Truly Intelligent Circuit Design and Implementation

Zhiyao Xie

Dept. Electrical & Computer Engineering

Duke University

Outline of My Talk

- Part 1: My Ph.D. Works

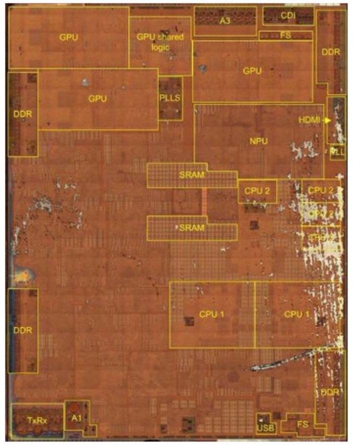
- Part 2: My Future Plan

Electronic Devices are Everywhere

*These images are found in public domain

Designers Try to Deliver Generational Gains

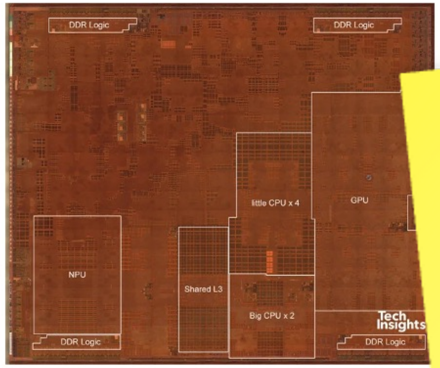
iPhone 8, X



Apple A11

10nm
4.3 B trasistors

iPhone XS, XR



Apple A12

7nm
6.9 B trasistors

iPhone 11

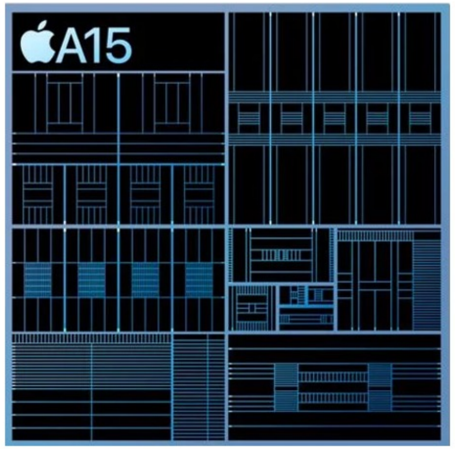
Looks good!
Any challenges?

