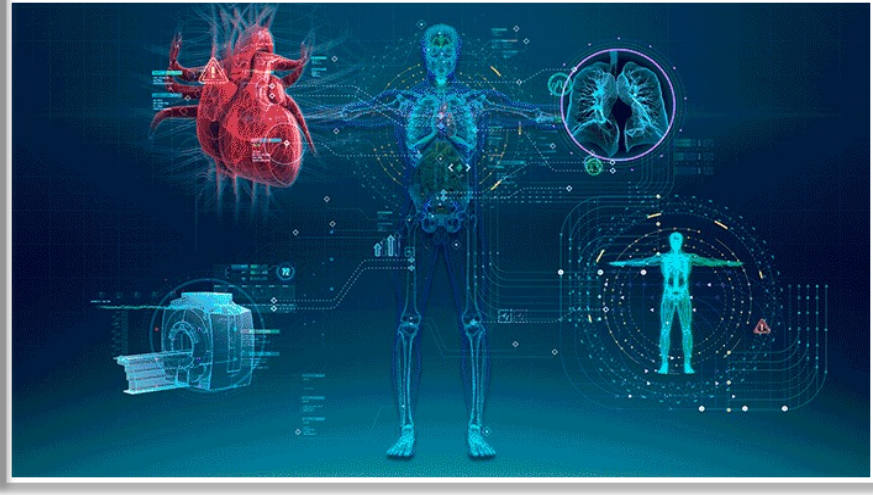
Duke UNIVERSITY



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

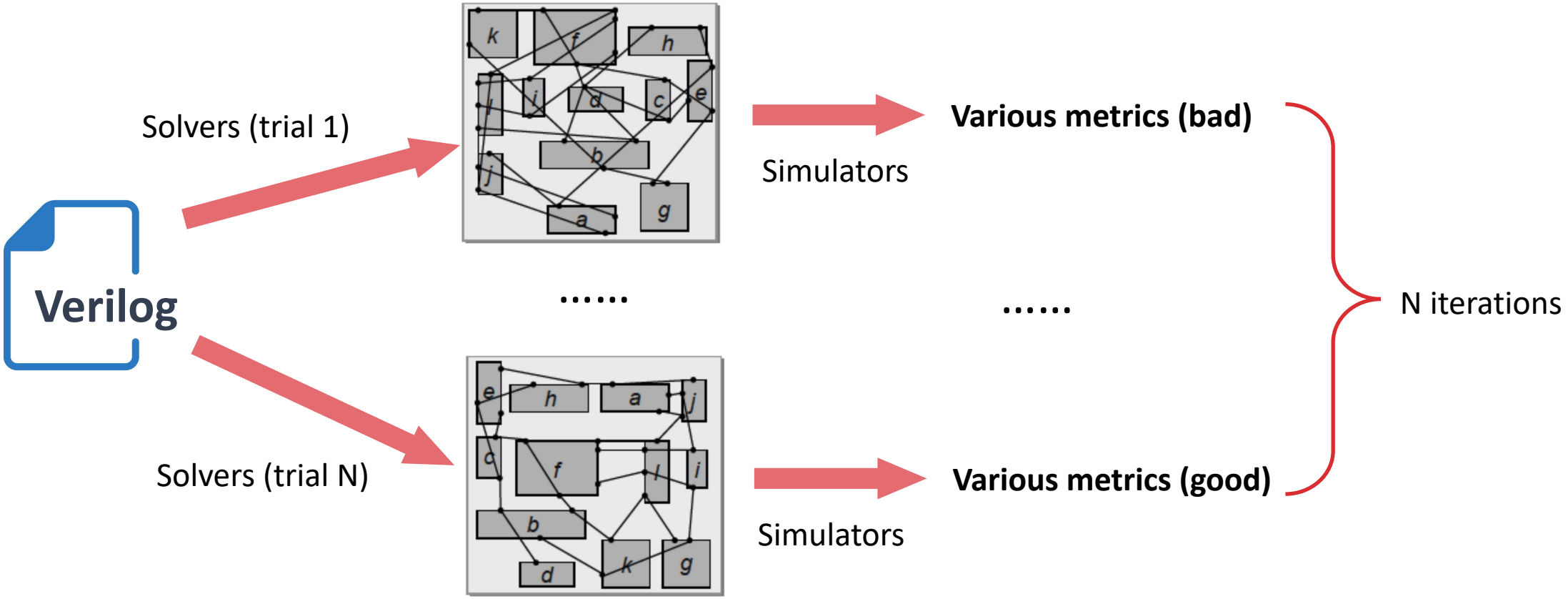
Financial Service

HPC

Security

Gaming

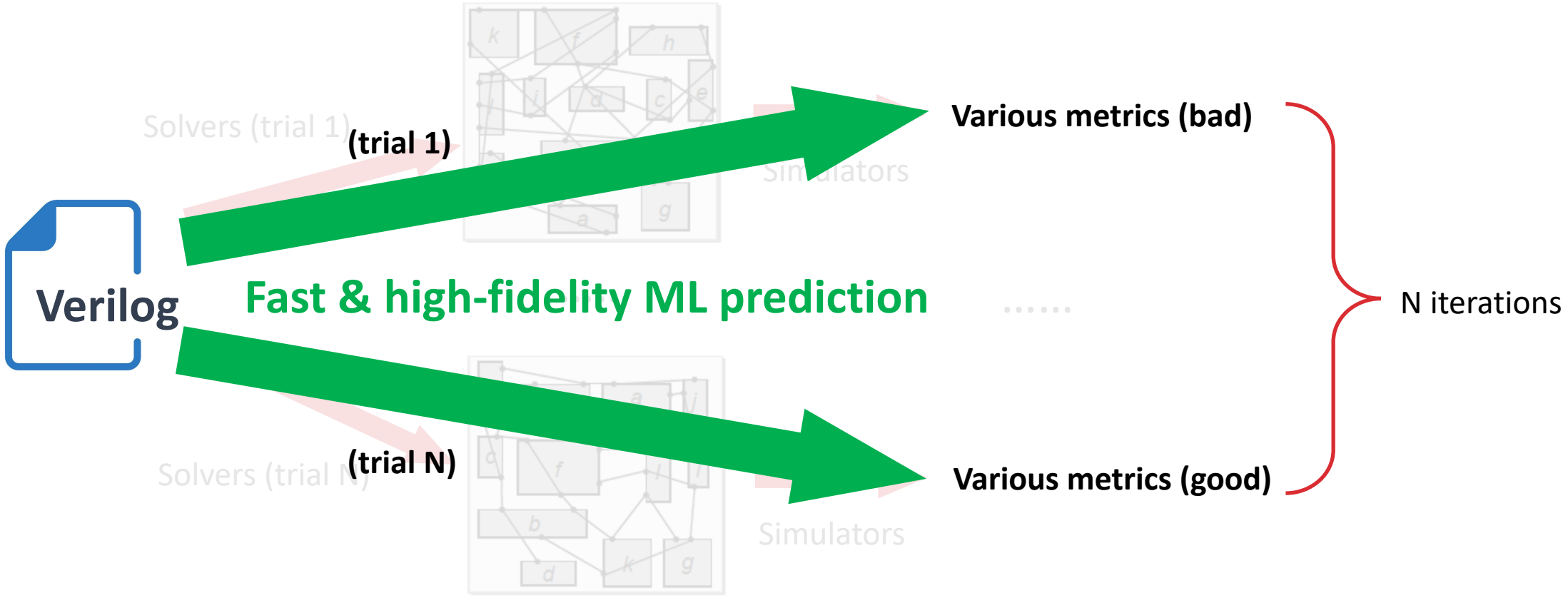
Why ML/Intelligence Helps Circuit Design?



- Producing solutions **repeatdly from scratch**

*Source: Kahng et al., VLSI physical design

Why ML/Intelligence Helps Circuit Design?



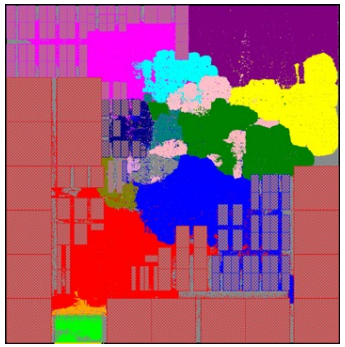
- Producing solutions **repeatdly from scratch**
- Why not learn from prior solutions?

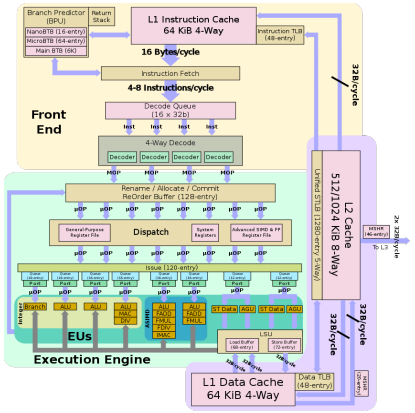
Duke UNIVERSITY

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 number of publications on ML for chip design automation

UT Austin

UCSD

CUHK

TAMU

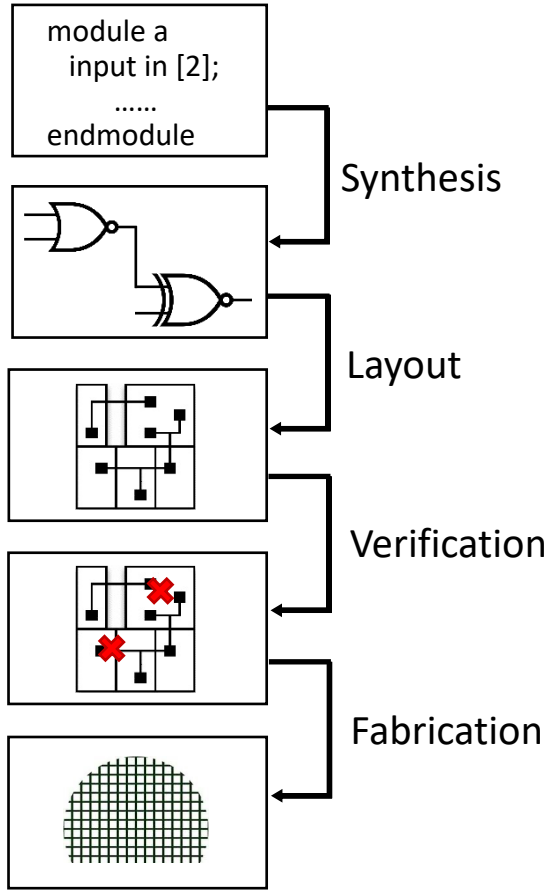
Duke

Cornell

Google

Nvidia

GaTech



ML for EDA in commercial tools

Cadence Innovus™

Synopsys ICC™ II

.....

ML for Chip Design

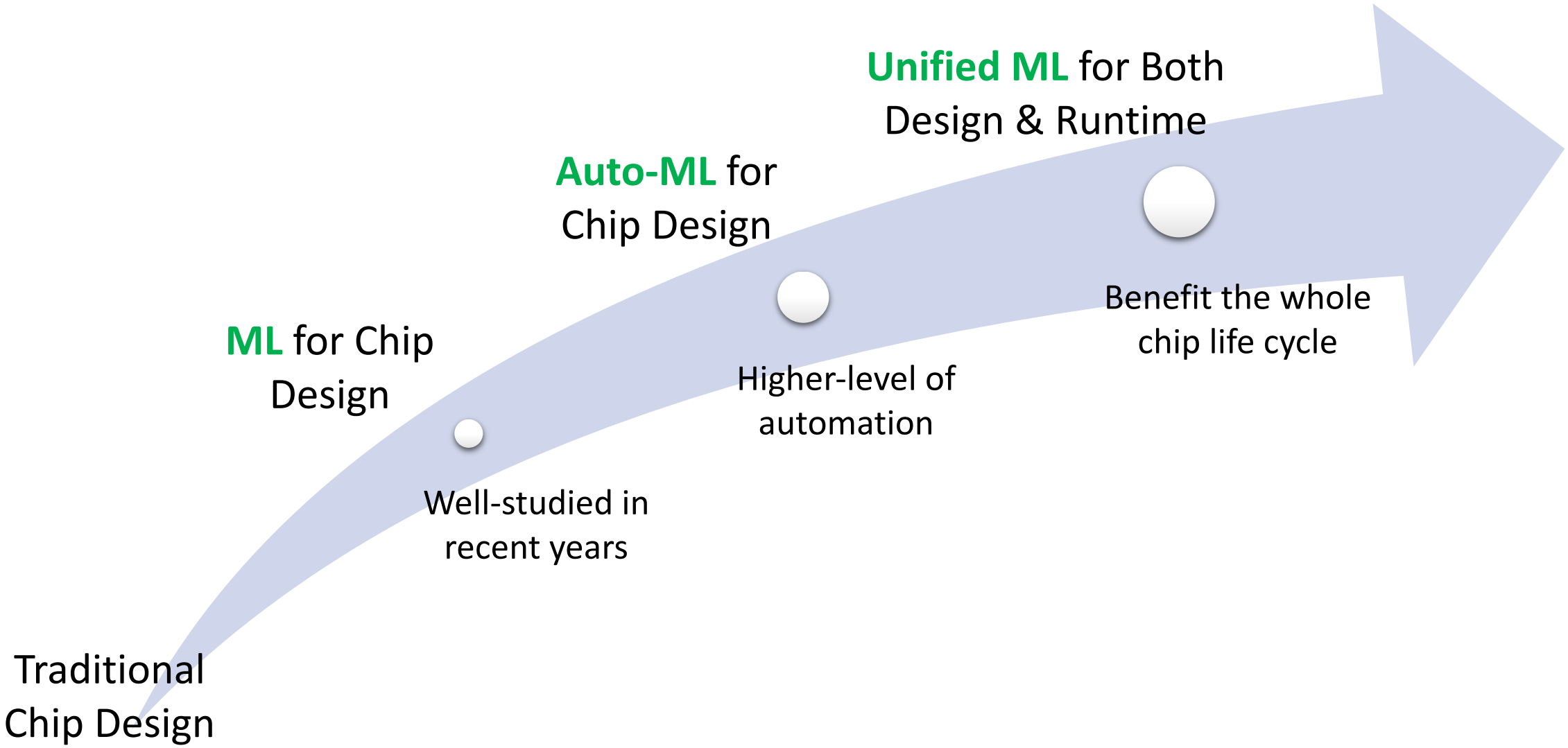
Traditional Chip Design

DARPA Electronics Research Initiative (ERI) – Design

Goal: 24 hours turnaround time & no human

Duke UNIVERSITY

What I Believe We Should Target



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] (under review)

Area

Routability Challenges

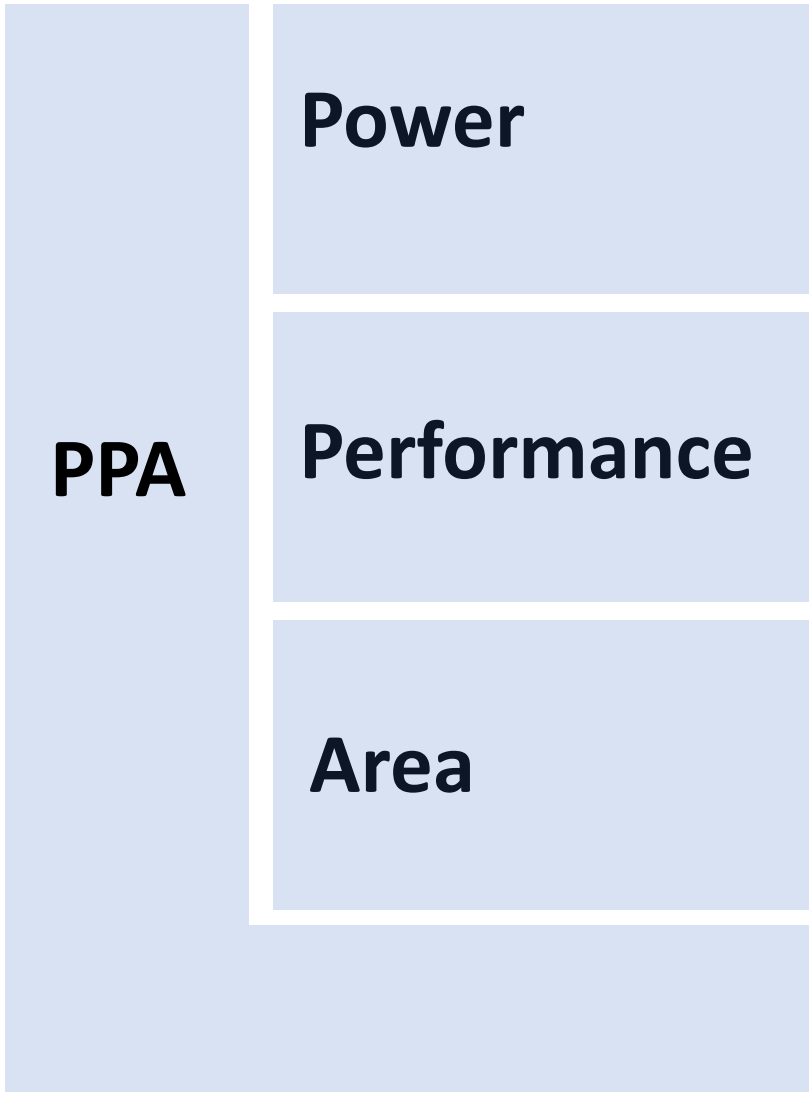
[ICCAD'18], [DATE'18], [ICCAD'21]

Overall Flow Tuning

[ASPDAC'20]

Covered in this talk

My Related Works



Power & Power Delivery Challenges

[ICCAD'20], [ASPDAC'20],

[MICRO'21] (Best Paper Award)

Timing & Interconnect Challenges

[ICCAD'20], [ASPDAC'21],

[TCAD'21] (under review)

Routability Challenges

[ICCAD'18], [DATE'18], [ICCAD'21]

Overall Flow Tuning

[ASPDAC'20]

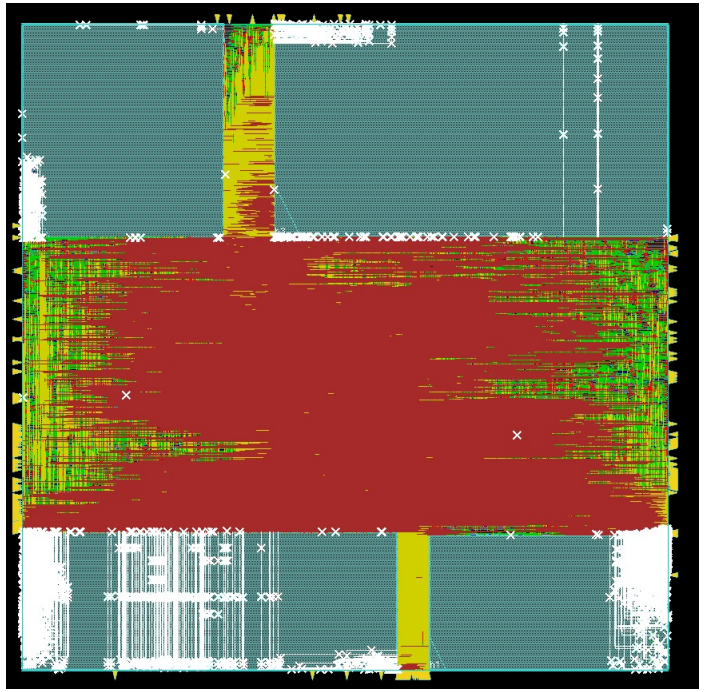
Covered in this talk

Case Study 1:

Routability Challenges

Routability Background

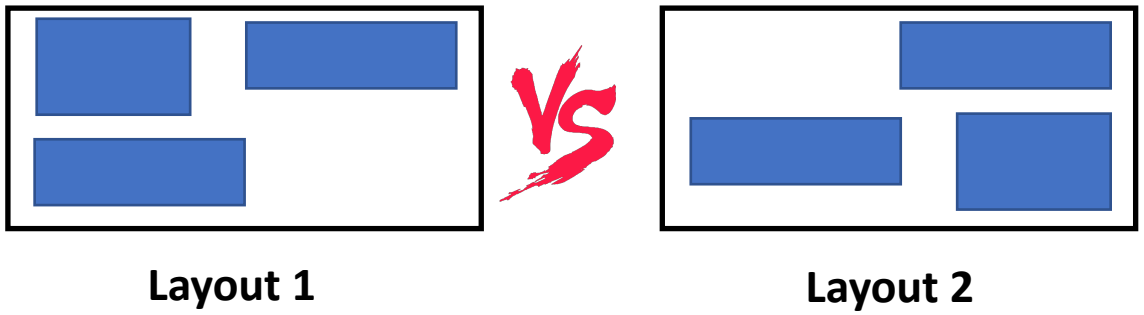
- Design Rule Checking (DRC)
  - Meeting manufacturing requirements
  - Less DRC violations (DRV) -> better routability
- DRV mitigation at early stages
  - Requires routability prediction/estimation
- Previous routability (DRV) estimations
  - ML model on small cropped regions
  - **Limited** receptive field and **missing** global information

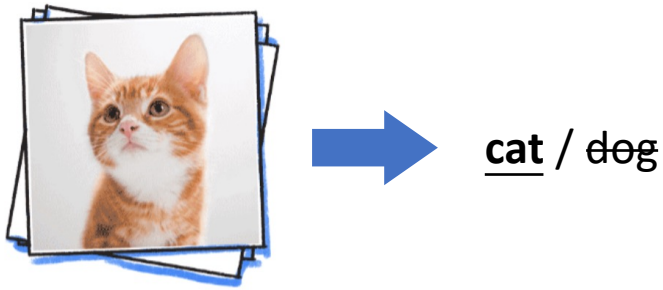


DRC violations (white) on circuit layout

First Deep Learning Method for Routability Prediction

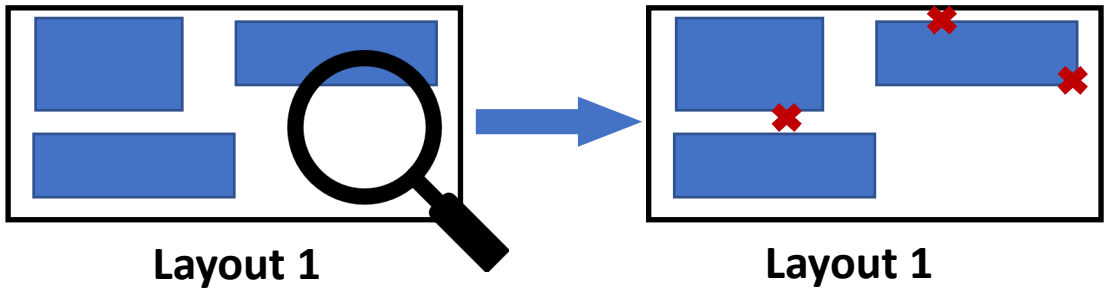
- Task 1: which one will result in less DRV count?

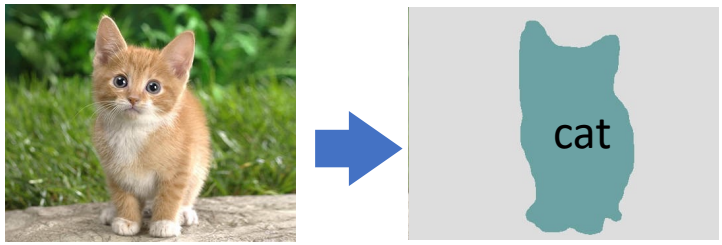




Customized CNN methods

- Task 2: where are DRC violations?



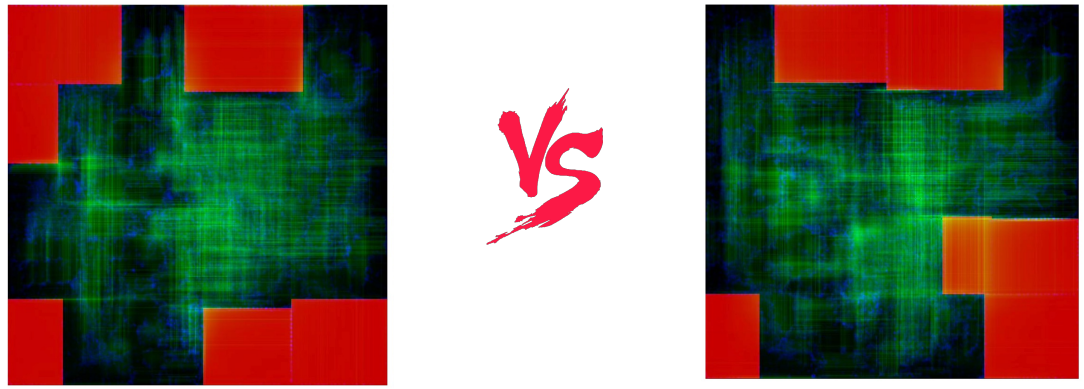


Customized FCN methods

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

First Deep Learning Method for Routability Prediction

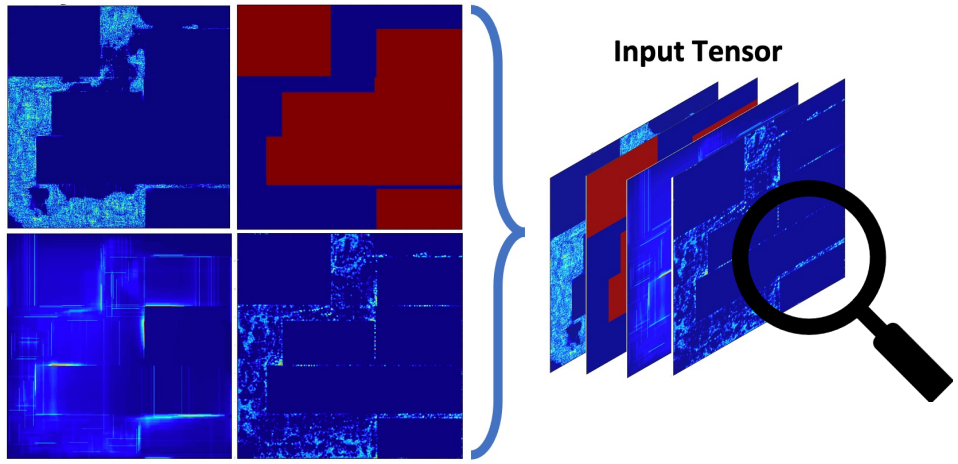
- Task 1: which one will result in less DRV count?