iPhone 12

Apple A14

5nm
11.8 B trasistors

iPhone 13



Apple A15

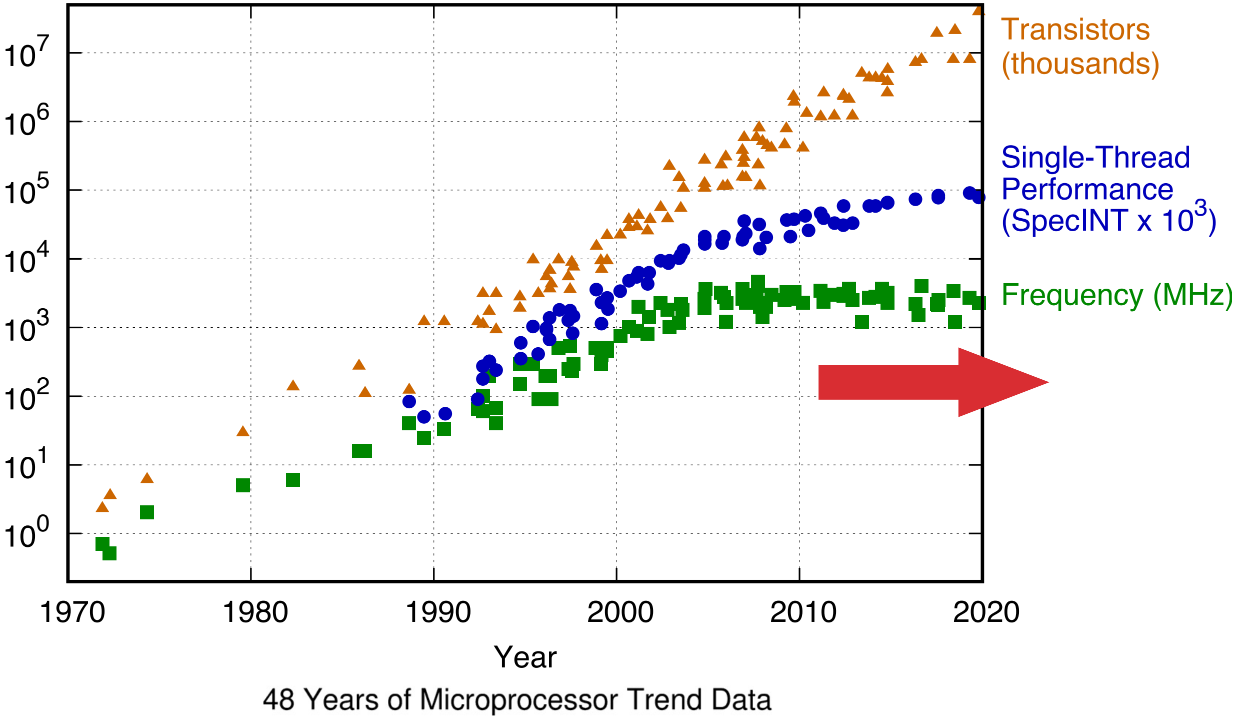
5nm
15 B trasistors

*Source: TechInsights Inc.

Chip Design Challenges

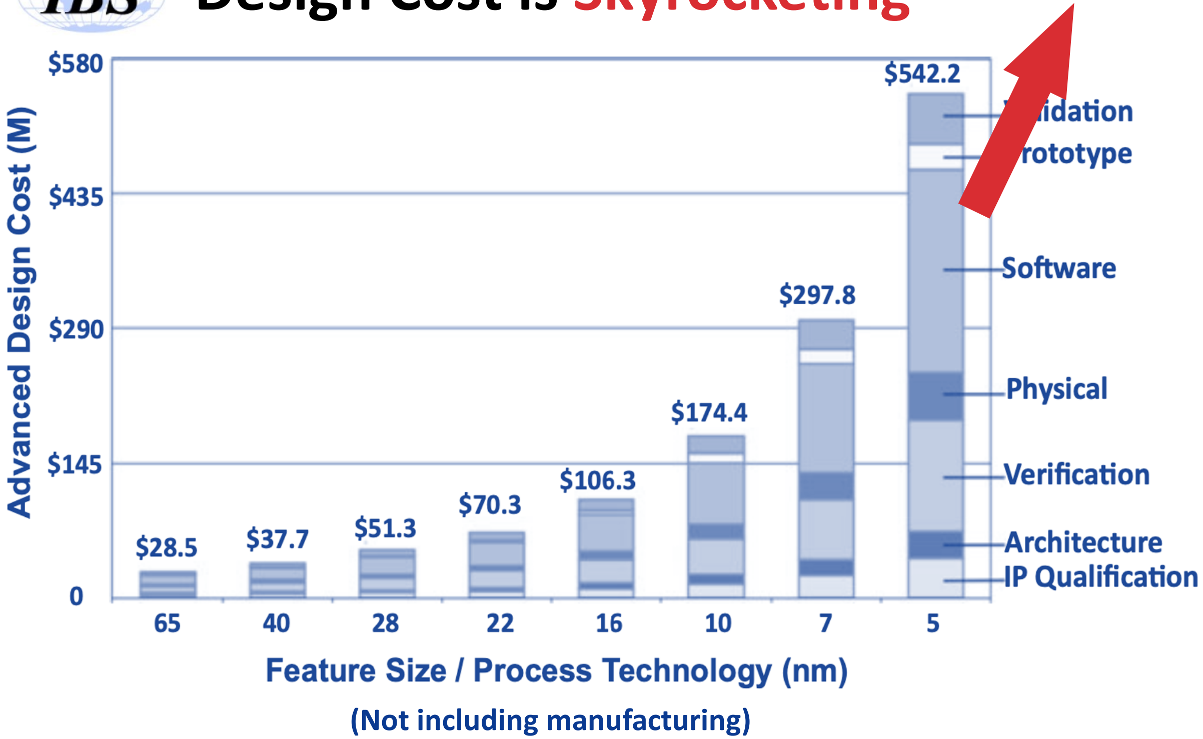
Diminishing performance gain and increasing design cost

Per-Core Performance Gain is Diminishing



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

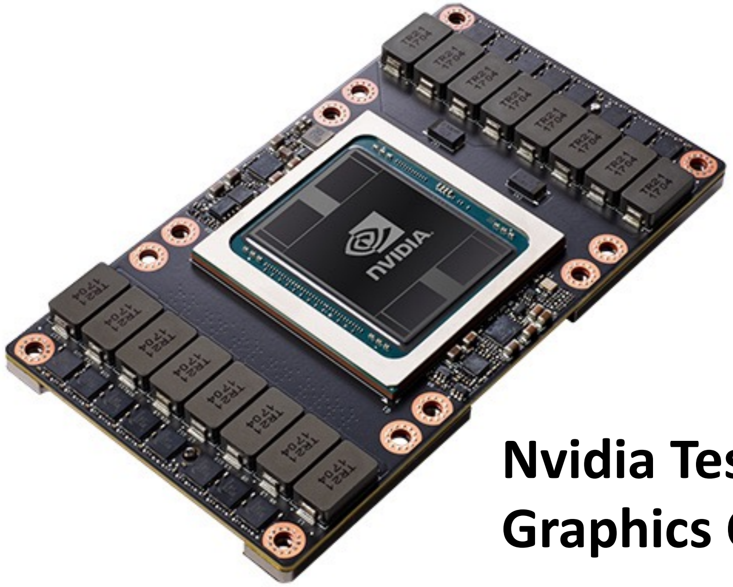
IBS Design Cost is Skyrocketing



International Business Strategies, 2020

Chip Design Challenges

Not only costly, also long turn-around time



Nvidia Tesla V100
Graphics Card

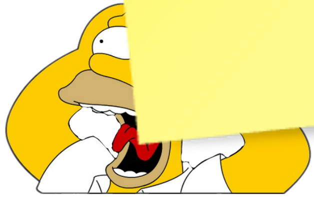
It took **several thousand** engineers **several** years to create, at an approximate development cost of **$3 billion**. – Jensen Huang, CEO of Nvidia

Nvidia GPU Technology Conference (GTC), 2017

This is Real Problem!

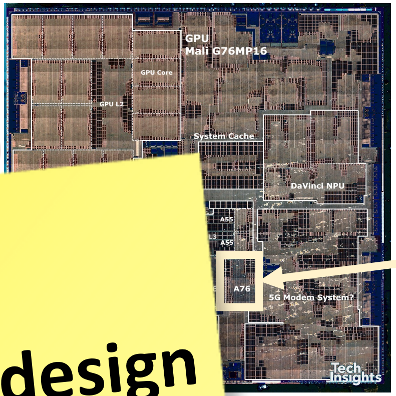
Challenges at advanced node

- **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!