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

In seconds, with similar accuracy

- Task 2: where are DRC violations?



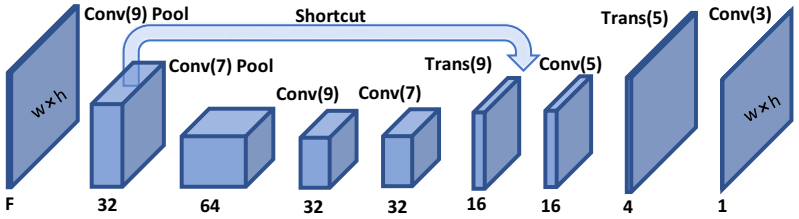
- Requires detailed routing
  ~~More hours~~ * Iterations

In seconds, outperform previous works

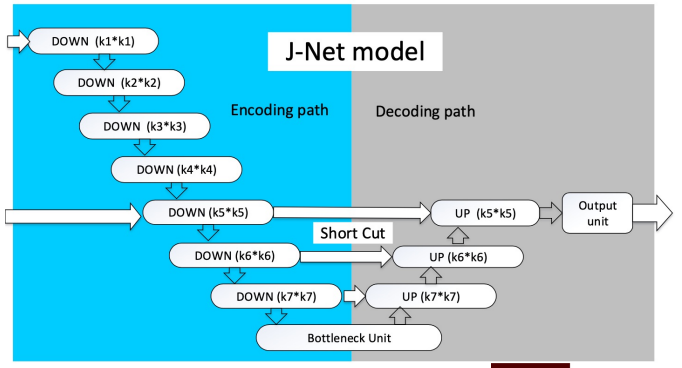
RouteNet [Xie, et al., ICCAD'18]

Duke UNIVERSITY

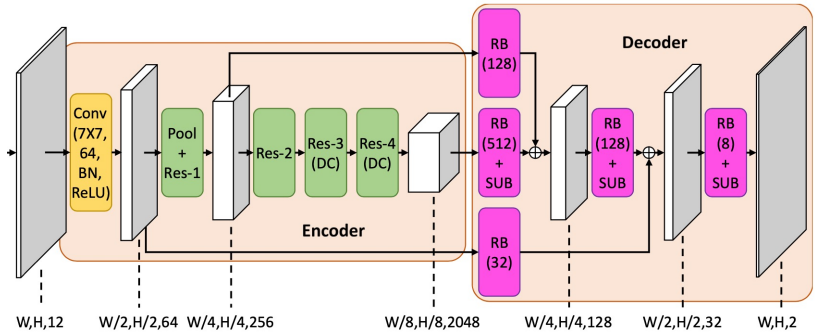
Many Excellent Deep Learning Methods



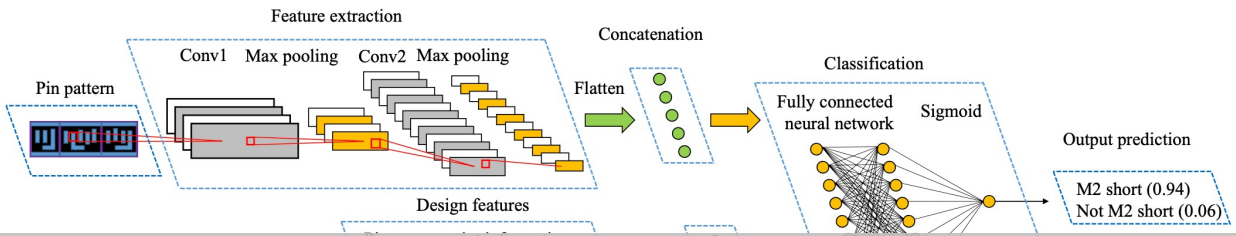
RouteNet [Xie, et al., ICCAD'18] Duke UNIVERSITY



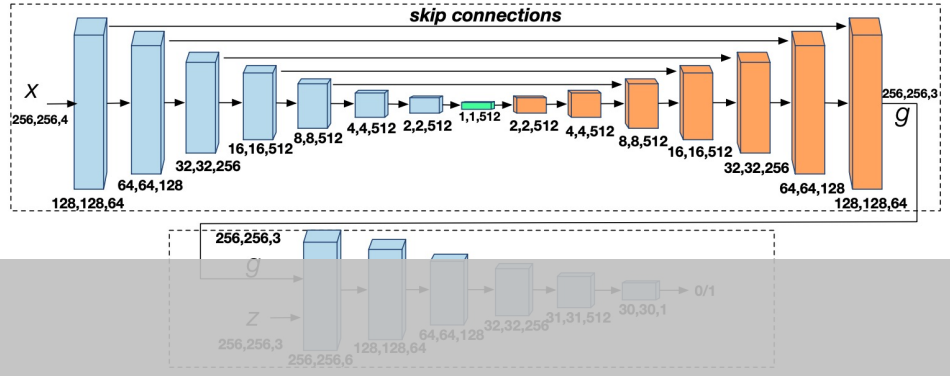
J-Net [Liang, et al., ISPD'20]



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

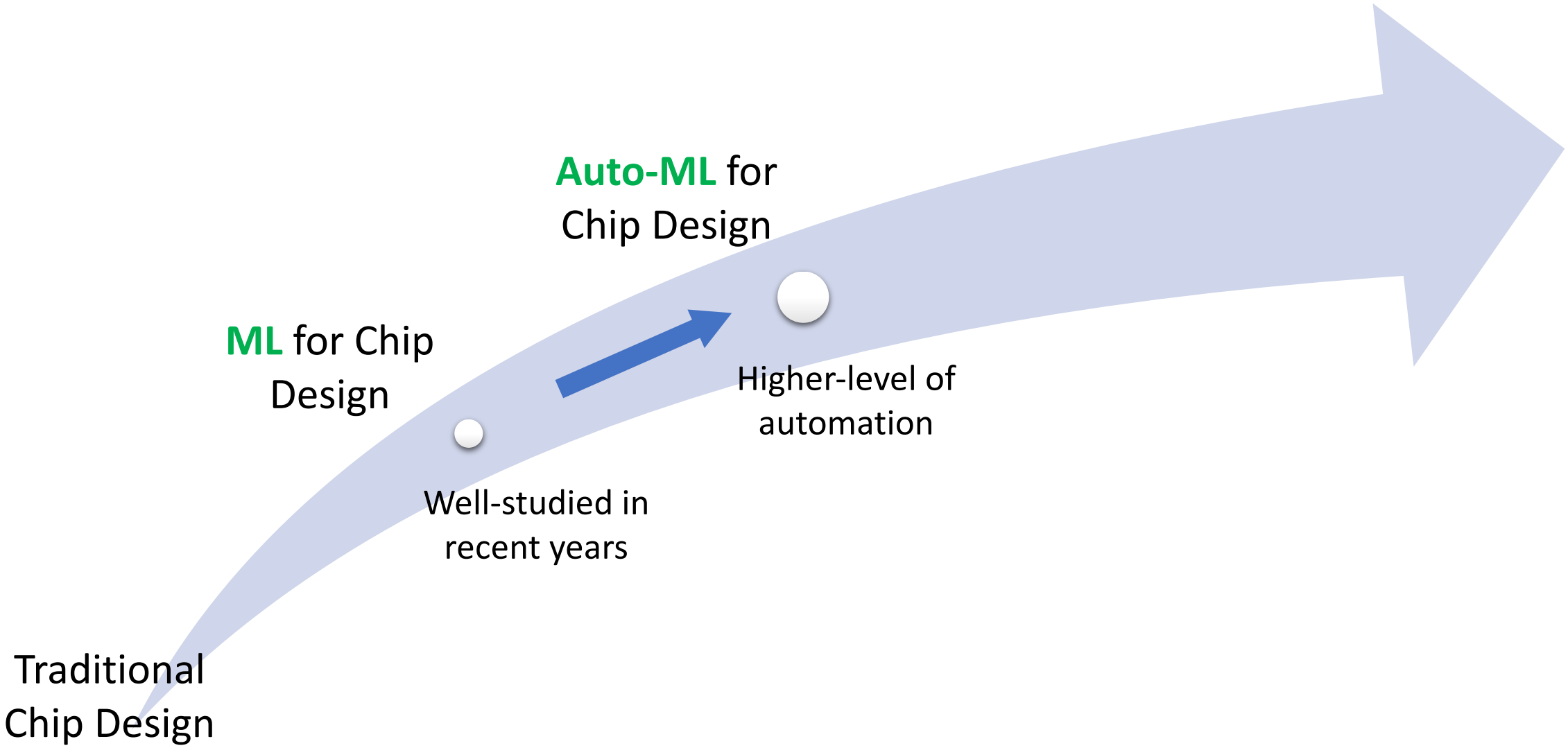


[Yu, et al., DAC'19]



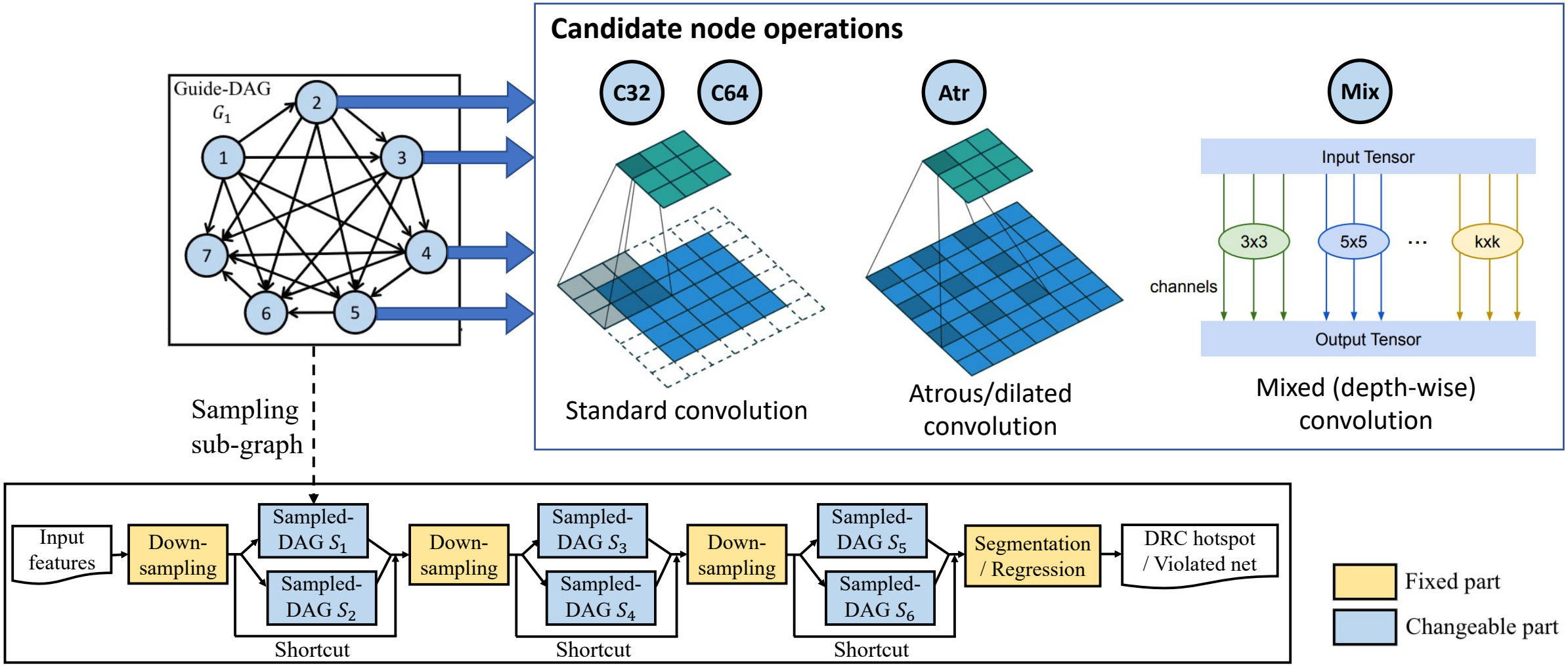
Tremendous Engineering Efforts Required!

What I Believe We Should Target



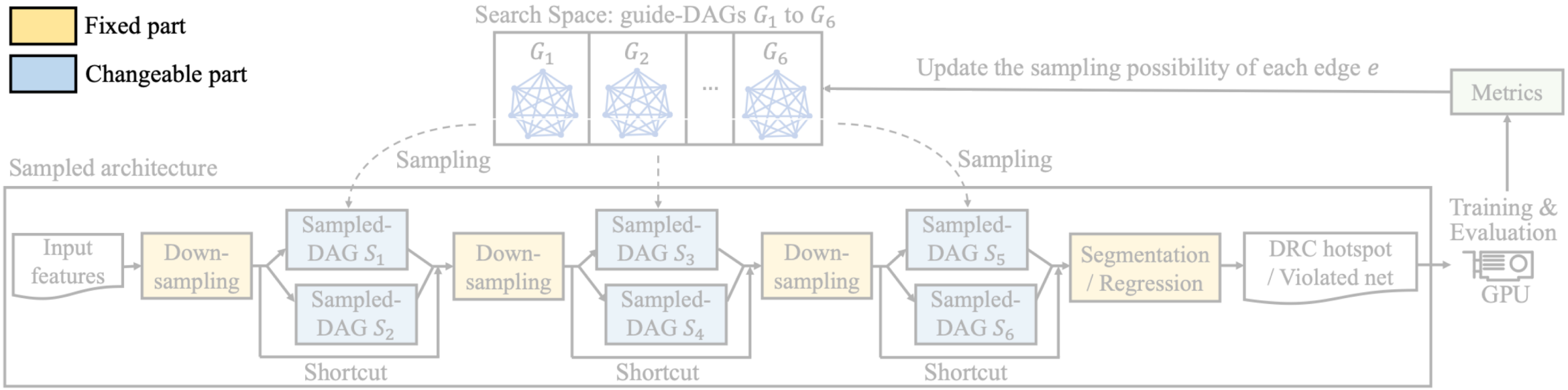
Duke UNIVERSITY

Automatic Estimator Development – Search Space



[Chang, et al., ICCAD'21]

Automatic Estimator Development – Searching Algorithm



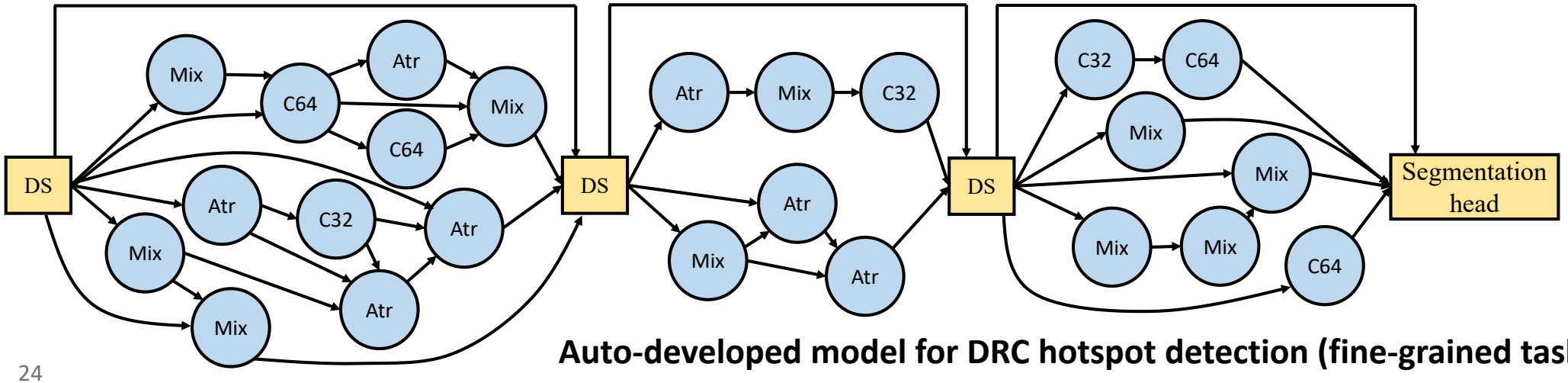
1. Sample from the completely-ordered graph (G_i) to get (S_i)
2. Evaluate the sampled model by training and testing
3. Update the sampling probability by evaluation result

- **Result**: **outperforms** previous works in both tasks; developed without human in **one day**

Duke UNIVERSITY

Auto-developed Model Structures

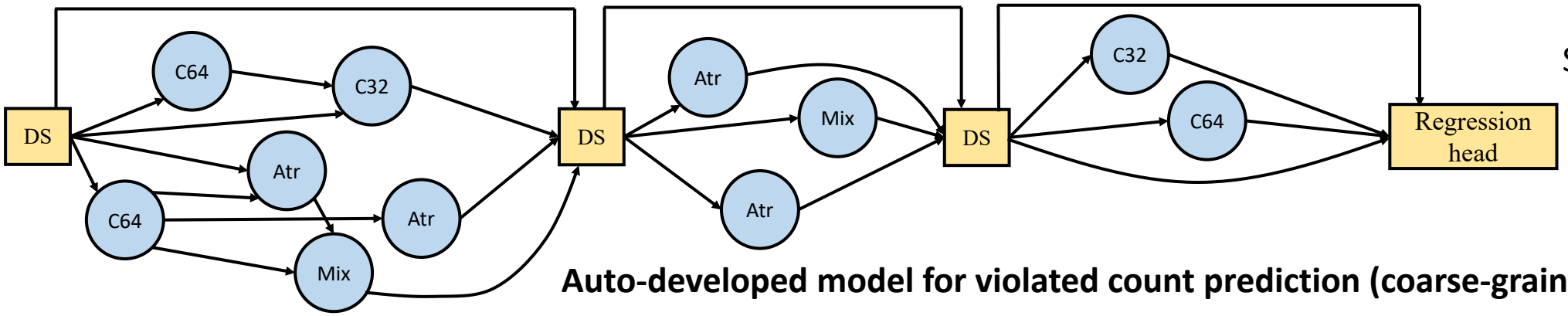
- Human-designed models:
  - Highly hierarchical and organized architecture
  - Limited operation types
- Auto-developed model:
  - Construct parallel branches and flexible interactions
  - Supports different operators



Auto-developed model for DRC hotspot detection (fine-grained task 2)

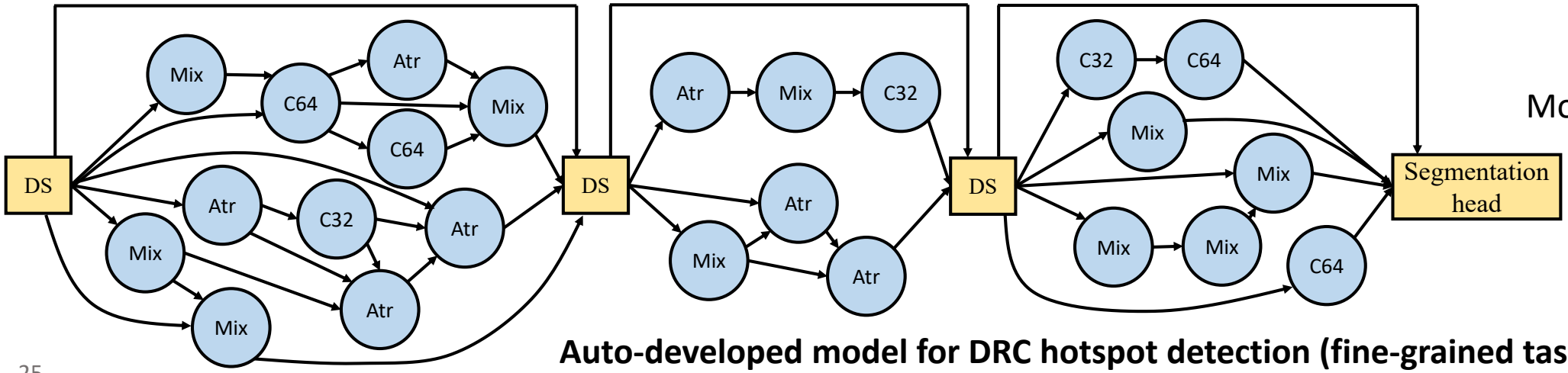
Auto-developed Model Structures

- Auto-developed model for DRC hotspot detection is significantly more complex



Simpler model

Auto-developed model for violated count prediction (coarse-grained task 1)



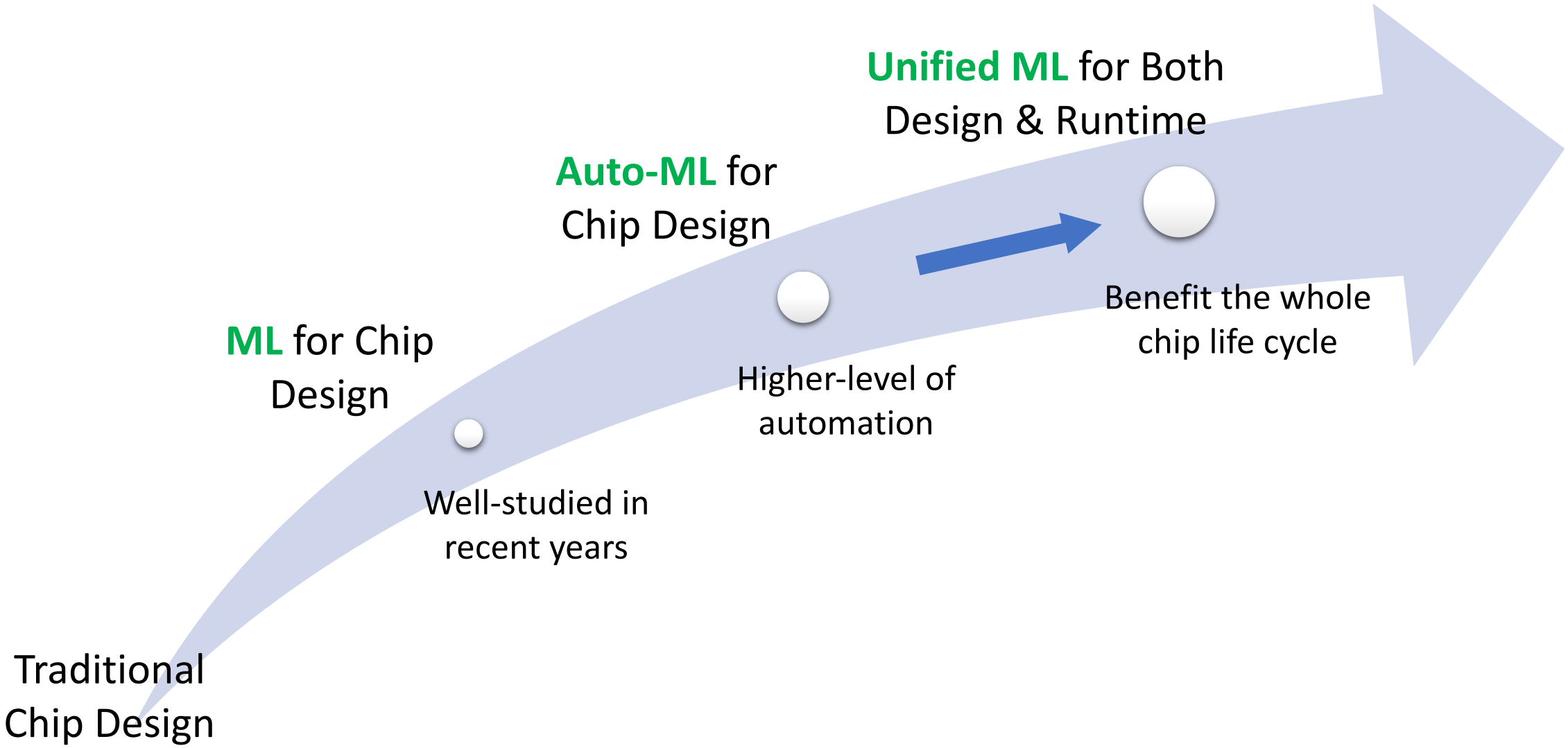
More complex model

Auto-developed model for DRC hotspot detection (fine-grained task 2)

Case Study 2:

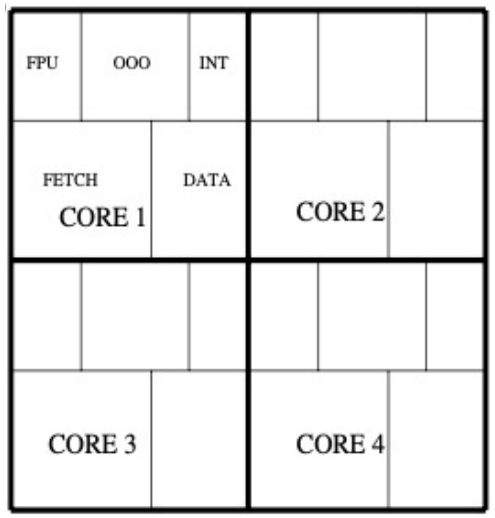
Power & Power Delivery Challenges

What I Believe We Should Target

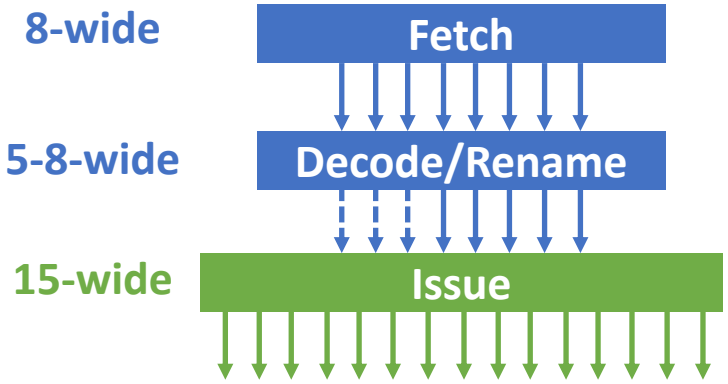


Duke UNIVERSITY

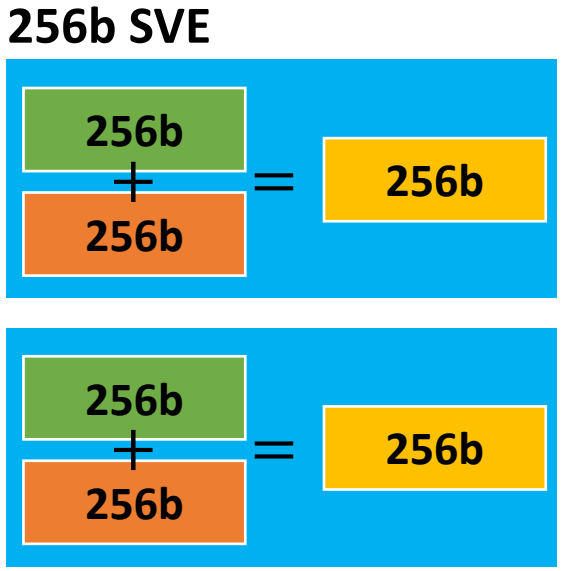
Challenge 1 – Design-time Power Introspection



Many-core CPU with more transistors



Wider issue



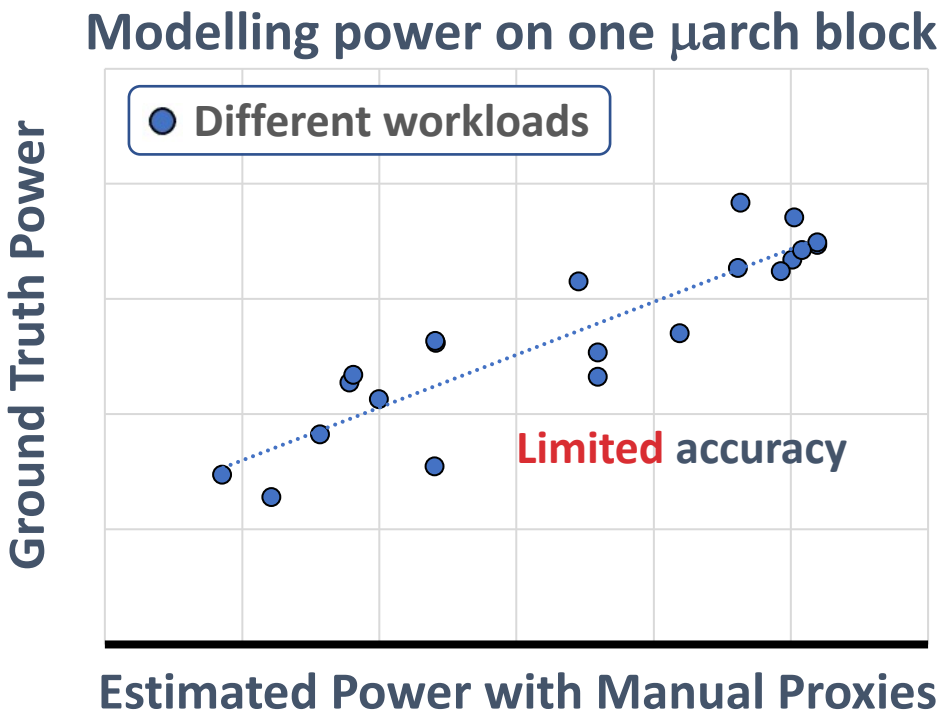
More vectored execution

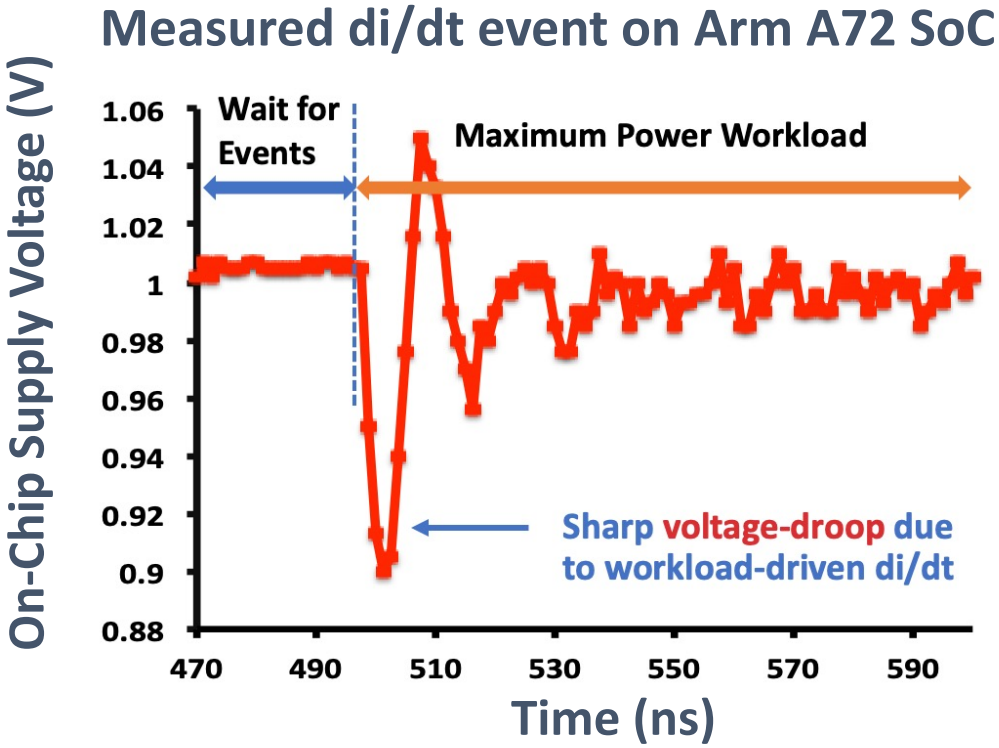
- Delivering generational performance gains **adversely impacts** CPU power
- Power-delivery resources **not keeping pace** with CPU power demands
- **Increasing power-sensitivity drives the need for design-time introspection**

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

Source: Arm Neoverse V1, 2021

Challenge 2 – Run-time Power Introspection

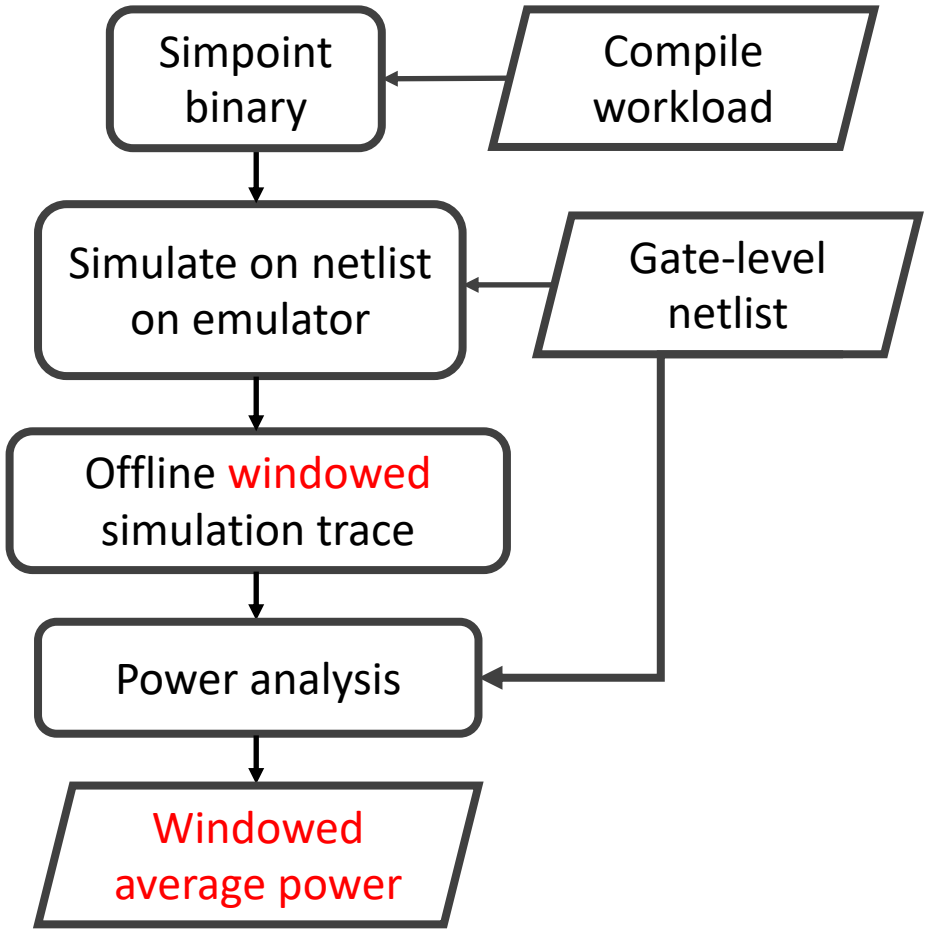




- **Peak-Power mitigation requires accurate power estimation to drive throttling**
  - Manually inferring proxies is very difficult in complex modern CPUs
- **Abrupt changes in CPU current-demand (di/dt event) leading to deep voltage-droop**

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

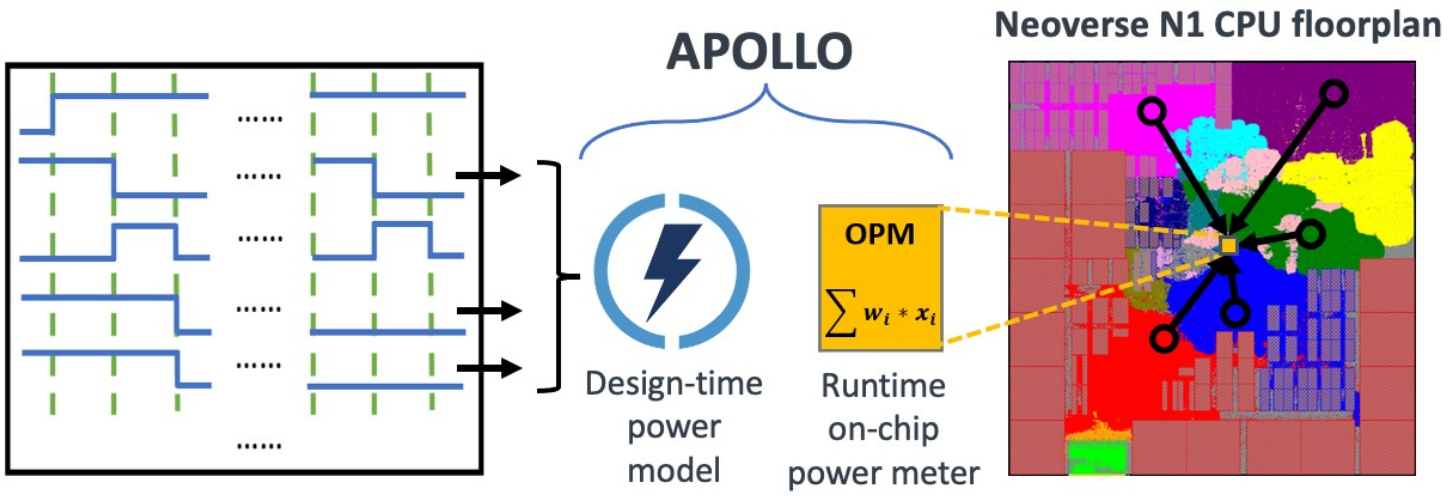
- Simulation on realistic workloads
  - **Expensive** and **slow**
  - **Limited** temporal-resolution

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**

Duke UNIVERSITY

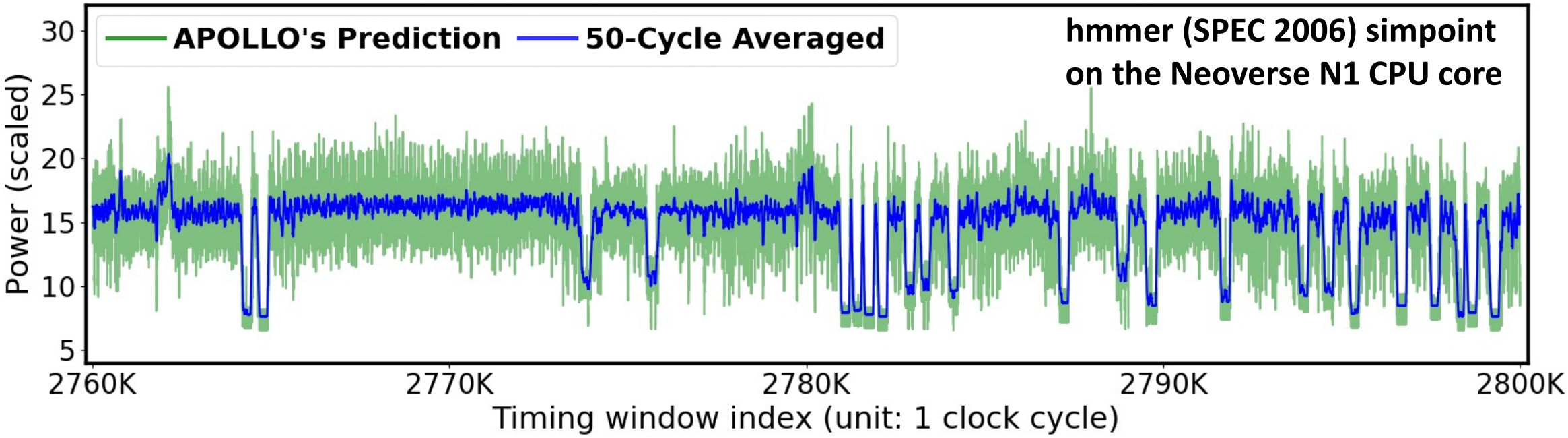
APOLLO: A Unified Power Modeling Framework



- **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

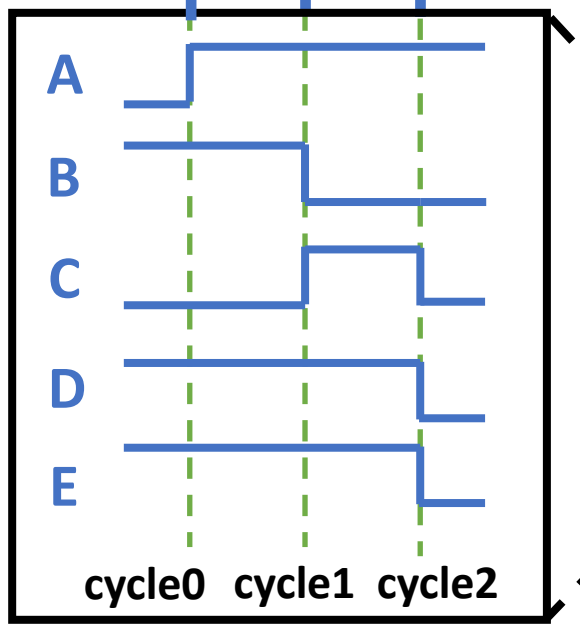
40K cycles of APOLLO Power Estimation out of a trace of 17M cycles



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

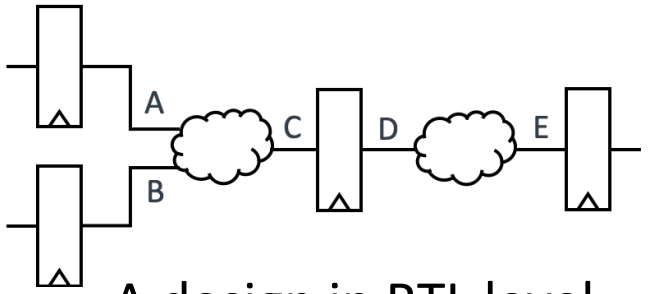
Duke UNIVERSITY

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: $F(X) = y$

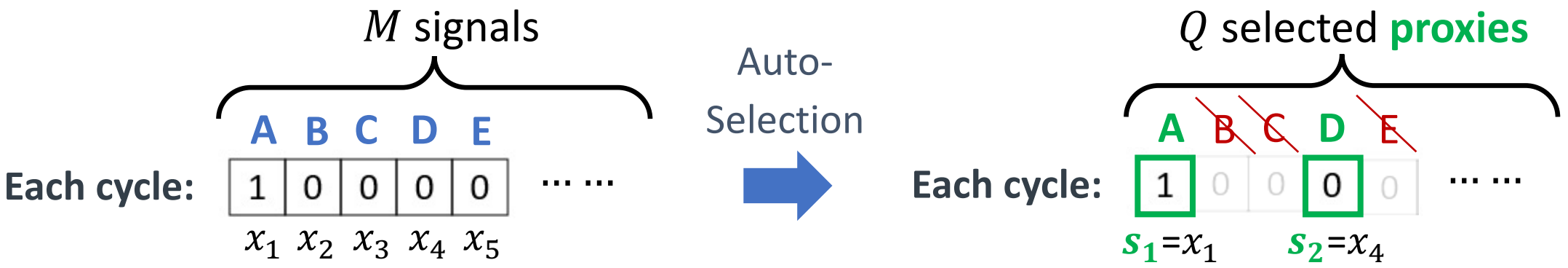
M > 500,000 in Neoverse N1

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

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

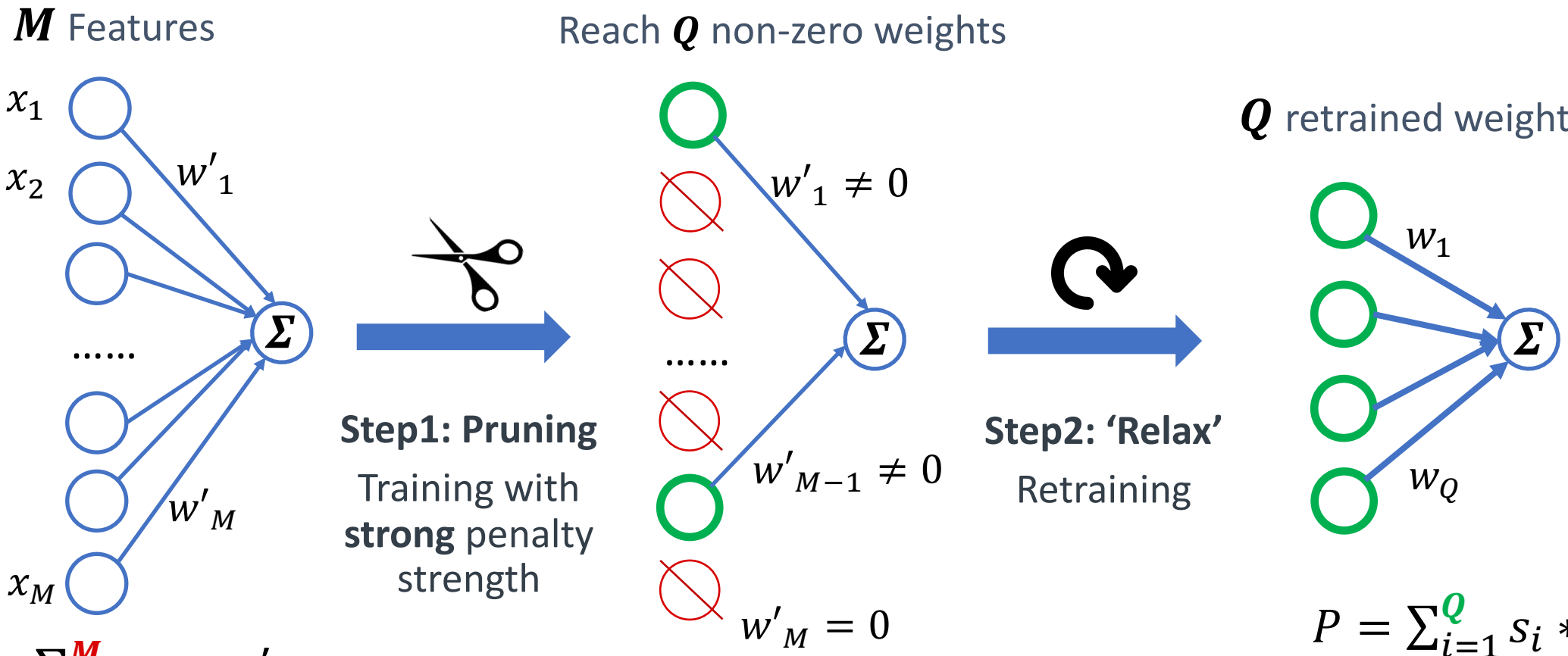
Linear model with Q selected proxies

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

Duke UNIVERSITY

ML-Based Power Proxies Selection

Model construction in two steps

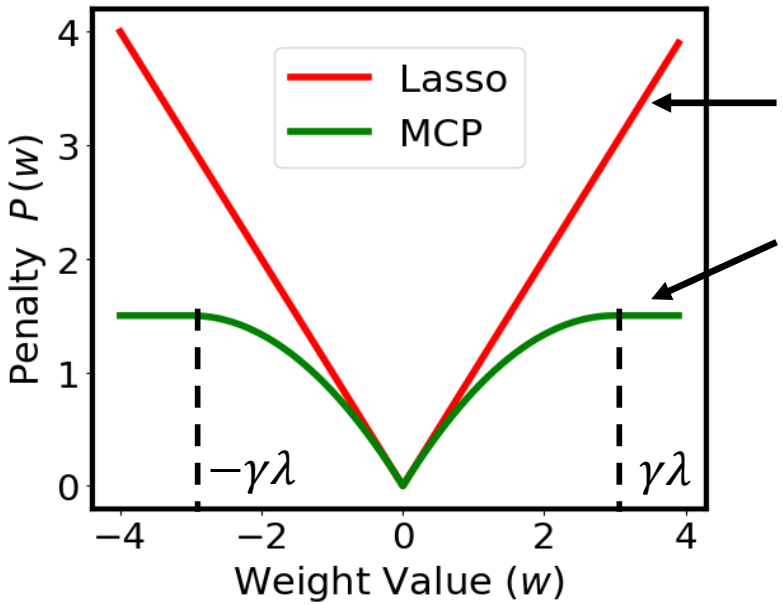


Minimax concave penalty (**MCP**) for pruning

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

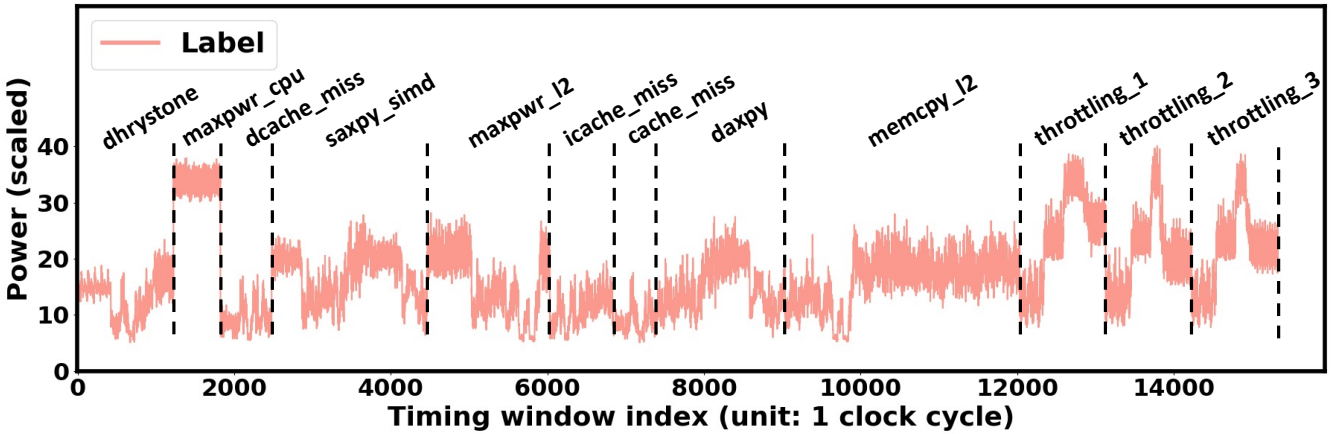
MCP protects large weights

Duke UNIVERSITY

Model Training and Testing

Neoverse N1 (infra)
Deployed in AWS Graviton

Cortex A77 (mobile)
Deployed In Snapdragon 865

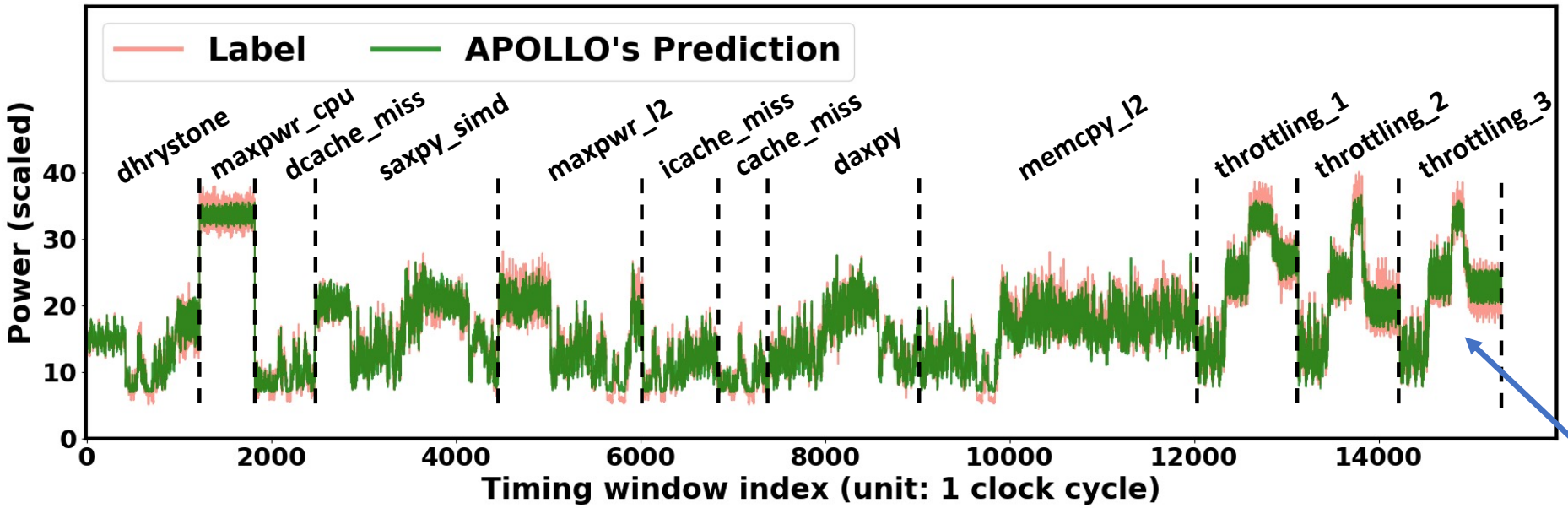


- Experiments on 3GHz 7nm Arm **commercial** microprocessors **Neoverse N1** and **Cortex A77**
- **Automatically** generate a “diverse” set of random micro-benchmarks for training
- Testing on **various** Arm power-indicative workloads