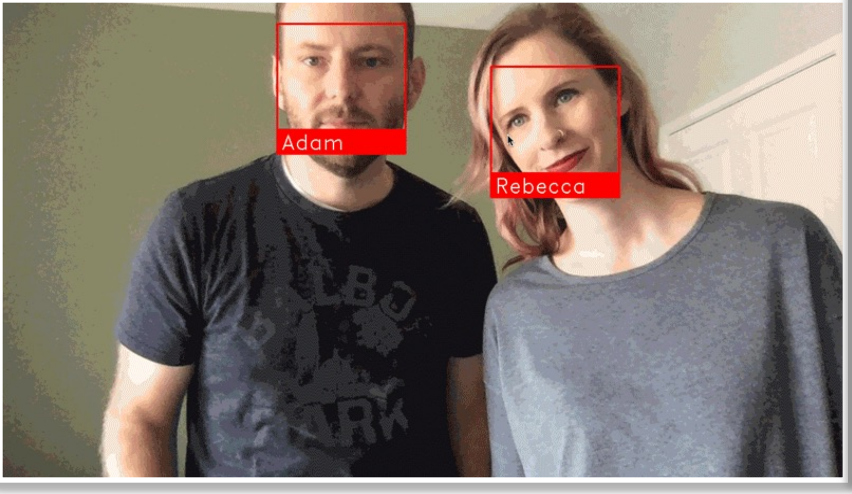
Inefficient chip design methodologies



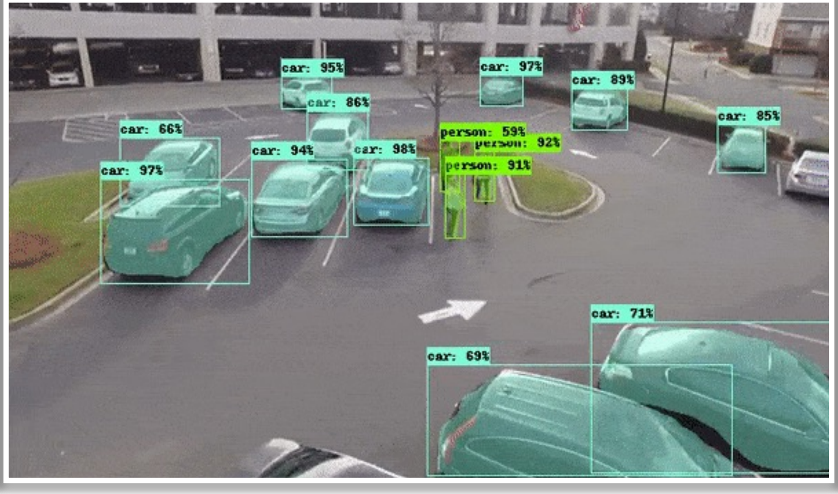
For one Arm CPU core with ~3 million gates

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

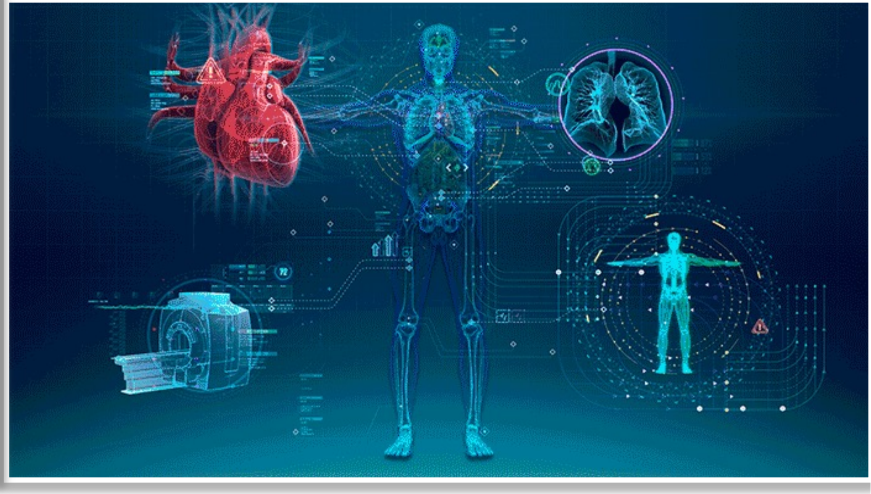
Source: The Kirin 990 SoC. TechInsights Inc.