Duke UNIVERSITY

Prediction Accuracy as Design-Time Power Model

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



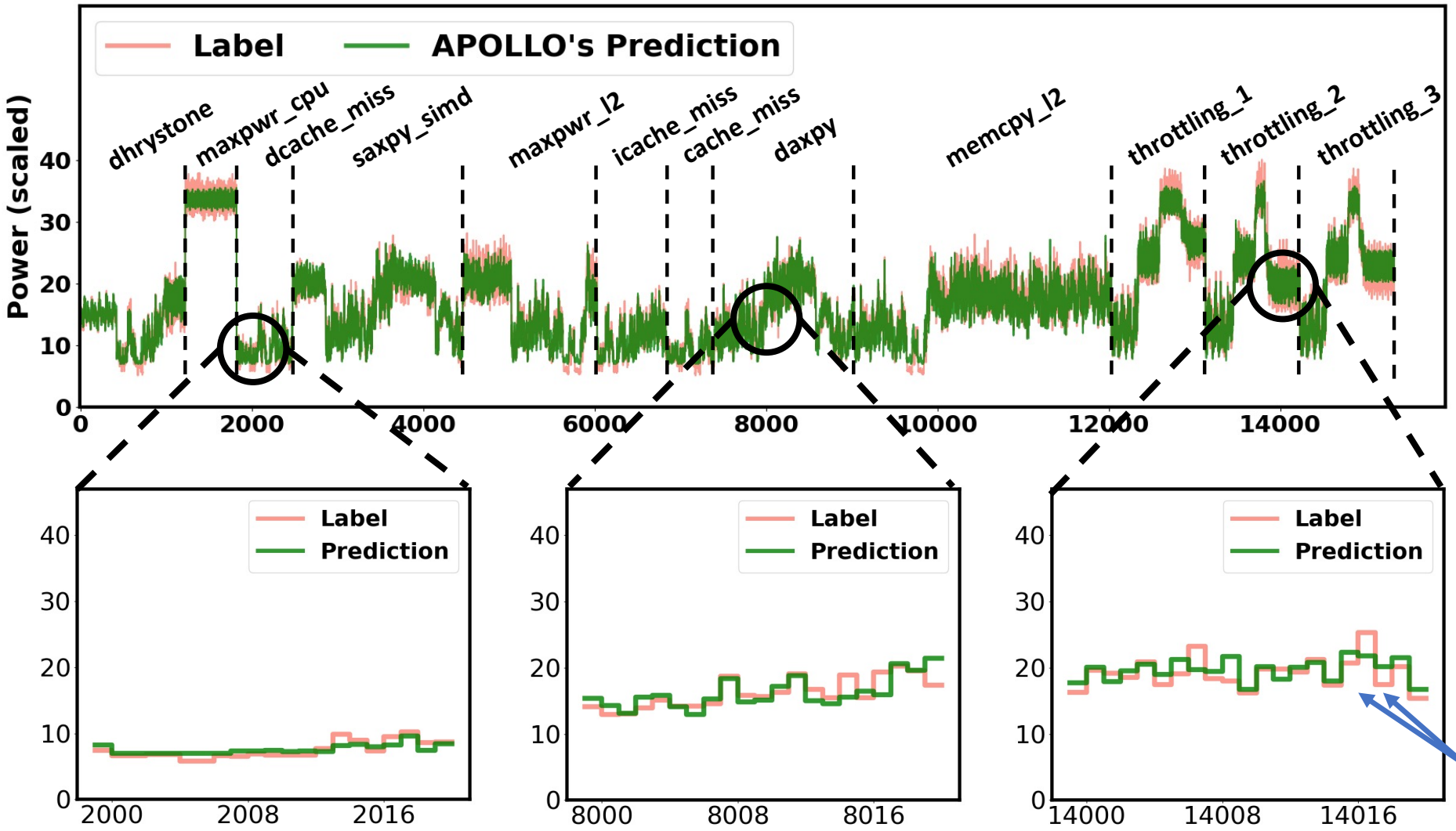
- MAE = 7.19%
- R^2 = 0.953

Prediction trace shows great agreement with ground-truth

Duke UNIVERSITY

Prediction Accuracy as Design-Time Power Model

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

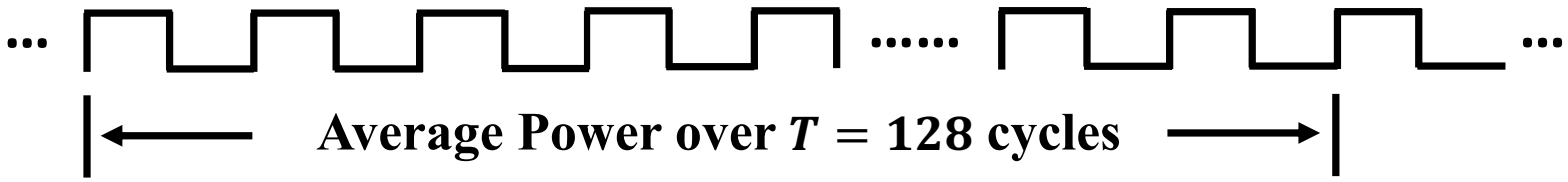


- MAE = 7.19%
- R^2 = 0.953

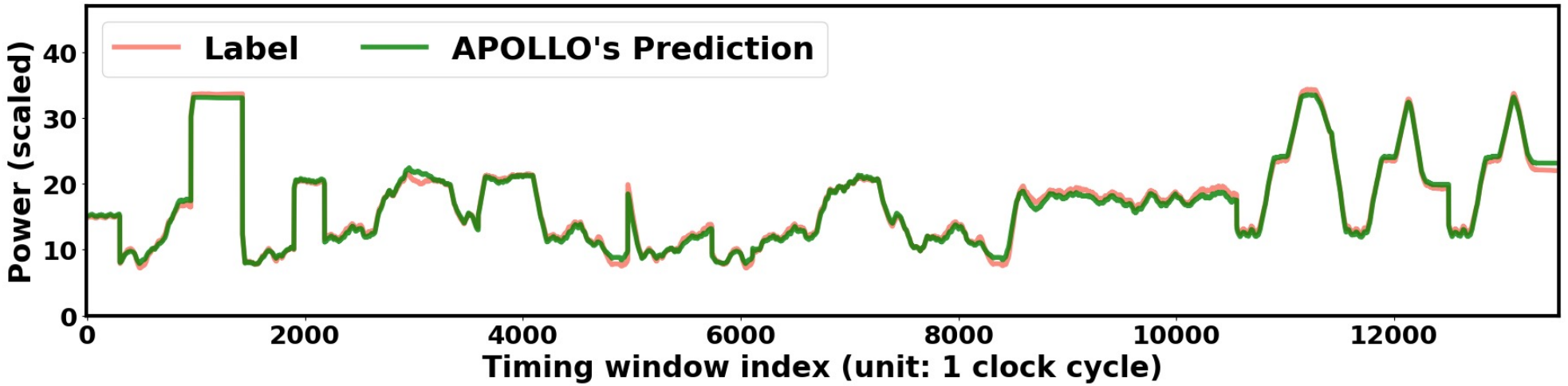
Per-cycle error can be averaged

Duke UNIVERSITY

Accuracy on Multi-Cycle Power Estimation



128-cycle prediction from APOLLO with Q=70 proxies

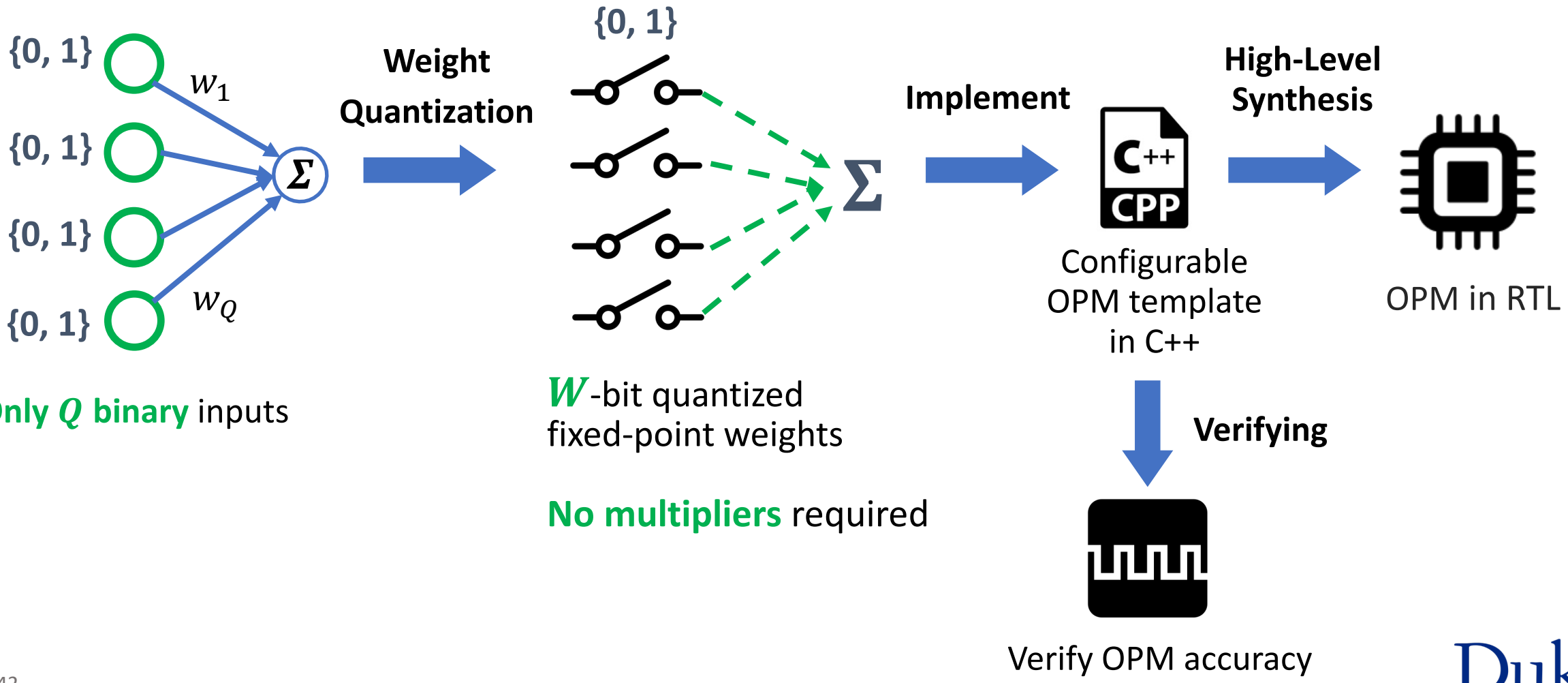


- MAE = 2.82%
- R^2 = 0.993
- Higher accuracy

Duke UNIVERSITY

Automated Low-Cost Runtime OPM Implementation

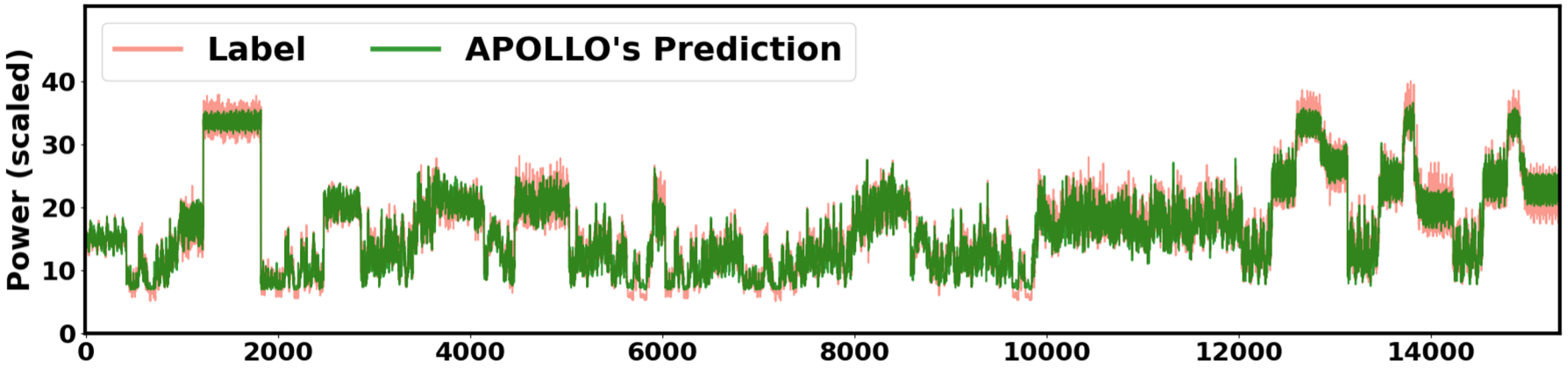
APOLLO is designed to be hardware-friendly



Duke UNIVERSITY

Prediction Accuracy from Design-time Model & OPM

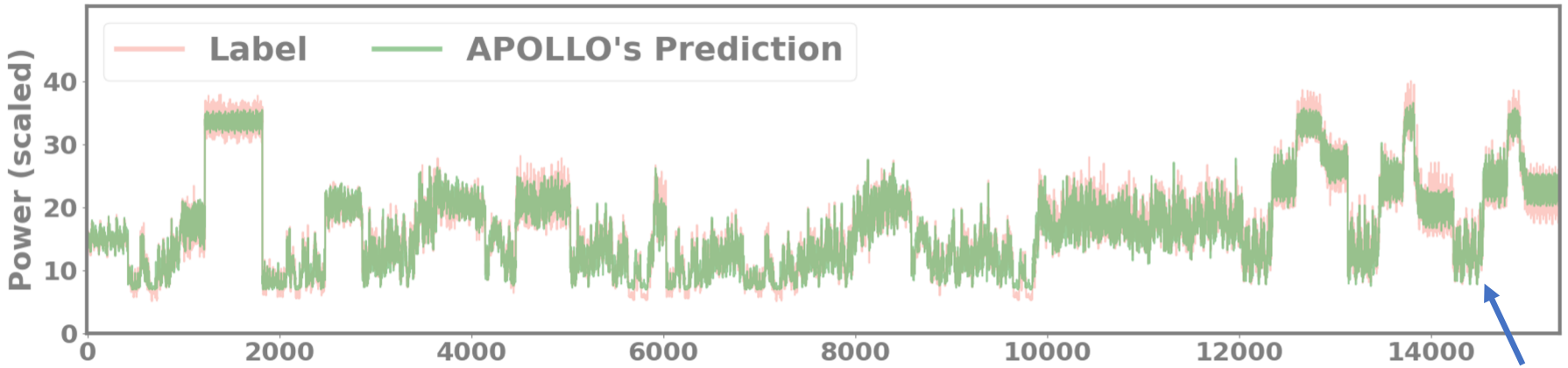
Per-cycle prediction from APOLLO with $\boldsymbol{Q}$=159 proxies



- MAE = 7.19%
- R^2 = 0.953

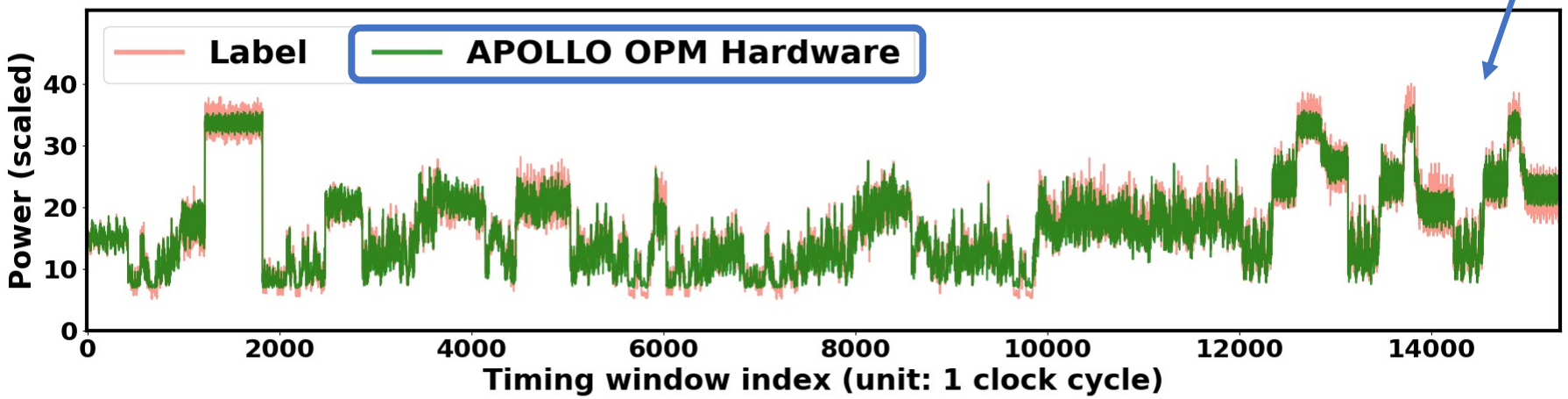
Prediction Accuracy from Design-time Model & OPM

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



Negligible difference

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



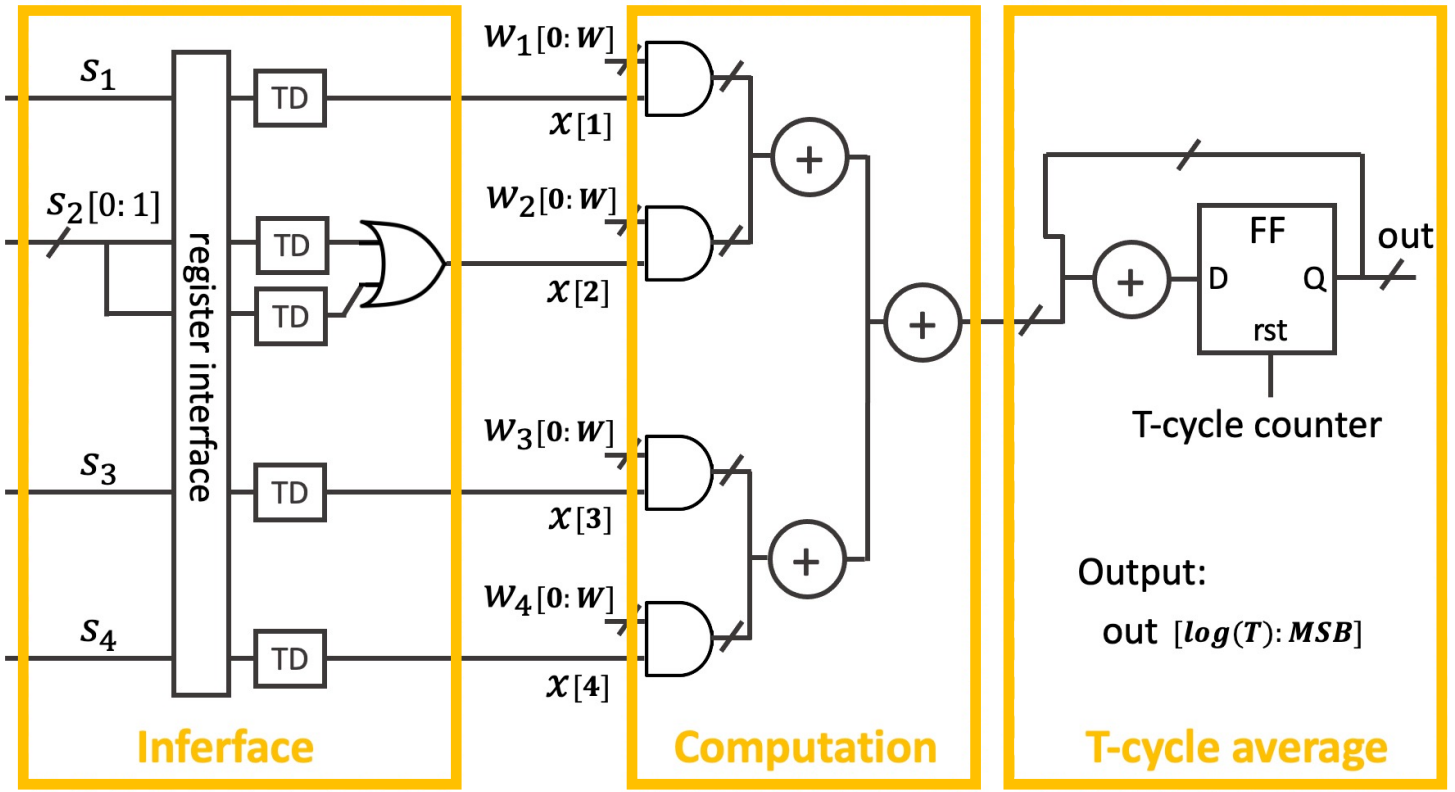
- MAE = 7.19%
- R^2 = 0.953

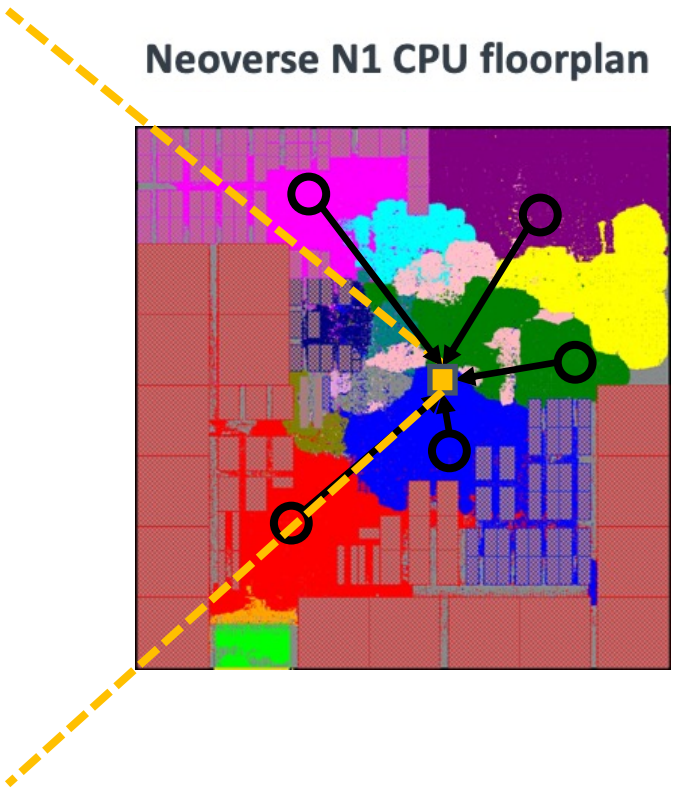
Negligible difference

- MAE = 7.19%
- R^2 = 0.953

- **W=11 bits after quantization**

Overview of the OPM Hardware Design





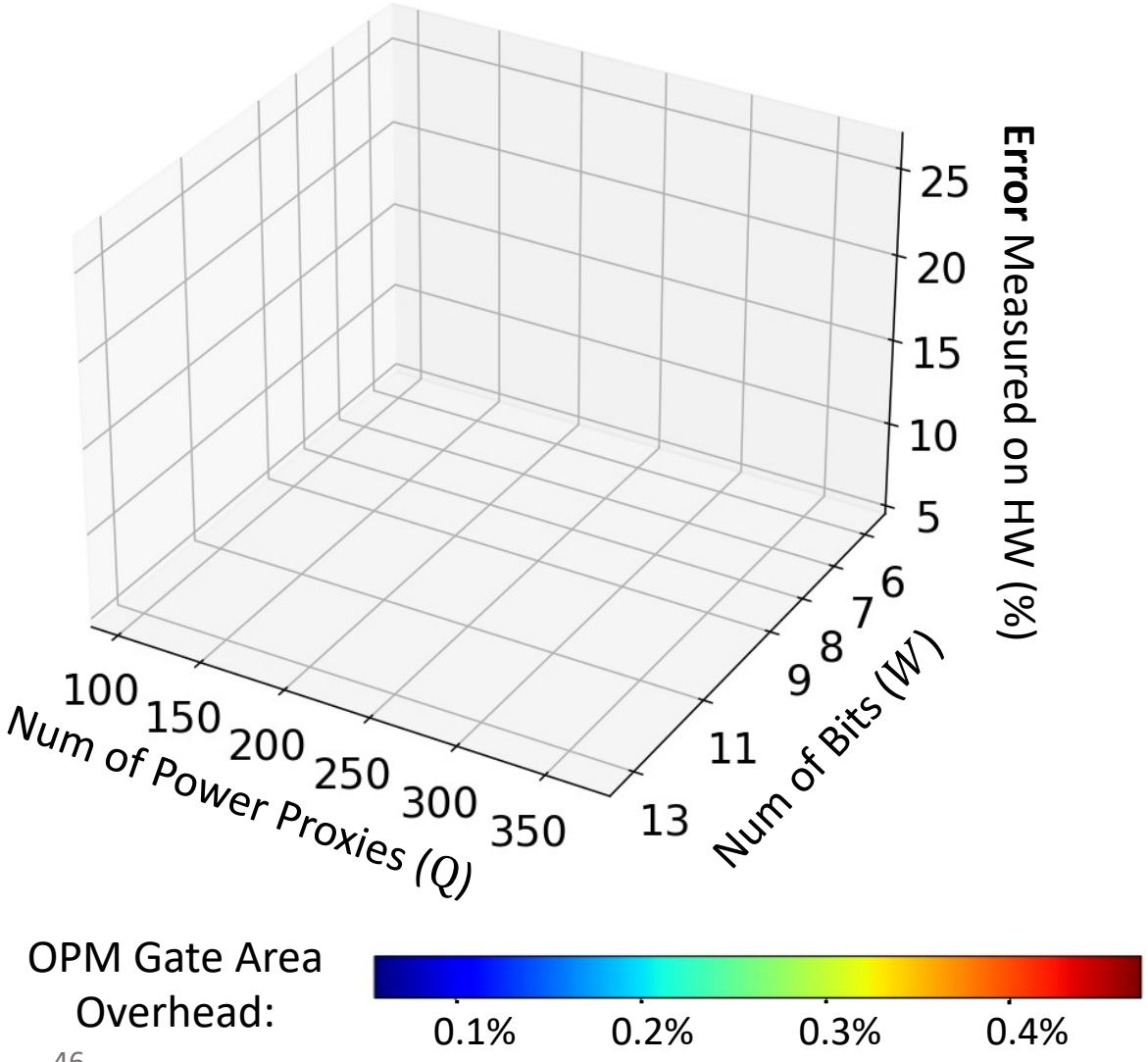
TD 1-bit toggle detector with one XOR and FF

FF Flip-flop

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

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

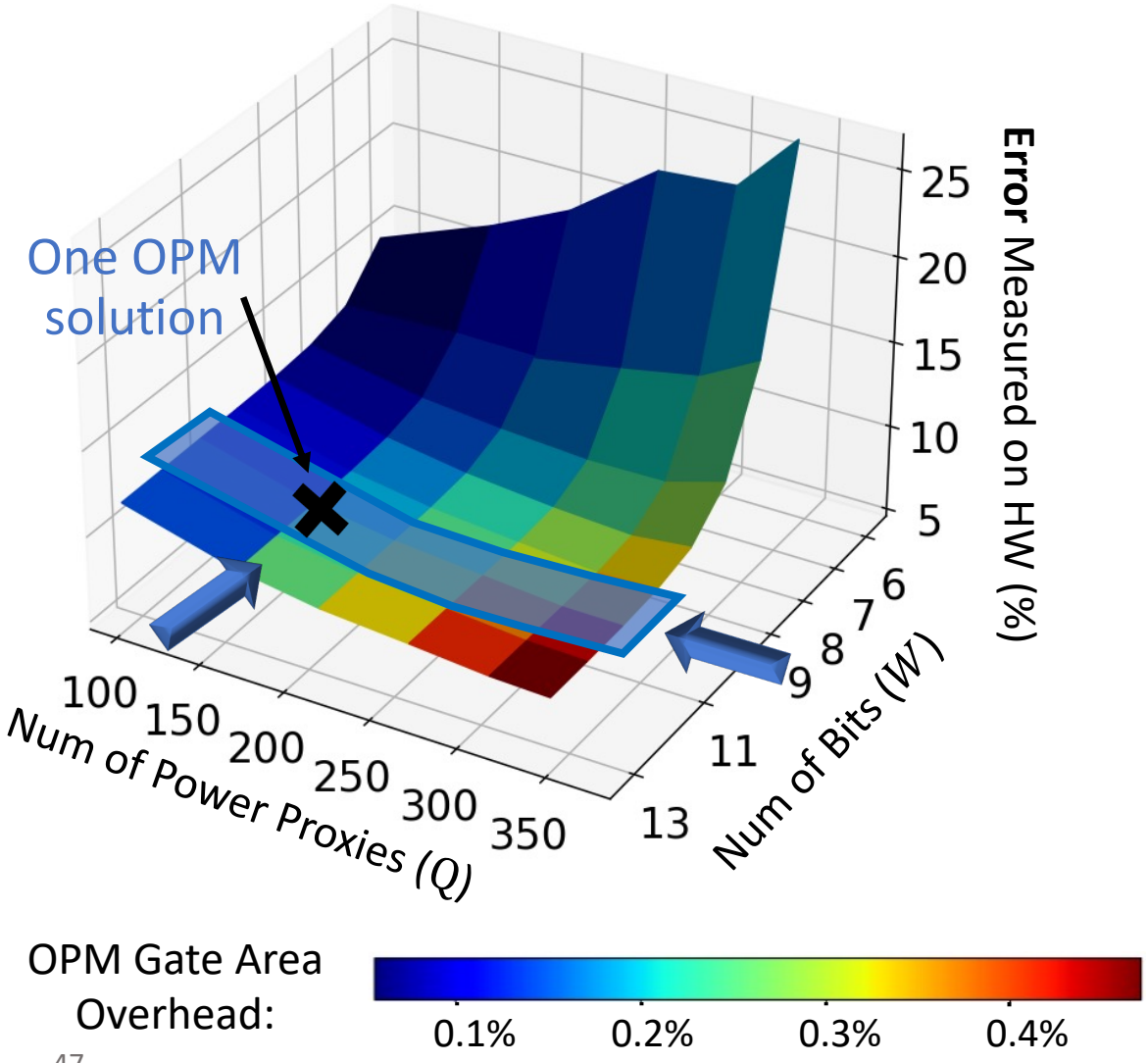
Runtime OPM implementation on Neoverse N1



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

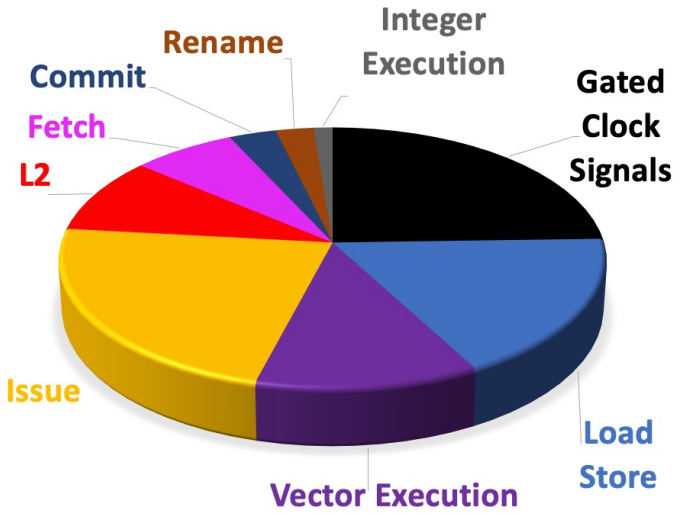
Accuracy vs. Hardware Cost (Area Overhead) of the OPM

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

Potential Application: Design-time Power Introspection



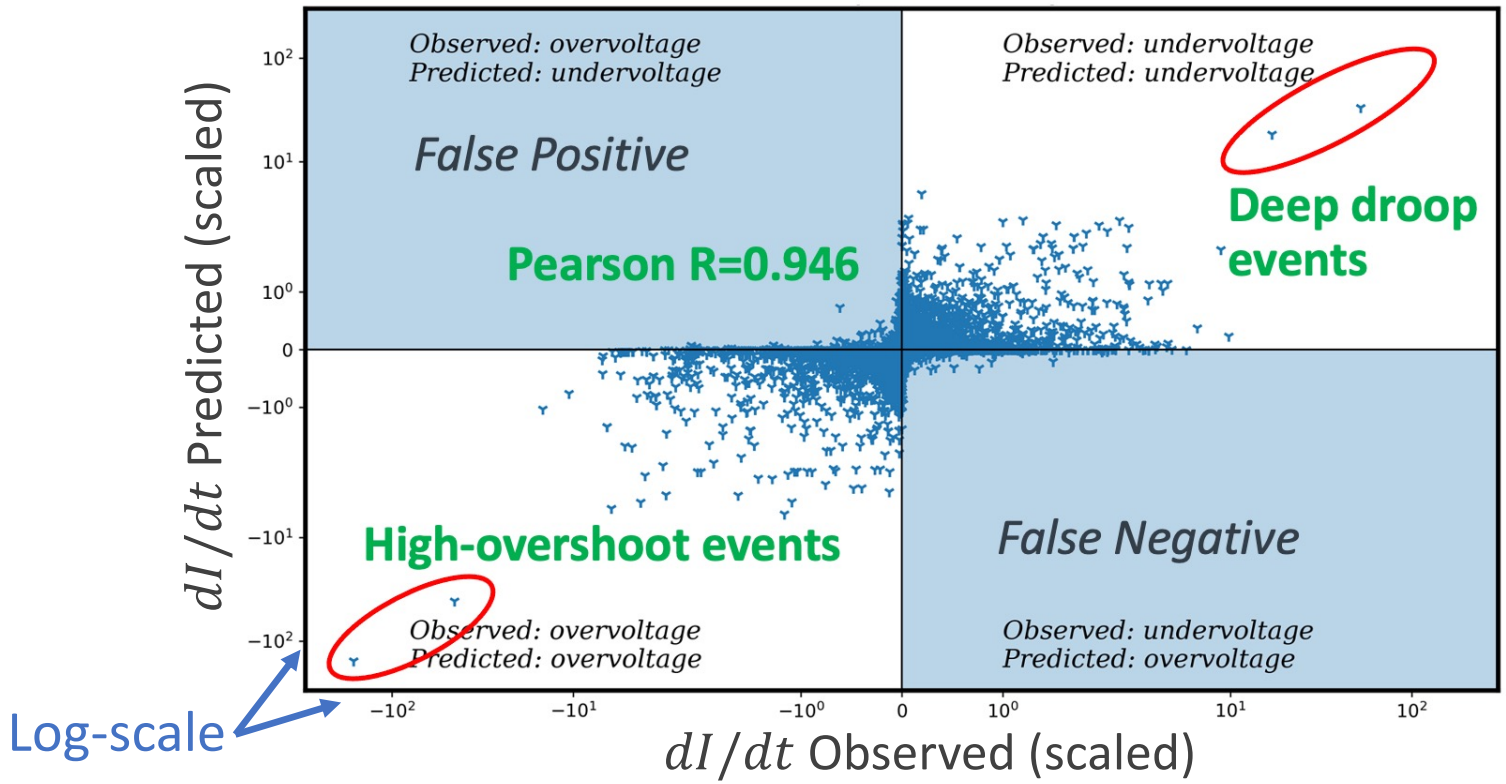
Distribution of power proxies on Neoverse N1

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

Better **interpretability**

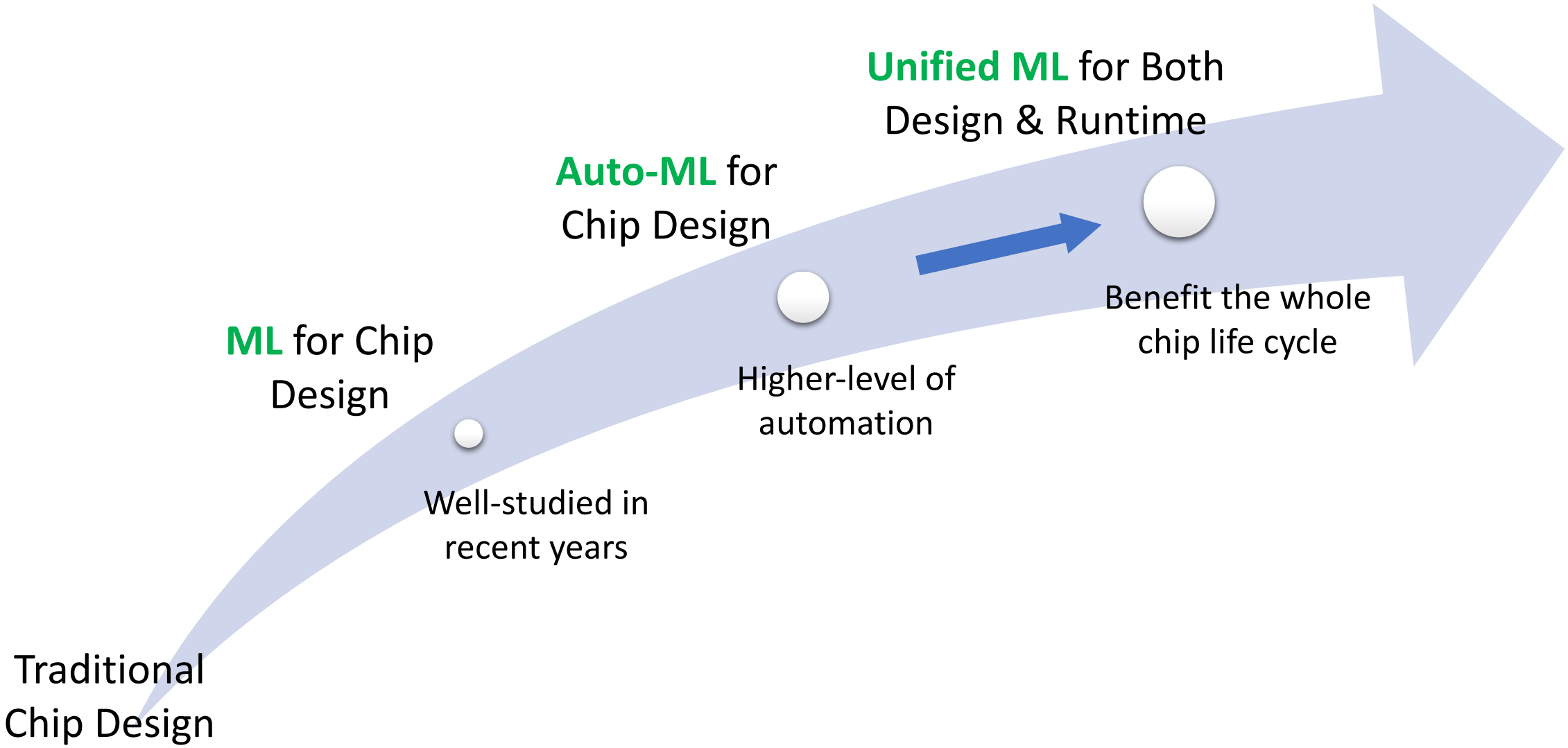
Identify power contributors for designers!

Potential Application: Runtime dI/dt Mitigation



Enable CPU-driven Proactive dI/dt Mitigation!

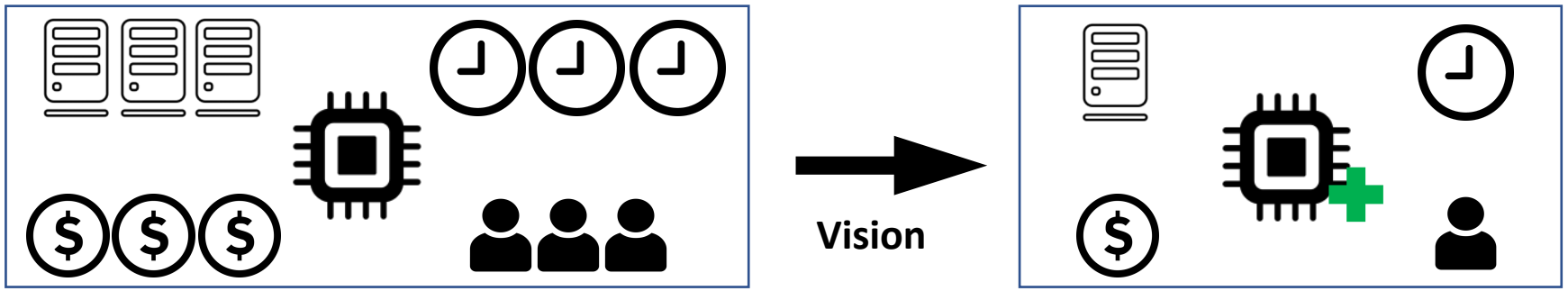
What I Believe We Should Target



Duke UNIVERSITY

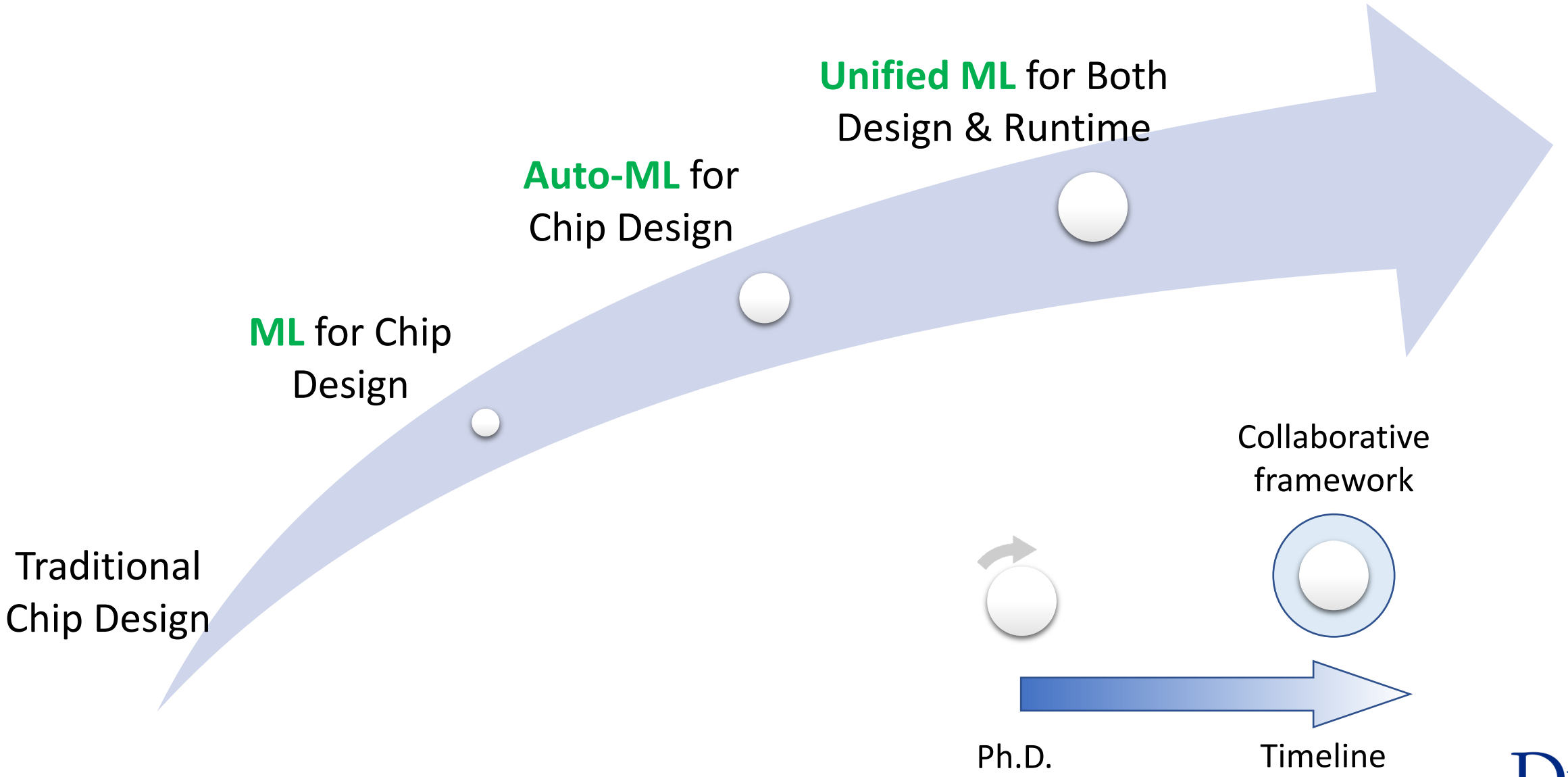
Summary and Takeaway

- Problem: Increasing Challenges in Chip Design
  - Cost, time-to-market, reliance on designers, diminishing performance return, ……
- **ML** in chip design
  - Less simulation time, faster feedback, less designer effort
- **AutoML** in chip design
  - Reduces months of model development to hours, no developers
- **Unified ML** in both design & runtime
  - Benefit the entire chip life cycle



Duke UNIVERSITY

Future Research Plan

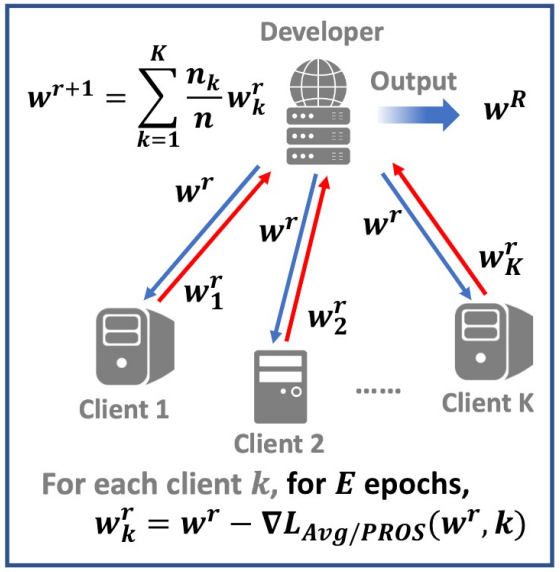


Duke UNIVERSITY

Future Works: Collaborative Framework

Collaborative ML in Chip Design

- Model quality depends on data
- Circuit data from different companies
- Design data is highly confidential



Federated Learning:

Train on local data

Communicate weights

Example – Collaborative Training

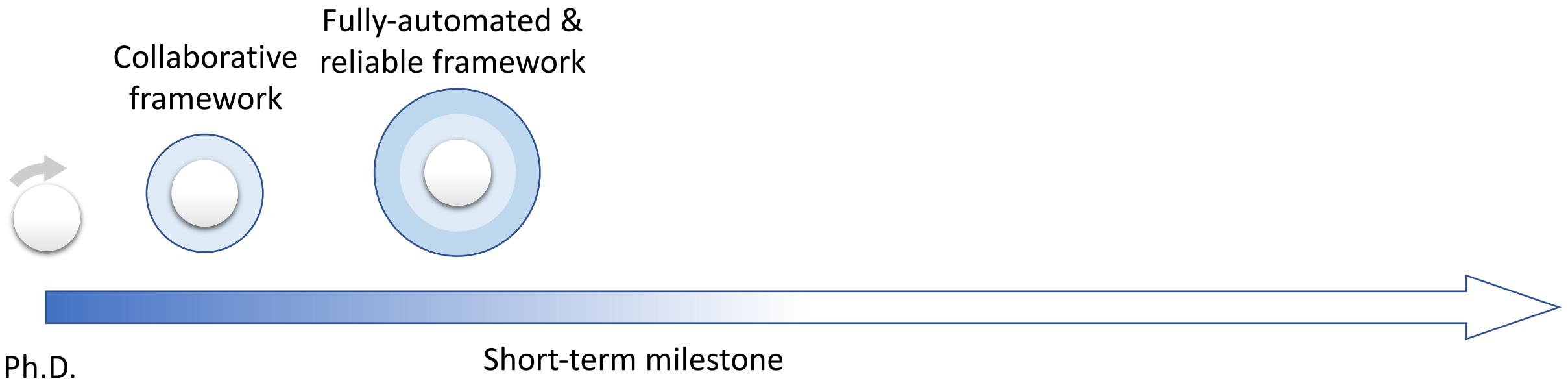
		Test on 9 Clients (C1 to C9)									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	**Avg**
Train on 9 Clients	C1	0.68	0.59	0.59	0.58	0.58	0.56	0.65	0.60	0.52	0.59
	C2	0.49	0.52	0.50	0.51	0.52	0.50	0.53	0.52	0.37	0.50
	C3	0.55	0.56	0.55	0.50	0.52	0.46	0.57	0.57	0.49	0.53
	C4	0.52	0.49	0.51	0.53	0.51	0.53	0.52	0.52	0.46	0.51
	C5	0.71	0.53	0.59	0.55	0.55	0.61	0.60	0.47	0.80	0.60
	C6	0.71	0.51	0.57	0.51	0.52	0.58	0.68	0.60	0.78	0.61
	C7	0.73	0.54	0.62	0.56	0.47	0.52	0.72	0.61	0.72	0.61
	C8	0.76	0.60	0.65	0.60	0.55	0.55	0.71	0.64	0.57	0.63
	C9	0.73	0.54	0.65	0.59	0.50	0.61	0.73	0.61	0.91	0.65
Train & Test Same Client		0.68	0.52	0.55	0.53	0.55	0.58	0.72	0.64	0.91	**0.63**
FedProx		0.63	0.83	0.71	0.72	0.66	0.67	0.63	0.57	0.42	**0.65**
FedProx + Finetuning		0.83	0.86	0.76	0.75	0.74	0.75	0.81	0.72	0.90	**0.79**

One same model in a row Nine different models in a row

- Assuming data distributed to 9 clients (C1 to C9)

Duke UNIVERSITY

Future Research Plan

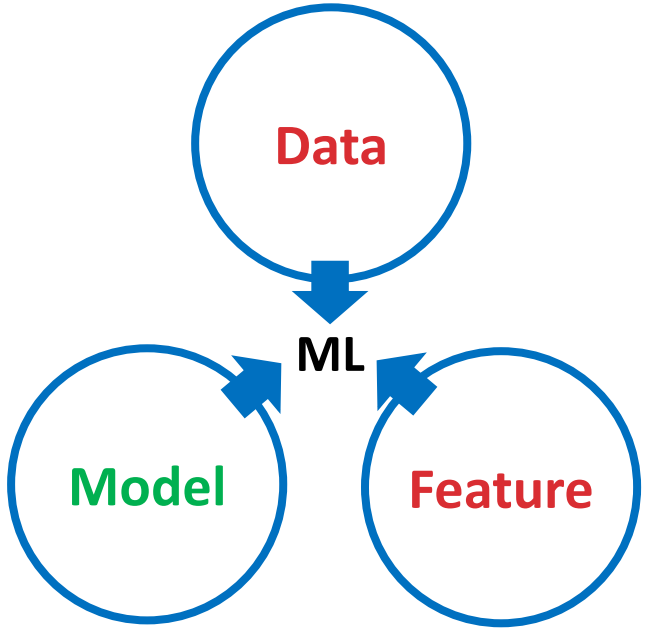


Duke UNIVERSITY

Future Works: Fully-Automated & Reliable Framework

Fully-Auto ML in Chip Design

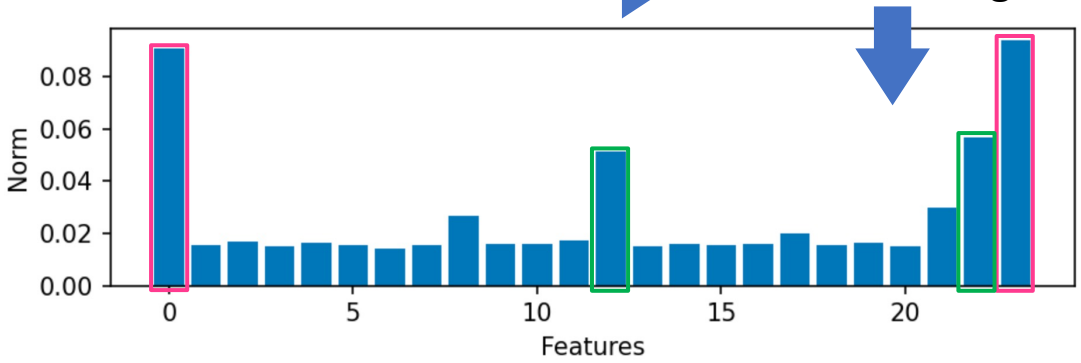
- Automated feature selection
- Automated data selection
- Automated data augmentation