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

SMART

Health Monitor

Personal Assistant

Manufacturing

Smart Grid

Financial Service

HPC

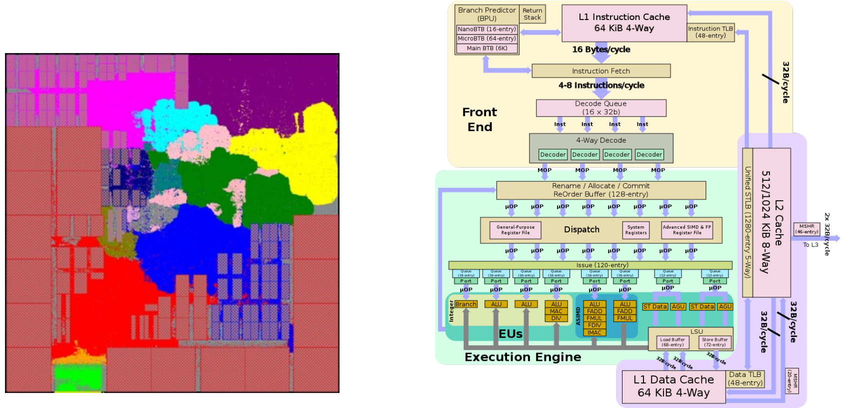
Security

Gaming

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)

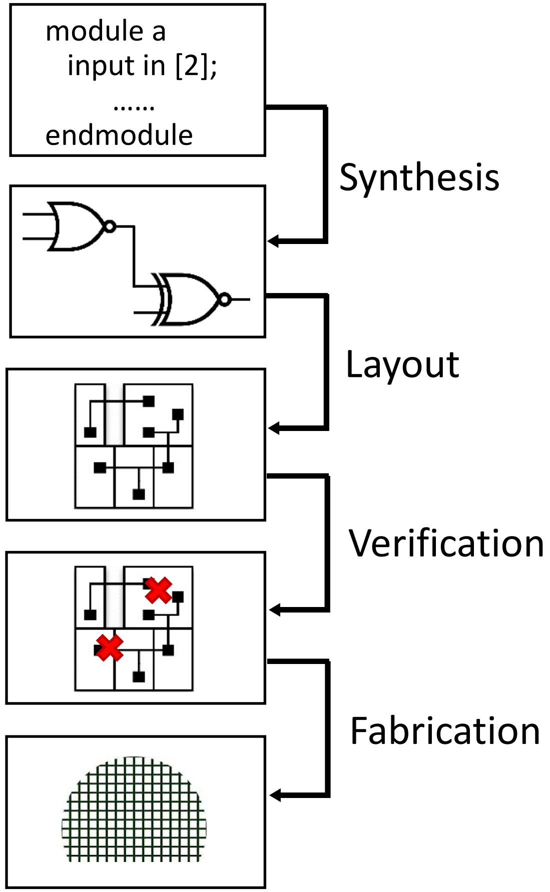
- Millions of connected components
- 100s GB of raw information
- Need simulations to get the label
- Data is hard to get
- ……

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



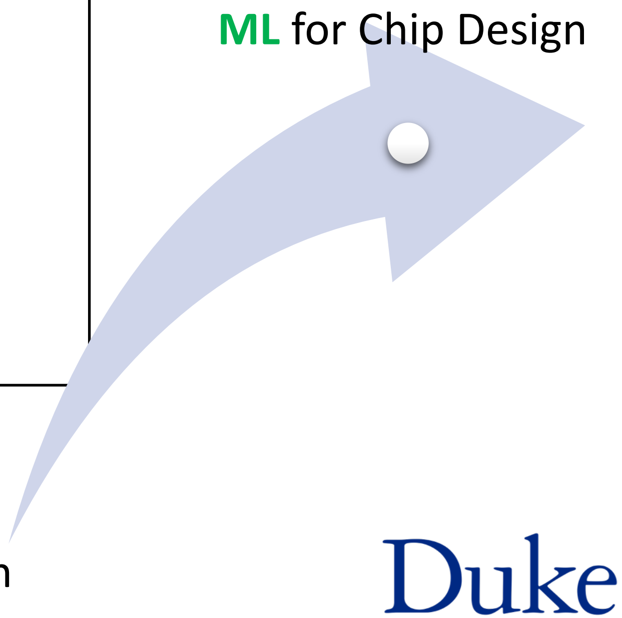
```
module a
  input in [2];
  ......
endmodule
```