Example – Feature Selection

- The first conv layer:

32 kernels with size 3x3x24 → 24 feature weight vectors



# Features	ROC-AUC
2 ( )	**0.866**
4 (+)	**0.867**
24 (all)	0.867

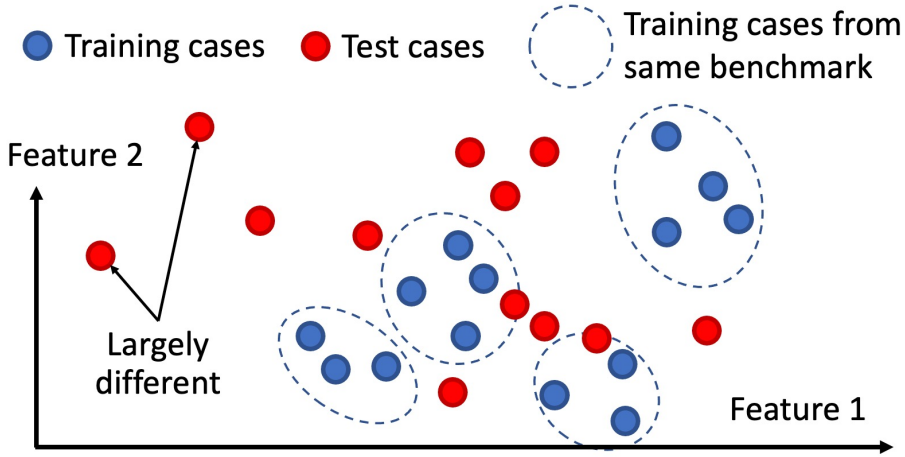
Selected Features:

0. Pin density
12. MST fly lines
22. DFF cell density
23. Clock tree cell density

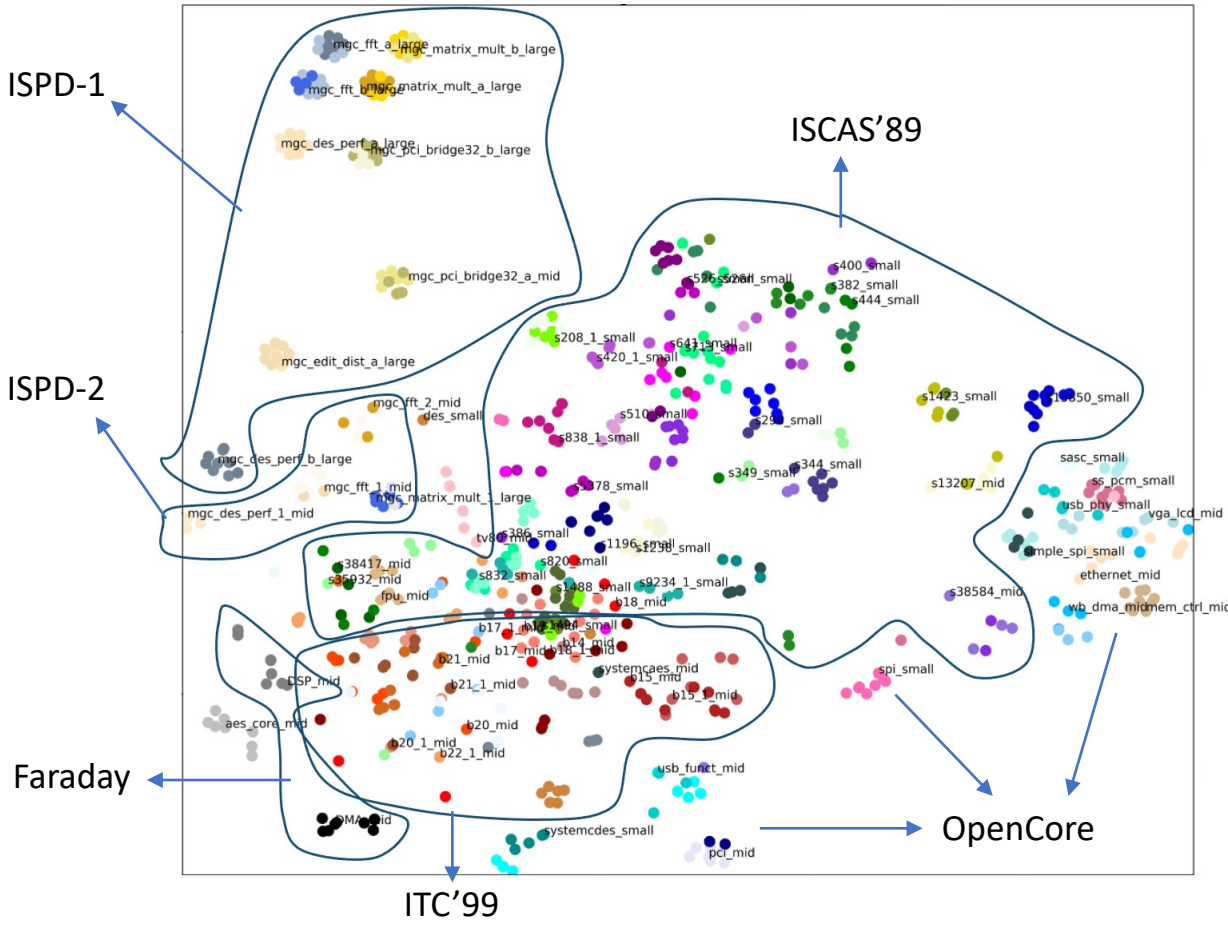
Future Works: Fully-Automated & Reliable Framework

Reliable ML in Chip Design

- Designs very sparsely distributed
- Almost impossible to perform well on every test case
- How can we trust each prediction?



Example – Design Difference



Future Research Plan

Need knowledge on ML, circuits, software, etc.

Need knowledge on optimization, computer architecture, etc.

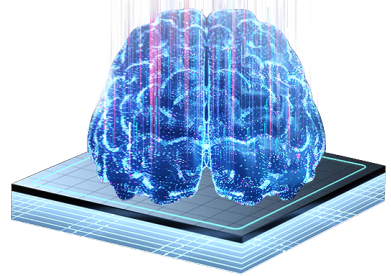
Collaborative framework

Fully-automated & reliable framework

Multi-domain/objective, efficient optimization

Comprehensive framework from system-level to testing

Accommodates emerging tech



True intelligence in chip design

Ph.D.

Short-term milestone

Longer-term milestone

Duke UNIVERSITY

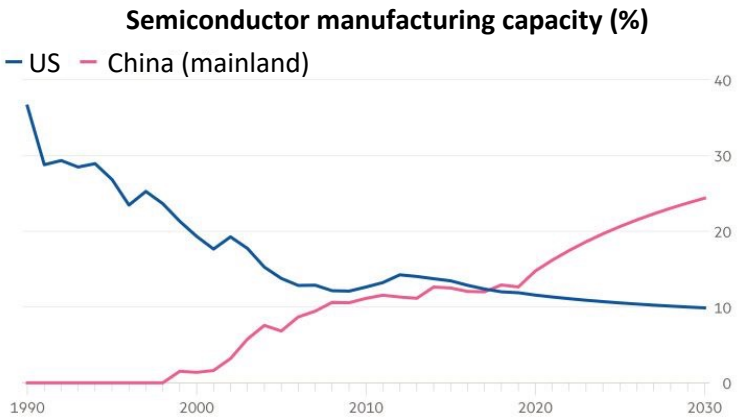
Future Funding and Collaboration Opportunities

- Agencies:
  - General Research Fund (GRF), Early Career Scheme, NSFC, ITF
- US companies:
  - Cadence, Synopsys, Nvidia, Arm, NXP
- Chinese companies:
  - Huawei, Alibaba T-head, Chinese EDA start-ups like UniVista

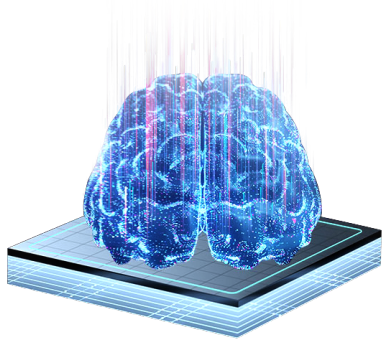
Congressional Research Service
Informing the legislative debate since 1914

China's New Semiconductor Policies: Issues for Congress

US restricts software exports to Chinese chip companies



Semiconductor switching to Asia, including 'Greater Bay Area'



A great chance to overtake leading EDA companies

Previous Collaborations and Grant Writing Experiences

- Many thanks for my advisors and collaborators:

Prof. **Yiran Chen** Duke University	Prof. **Hai "Helen" Li** Duke	Prof. **Jiang Hu** TAMU	Dr. **Brucek Khailany** Nvidia	Dr. **Haoxing Ren** Nvidia
Dr. **Shidhartha Das** Arm	Dr. **Xiaoqing Xu** Arm	Dr. **Brian Cline** Arm	Dr. **Chand Kashyap** Cadence	Dr. **Aiqun Cao** Synopsys

- My previous grant writing experiences (funded):
  - **NSF**: Revitalizing EDA from a Machine Learning Perspective
  - **SRC**: A Machine Learning Approach for Cross-Level Optimizations
  - **SRC**: A Collaborative Machine Learning Approach to Fast and High-Fidelity Design Prediction
  - **Industry (Cadence)**: NAS-based Fully Automatic ML Estimator Development Flow in EDA
  - **Industry (Cadence)**: A Machine-Learning based Pre-placement Wirelength Estimator

Courses I am Qualified to Teach

- Circuit and Computer Engineering Courses
  - Digital VLSI design, digital integrated circuits (**CAD admin at Duke**)
  - Chip design methodologies
  - Digital logic & systems (**TA of undergraduate course at Duke**)
  - Computer organization and architecture
- Machine Learning Courses
  - Linear algebra for engineering (**TA of graduate course at Duke**)
  - Data mining, artificial intelligence, machine learning
  - Computer vision, deep learning

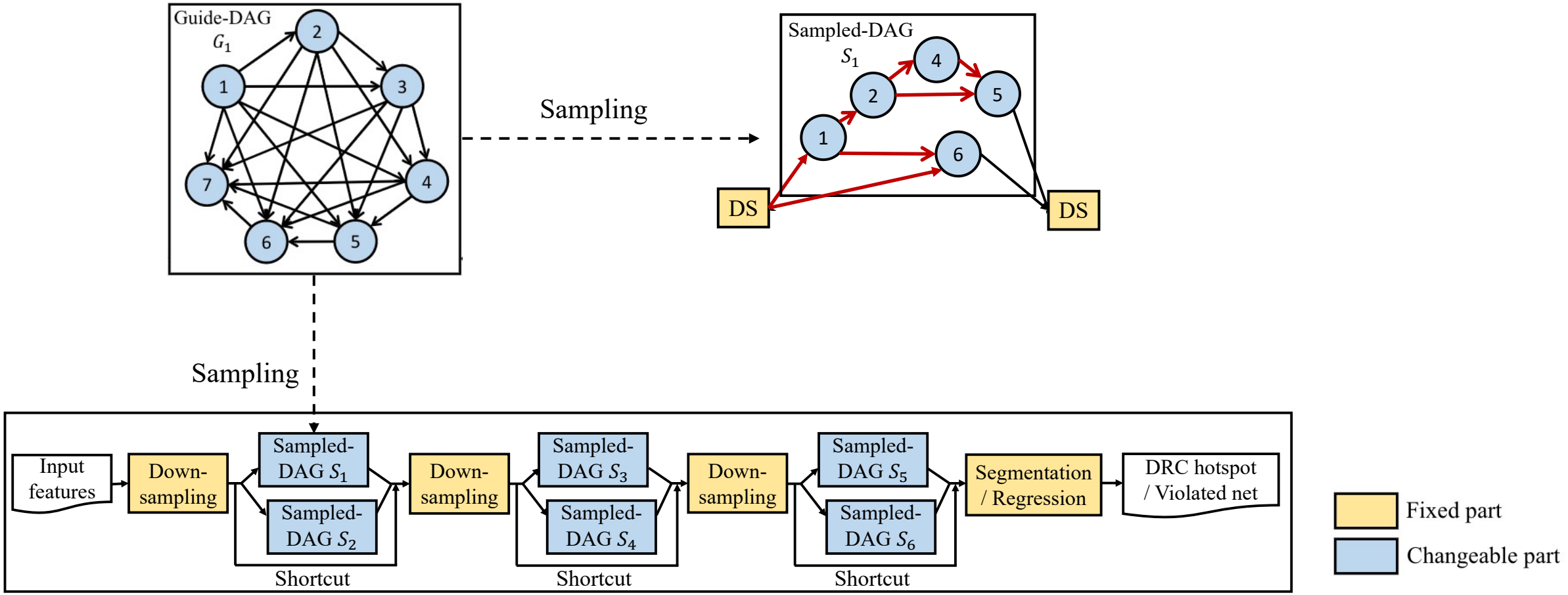
Thanks! Questions?

If you have further questions, please contact me:

zhiyao.xie@duke.edu

Duke

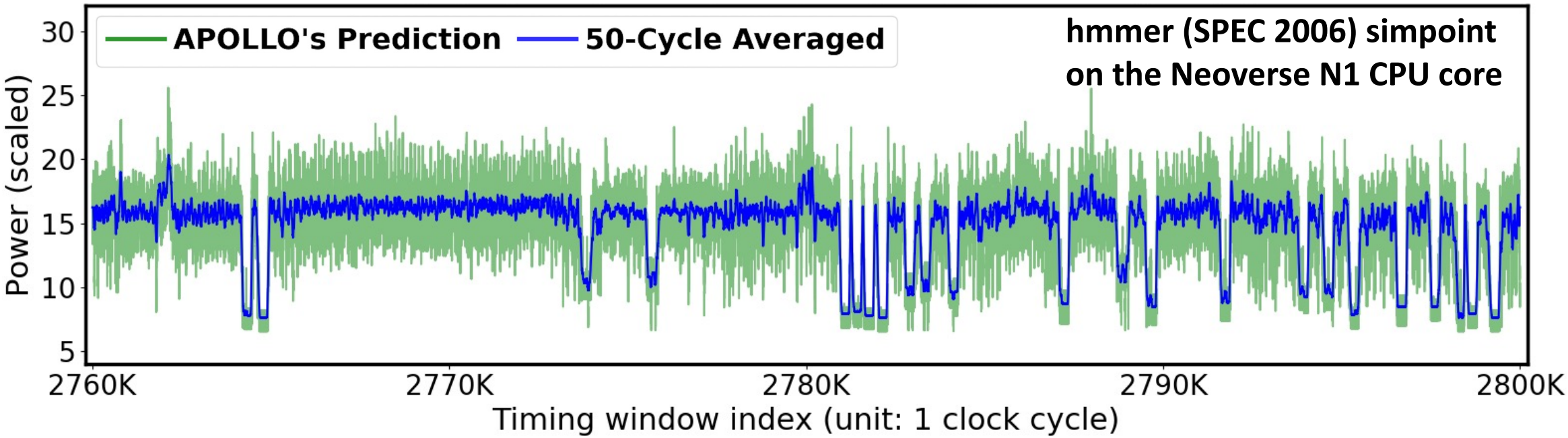
Automatic Estimator Development – Search Space



[Chang, et al., ICCAD'21]

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 (90-95% accuracy)**
- Unprecedented **single-cycle** temporal resolution

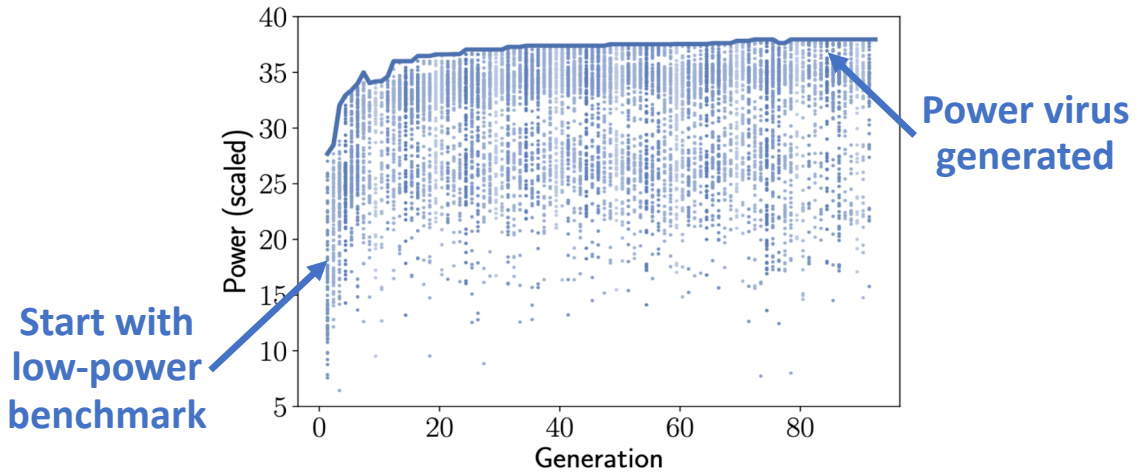
Duke UNIVERSITY

Our Proposed Power Modeling Approach

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

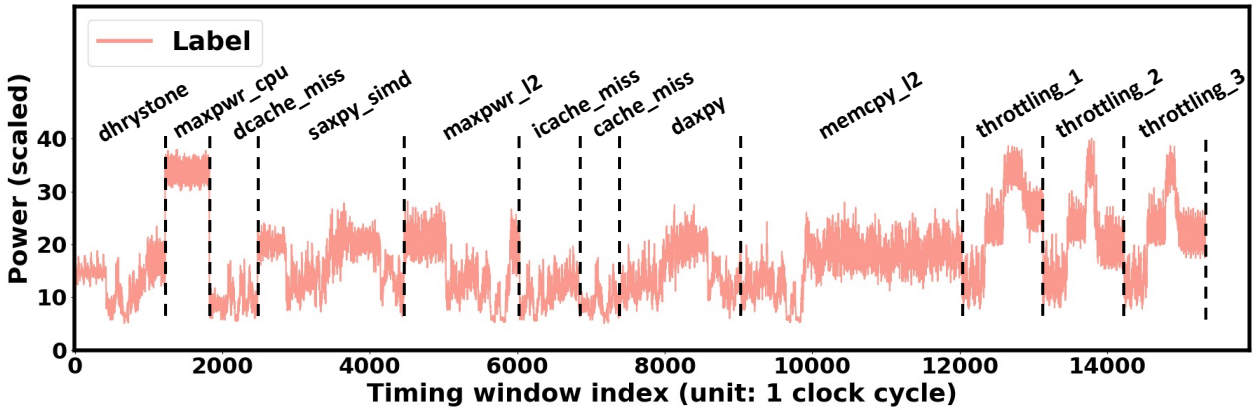
Training data automatically generated

- 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

Summary and Conclusions

- Fast power-modelling has a material impact in how we design and deploy CPUs
- Micro-architecture agnostic methodology is automated and can scale to multiple compute-solutions – CPUs, GPUs, NPUs and even for sub-blocks
- Potential applications extend from power/thermal management in many-core SoCs to CPU-driven proactive droop-mitigation
- ML/Data-Science approaches are potential disruptors to many aspects of design

Duke
UNIVERSITY

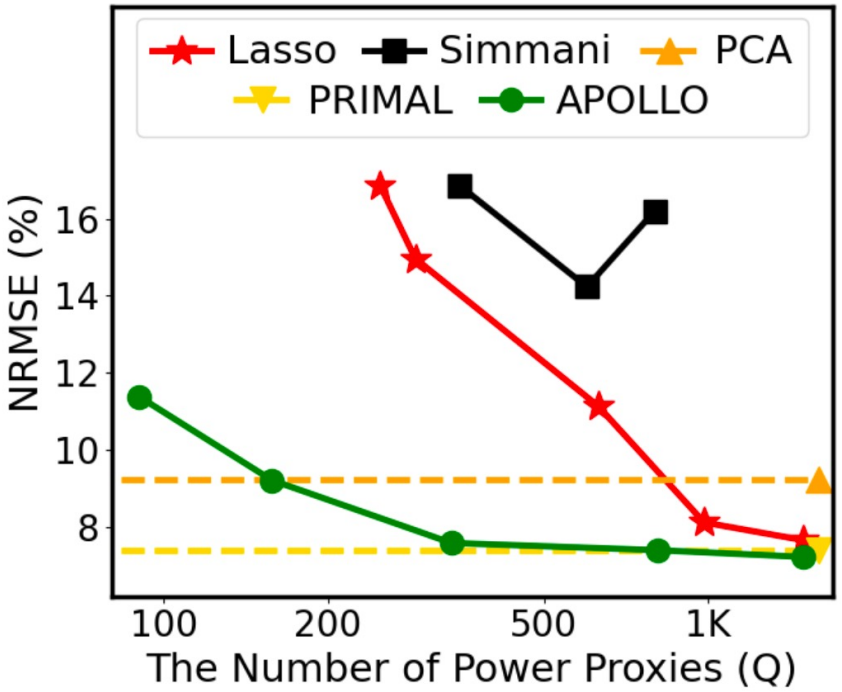
FAQ: Prior Power Modeling Works

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

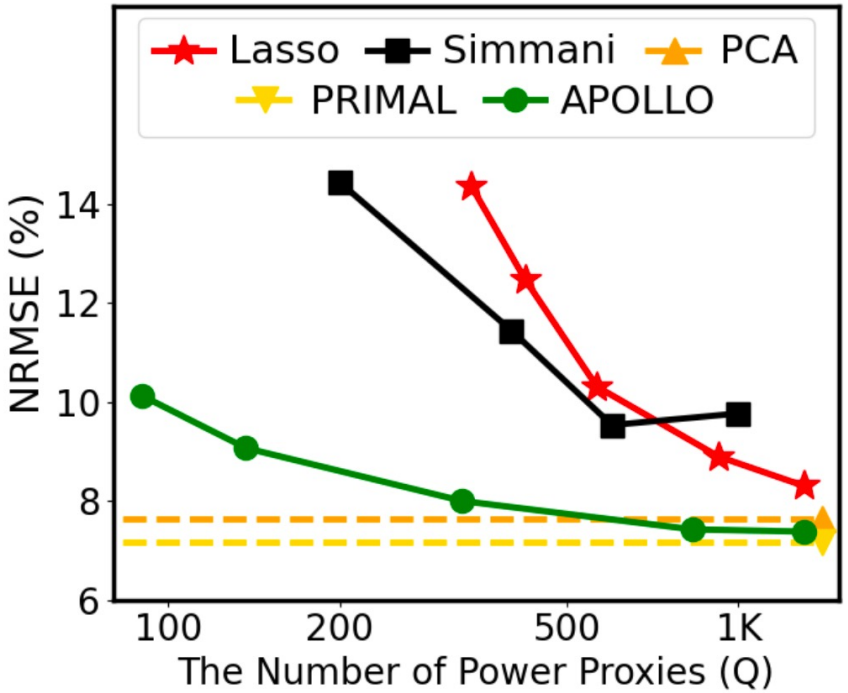
Comparison among various power modeling approaches. The percentage numbers are area overheads.

Duke UNIVERSITY

FAQ: Accuracy Comparison



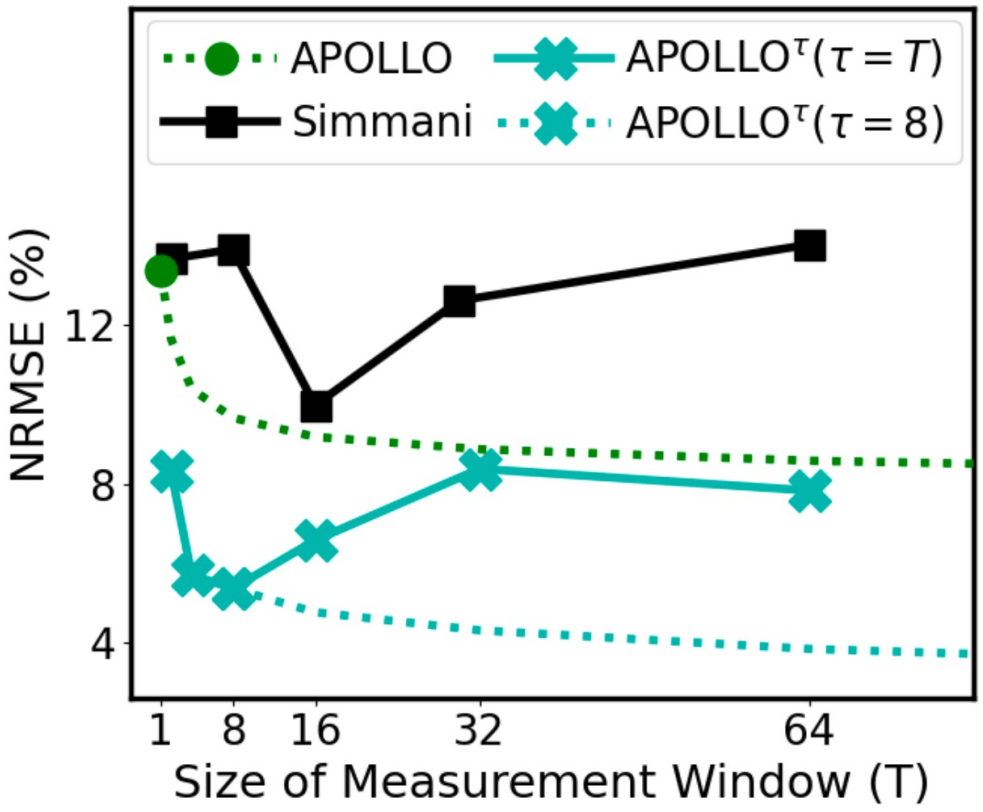
Neoverse N1 CPU core



Cortex-A77 CPU core

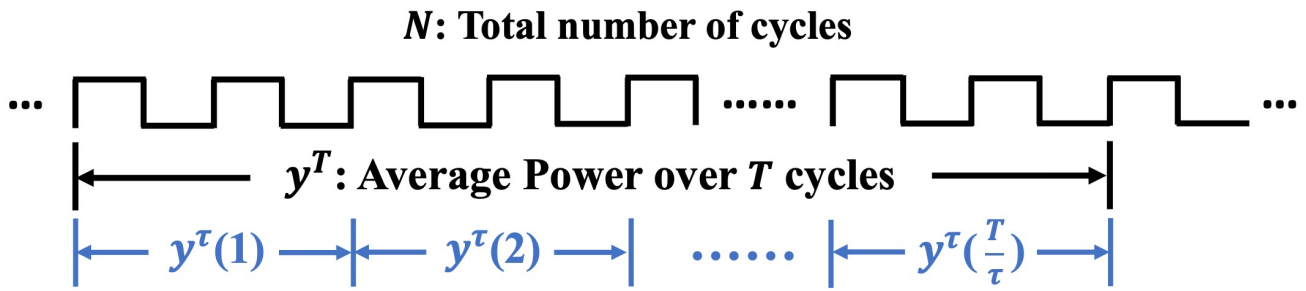
Duke UNIVERSITY

FAQ: Accuracy with Multi-cycle Measurement Window



Q = 200 for Simmani

Q = 70 for APOLLO

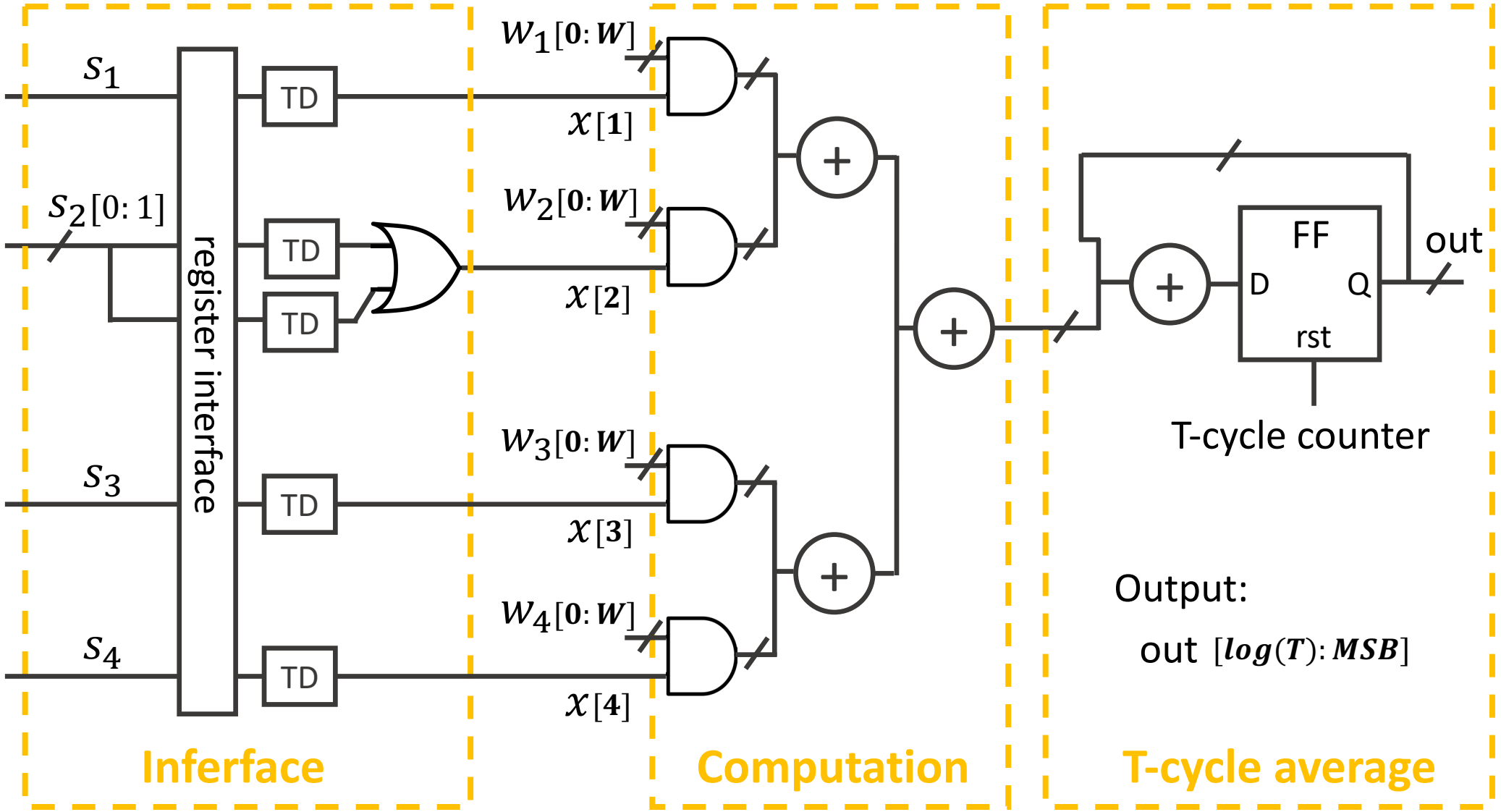


During inference:

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

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

Duke UNIVERSITY

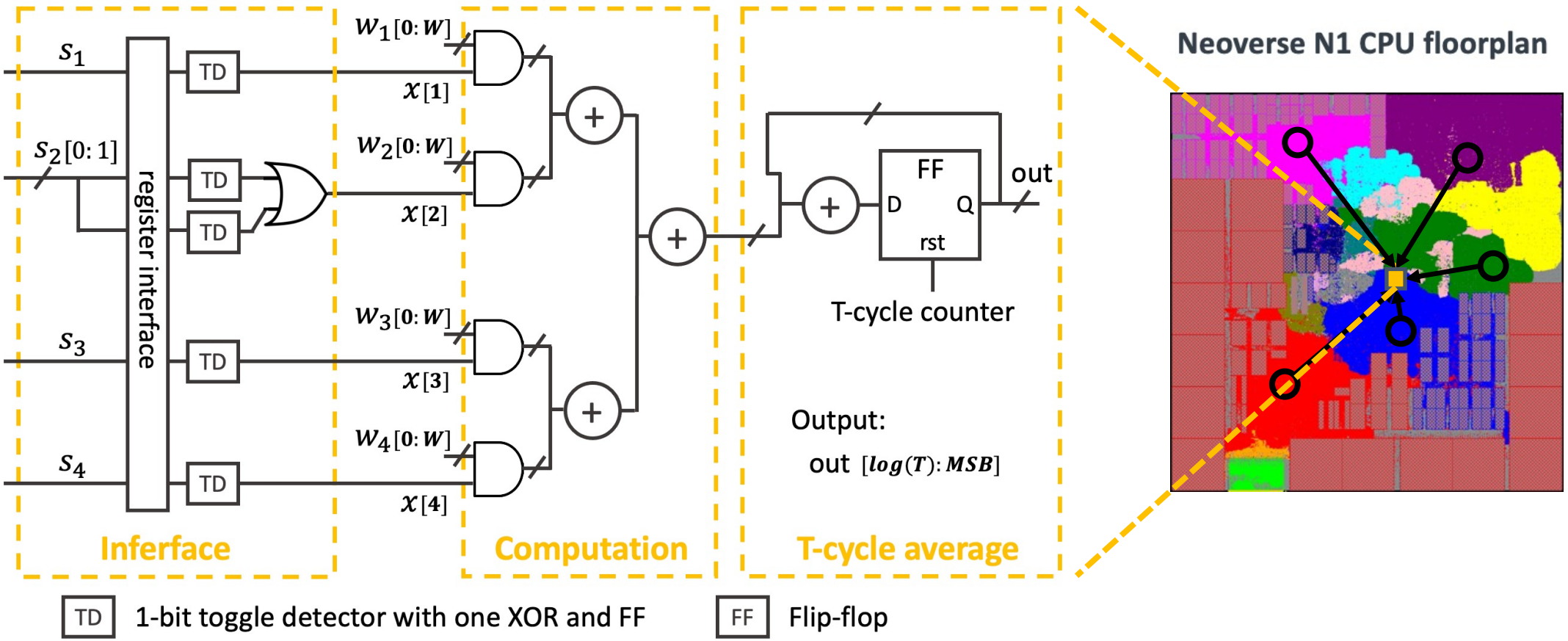


TD 1-bit toggle detector with one XOR and FF

FF Flip-flop

Duke UNIVERSITY

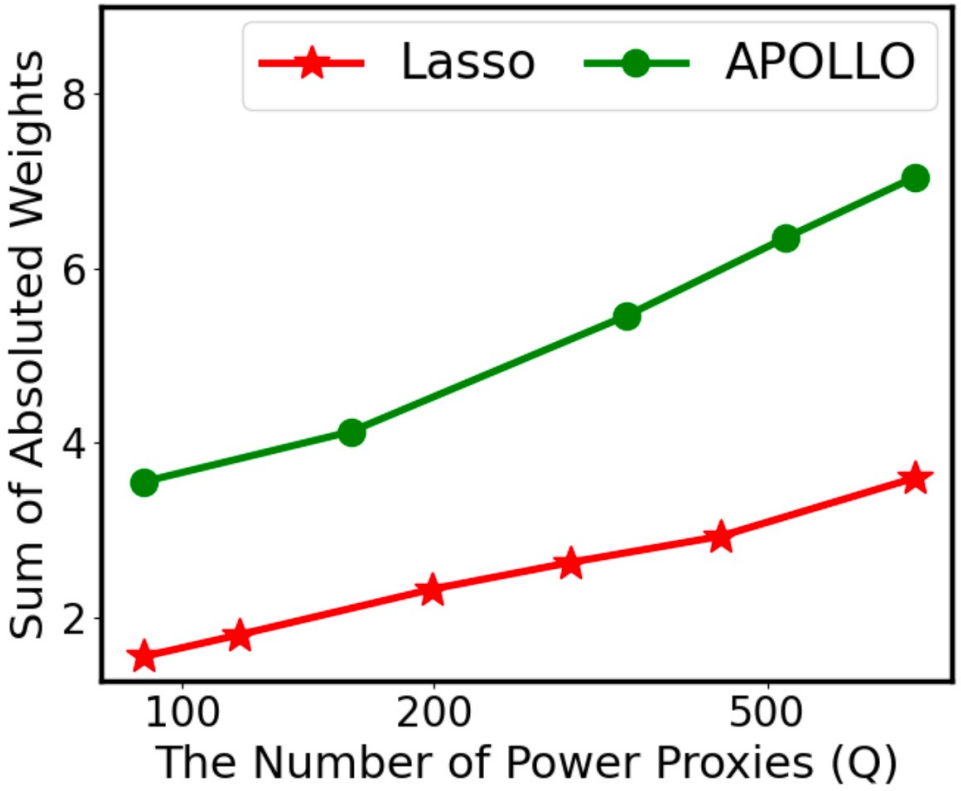
Overview of the OPM Hardware Design



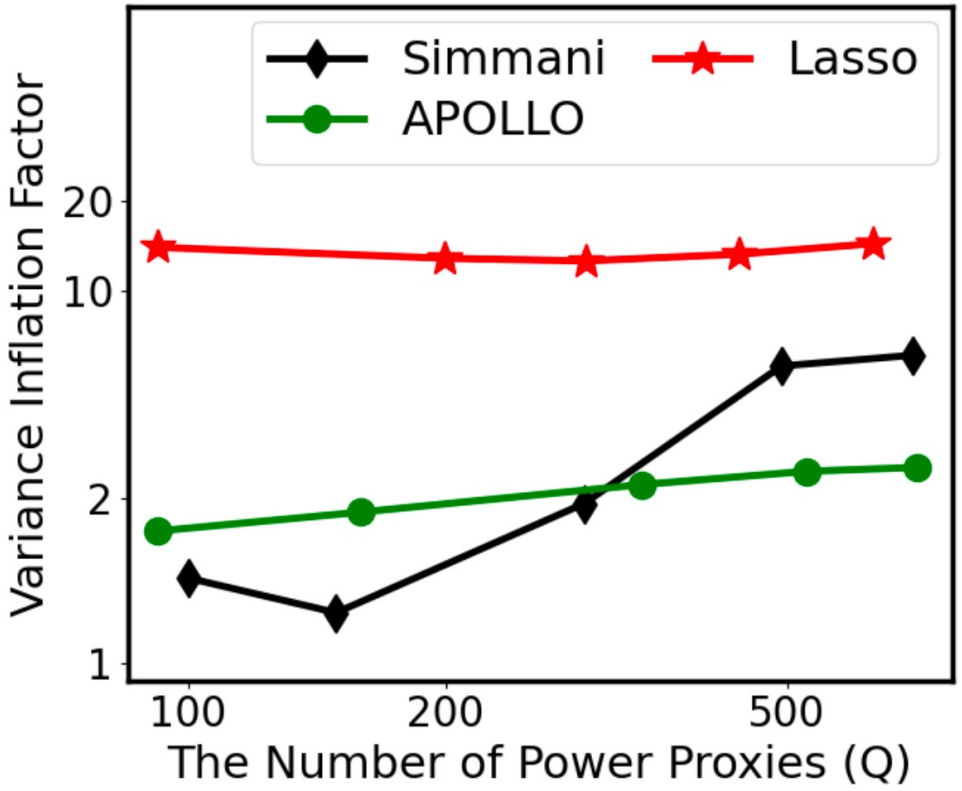
- **No** multipliers or dividers, only **Q** binary inputs and **W**-bit quantized weights

Duke UNIVERSITY

FAQ: Model Discussion



Sum of all absolute weights

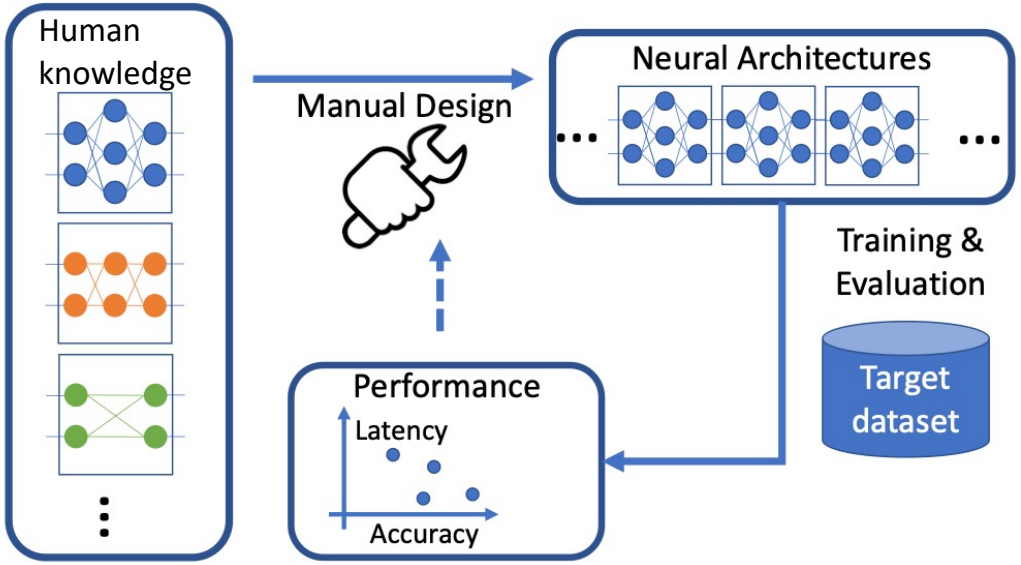


Variance inflation factors (VIF)

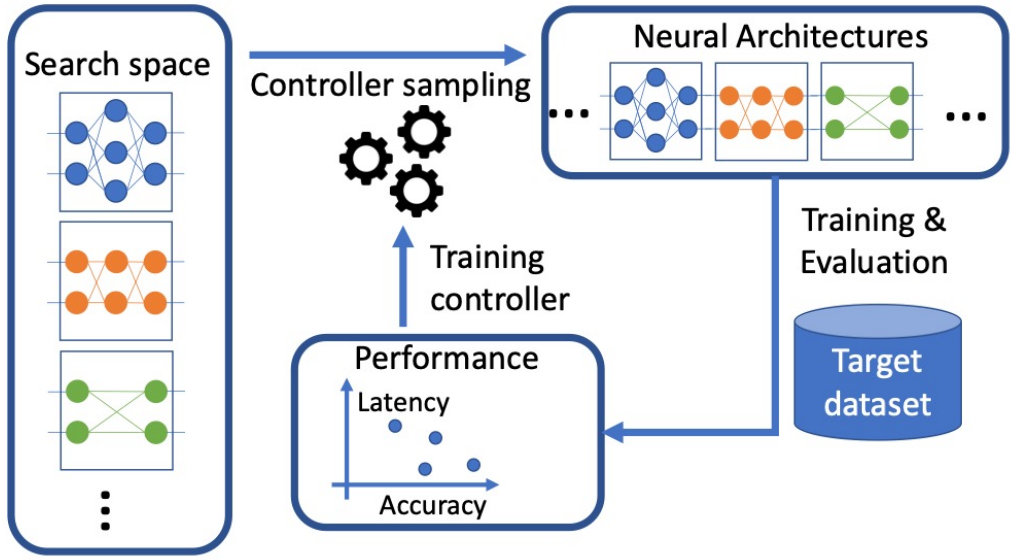
Duke
UNIVERSITY

Neural Architecture Search

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



Traditional DNN Design



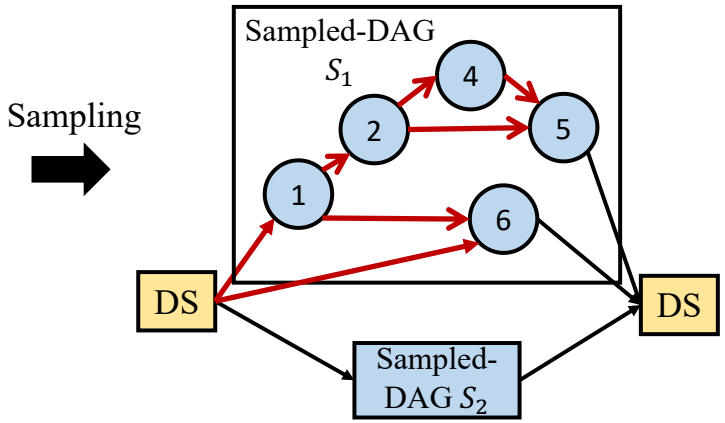
Neural Architecture Search

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

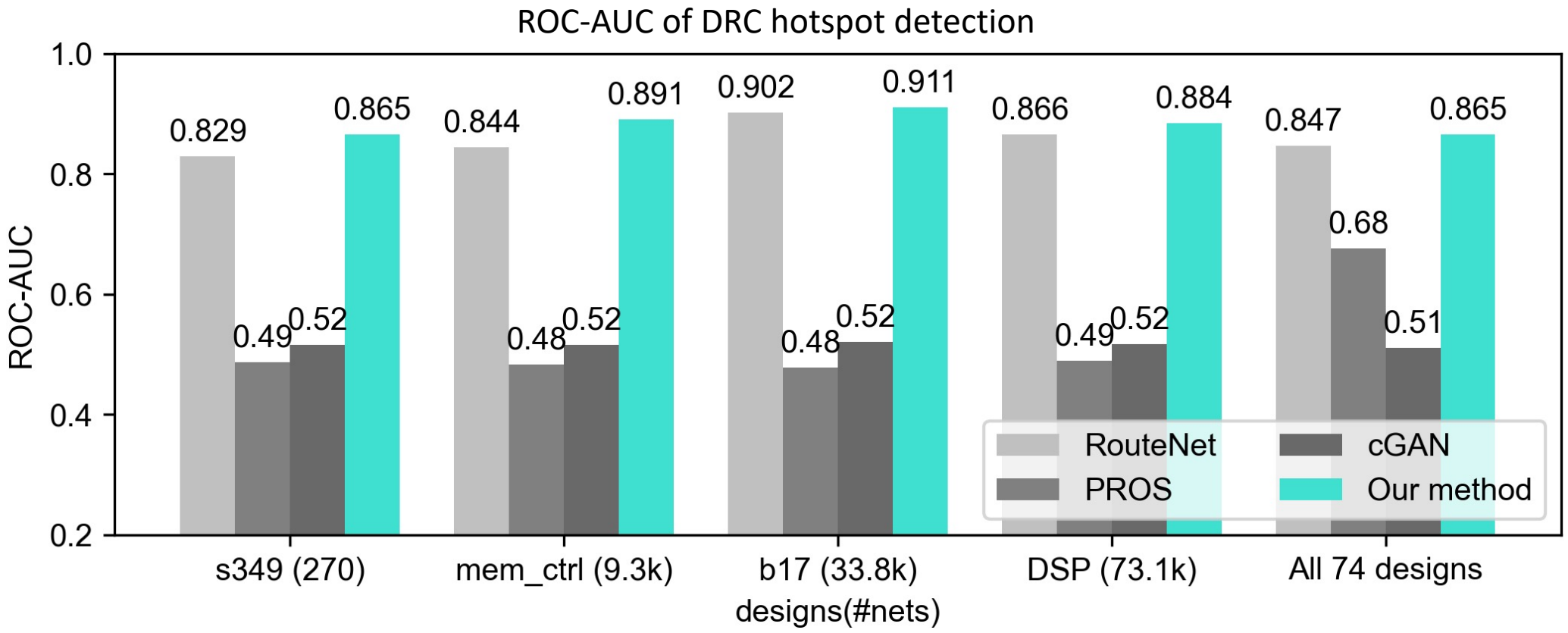
Search Strategy

- 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



Results from Automatically Developed Estimator

- This model **outperforms** RouteNet [ICCAD'18], PROS [ICCAD'20], and cGAN [DAC'19]
- Automatically developed without human in **one day**



[ICCAD'21], Session 9c: Automatic Routability Predictor Development Using Neural Architecture Search

Duke UNIVERSITY

Experimental Results – Accuracy

Comparison of the violated net count prediction

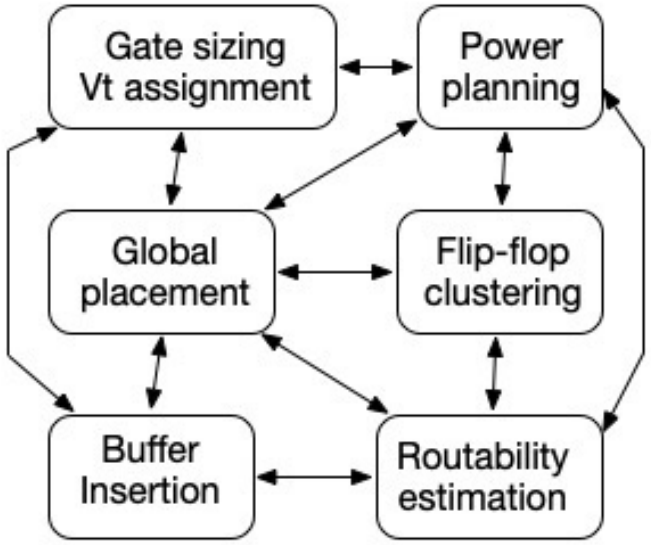
Models	Kendall's τ on designs (#nets)				Kendall's τ on all 74 designs	Pearson's correlation on all 74 designs
	s349 (270)	mem_ctrl (9.3k)	b17 (33.8k)	DSP (73.1k)		
RouteNet [3]	0.3620	0.1547	0.1779	0.4414	0.5264	0.7224
NAS-crafted model	**0.6369**	**0.4657**	**0.2683**	**0.7302**	**0.5572**	**0.7930**

Comparison of the DRC hotspot detection

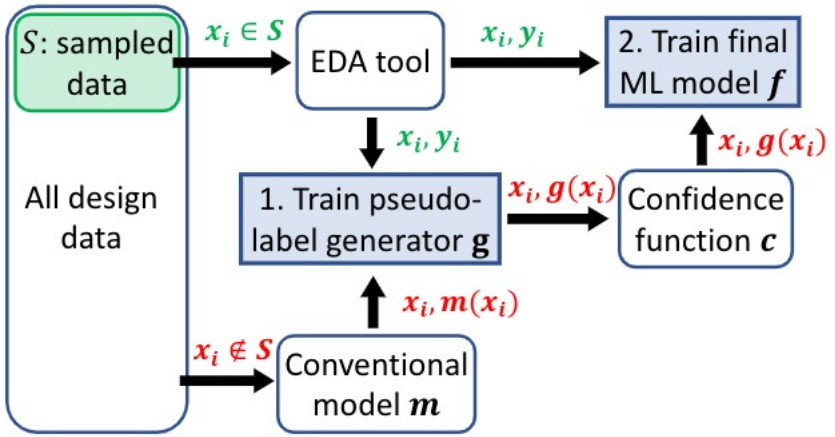
Models	ROC-AUC on designs (#nets)				ROC-AUC on all 74 designs
	s349 (270)	mem_ctrl (9.3k)	b17 (33.8k)	DSP (73.1k)	
RouteNet [3]	0.829	0.844	0.902	0.866	0.847
PROS [6]	0.487	0.483	0.478	0.489	0.676
cGAN [4]	0.516	0.515	0.521	0.517	0.510
NAS-crafted model	**0.865**	**0.891**	**0.911**	**0.884**	**0.865**

Proposals

A scenario of cross-domain optimization



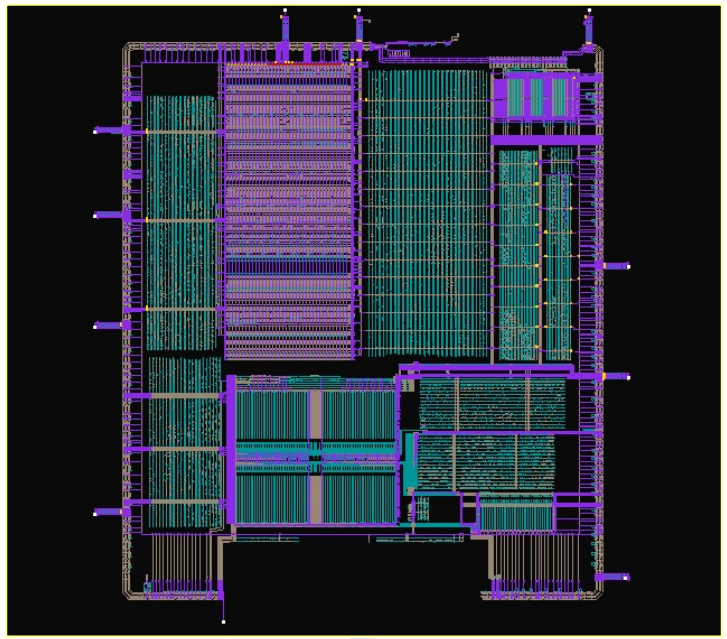
Tuning by semi-supervised learning



Duke UNIVERSITY

Cost on IR Drop Identification

Time-consuming simulations performed repeatedly

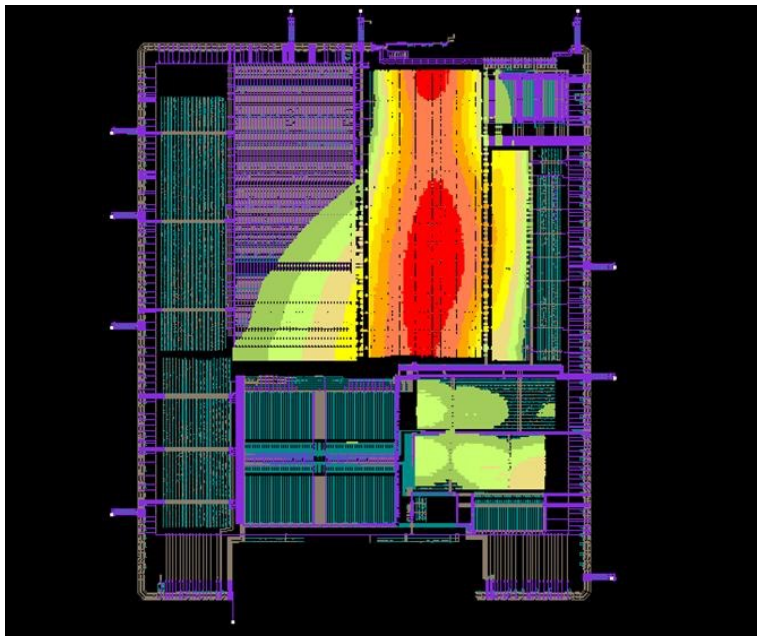


Provide feedback
Edit the design

Simulation to get IR drop hotspots

Ansys ~~~4 hours~~ on 1M cells

~~Not only once!~~



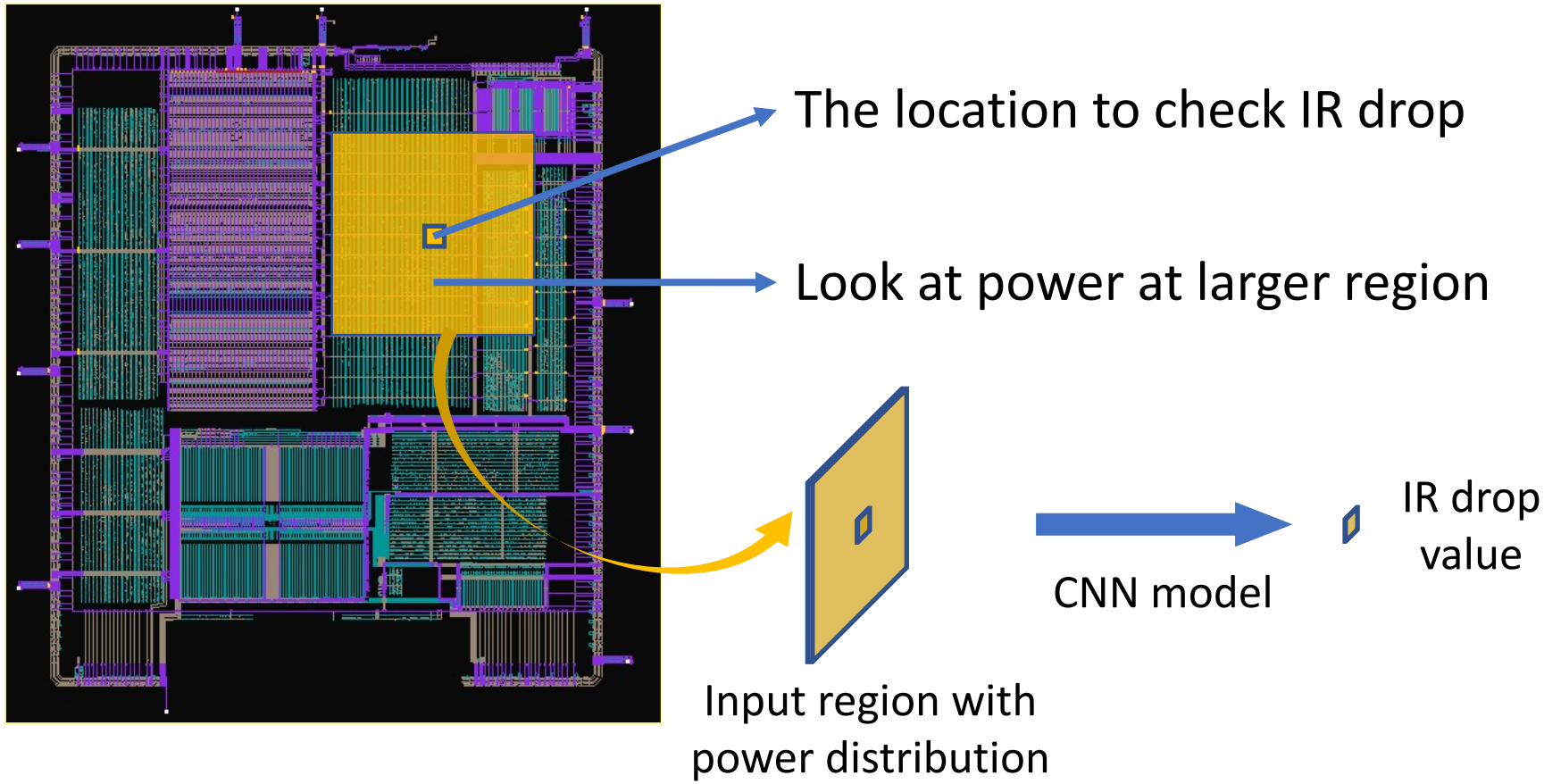
~~Simulate again~~

Check other trade-off design objectives

ML-based algorithm: 5 mins, 93-99% accuracy

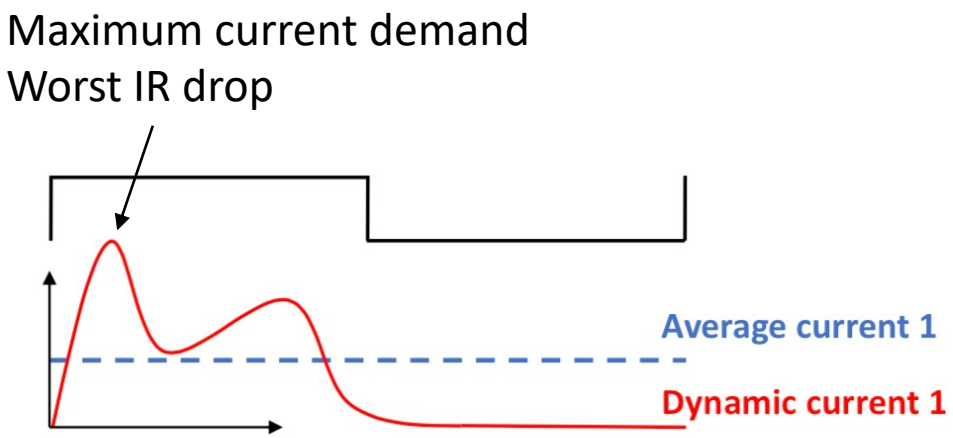
Duke UNIVERSITY

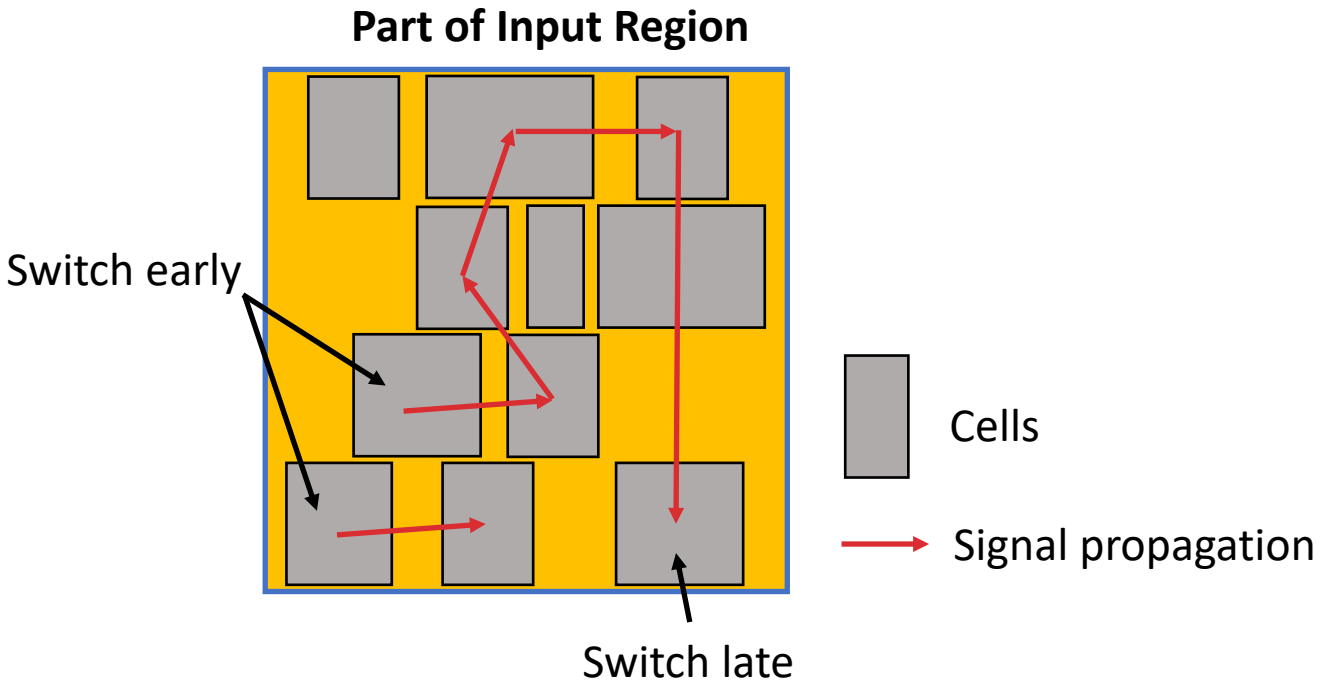
IR Drop Estimation Methodology



- Neighbors contribute power/current demand at the local region

Incorporate Timing with Customization

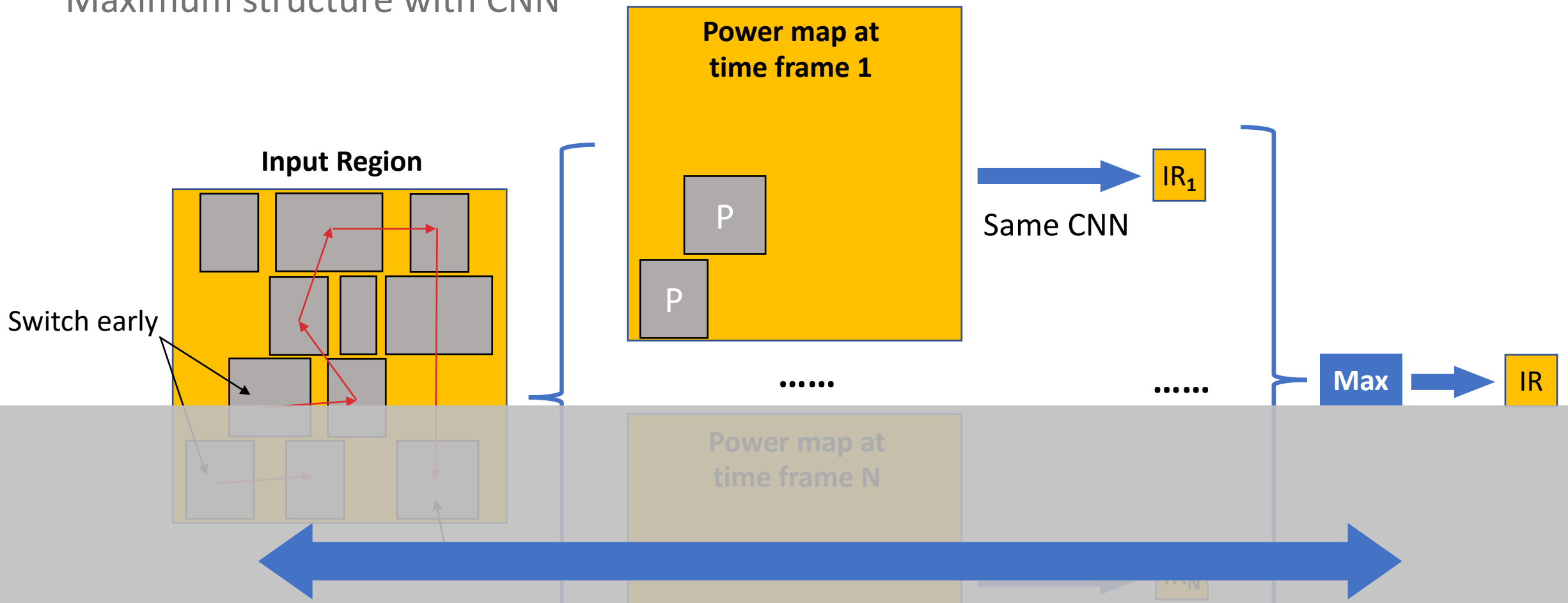




- Power/current consumption depends on switching activities
- Gates switch at different time frame!

Our Customized Method

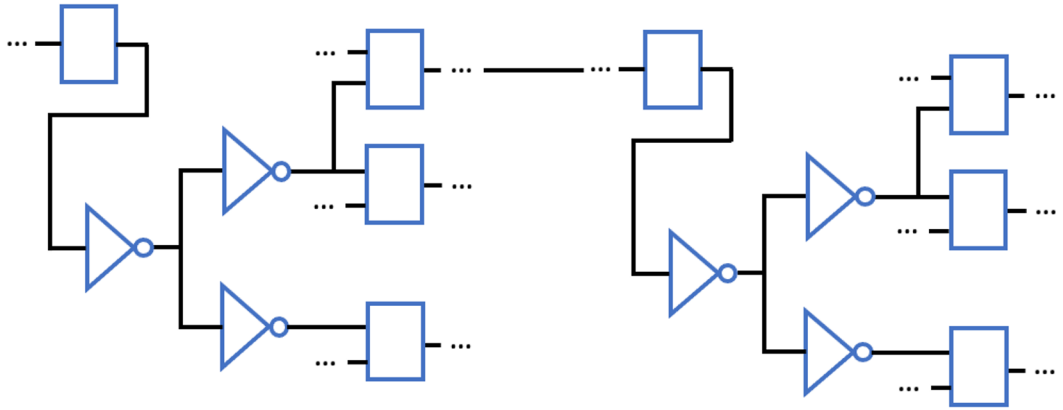
Maximum structure with CNN



Performs end-to-end training

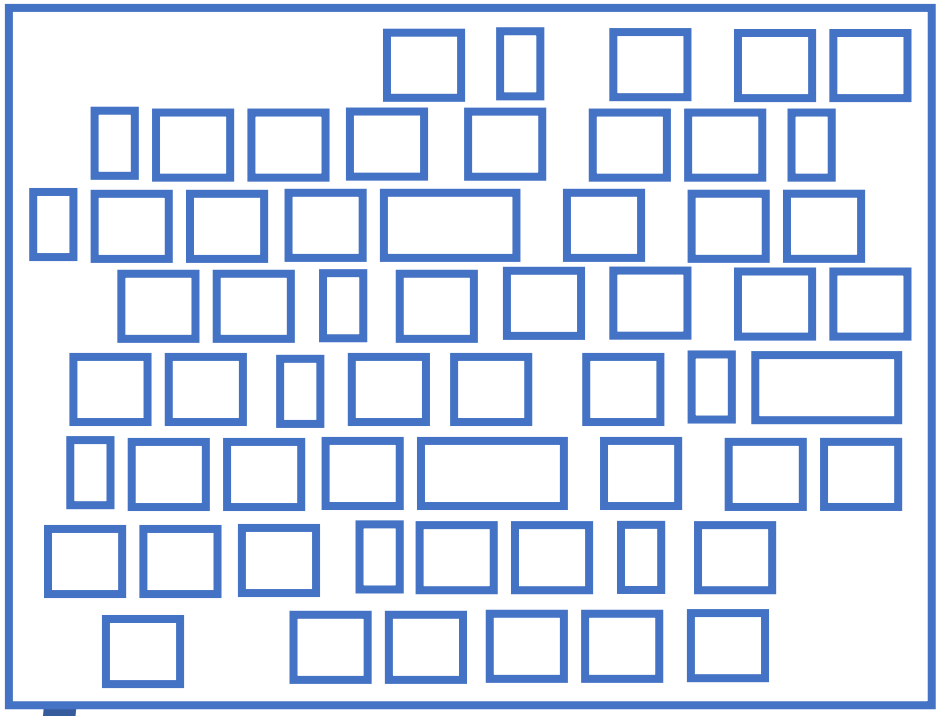
Timing Estimation on Netlist is Inaccurate

Circuit Netlist



Placement
Hours-days

Circuit Layout



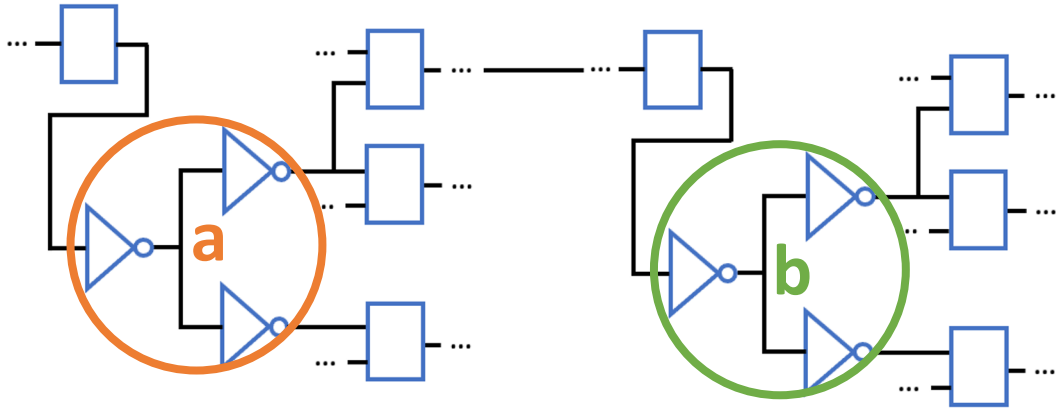
Inaccurate timing estimation due to **absence** of wire length

Better timing estimation

Edit netlist if necessary

Example: Timing Estimation before Placement

Circuit Netlist

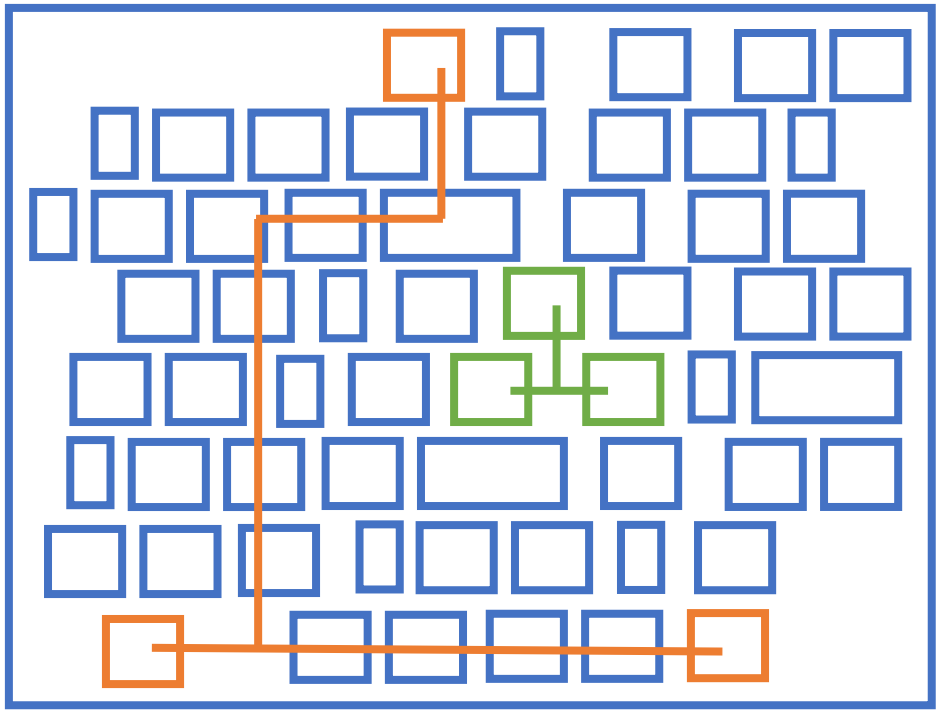


Net-length?

Net-length?

Placement
Hours-days

Circuit Layout

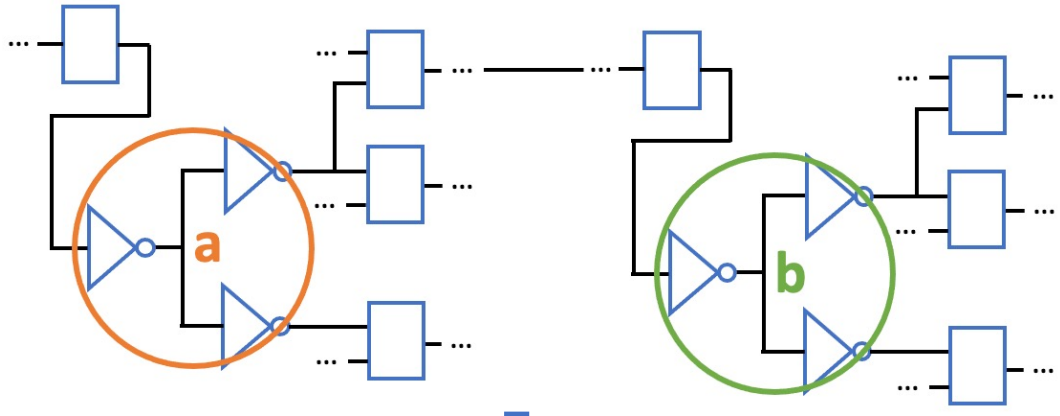


Traditional tools: **Same!**
Previous works: **Similar!**

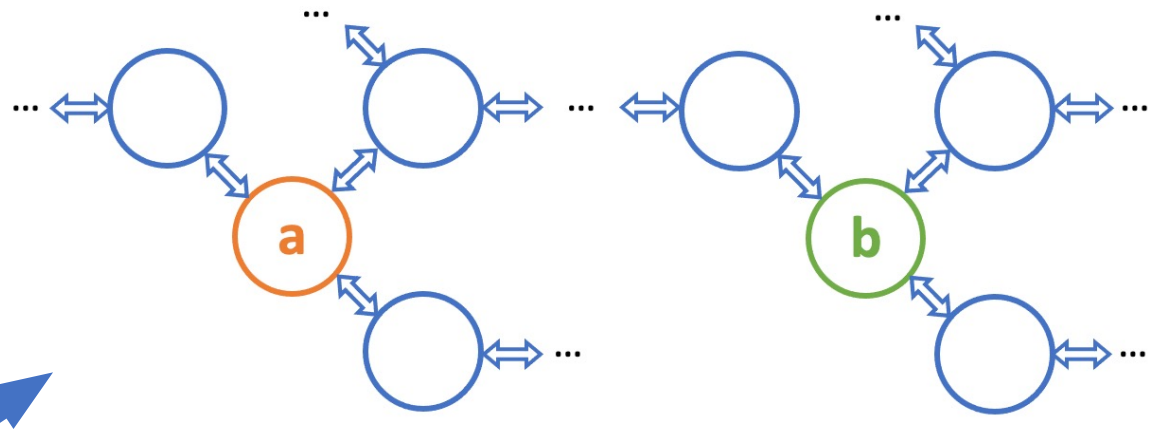
Net-length **a** >> Net-length **b**

Delay **a** >> Delay **b**

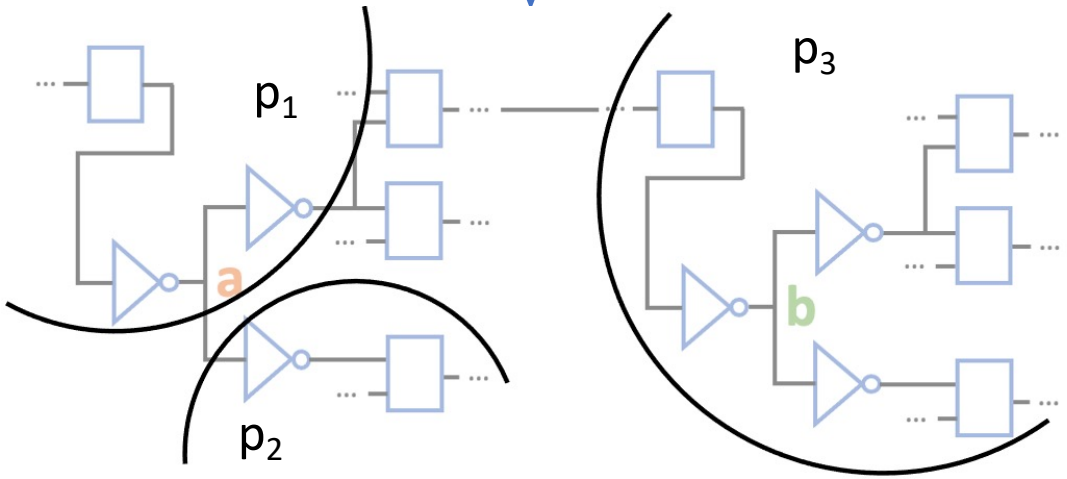
Our Solution to Capture Global Information



Process as Graph



Partitioning



Extra Info

Customized GNN

Recognize net-length a >> net-length b

Timing model

Reduces 50% error in commercial timing engine