ML for EDA in commercial tools

Cadence Innovus™

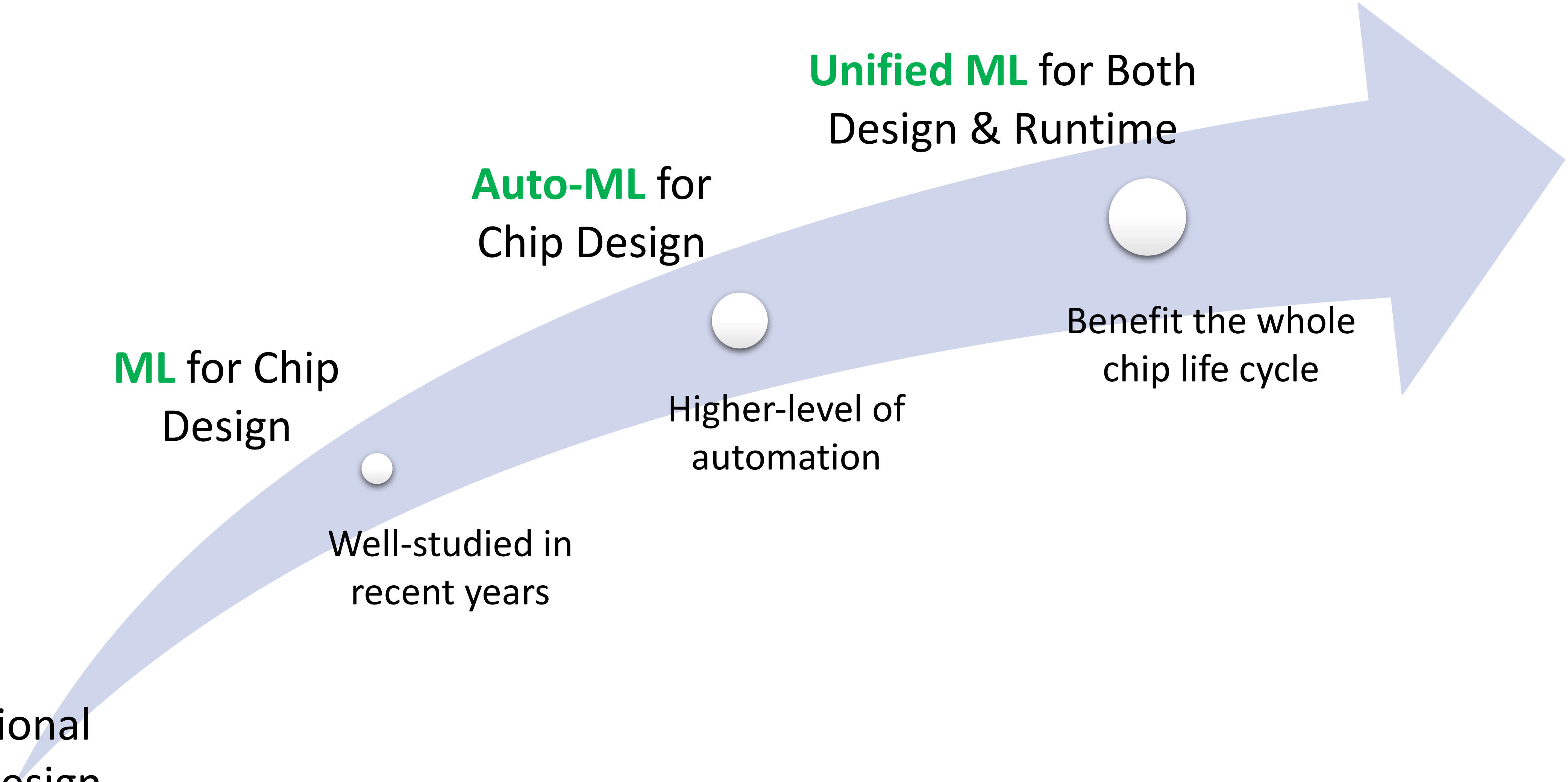
Synopsys ICC™ II

......



Electronics Research Initiative (ERI) – Design
Goal: 24 hours turnaround time & no human

What I Believe We Should Target



Traditional Chip Design

ML for Chip Design

Well-studied in recent years

Auto-ML for Chip Design

Higher-level of automation

Unified ML for Both Design & Runtime

Benefit the whole chip life cycle

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]

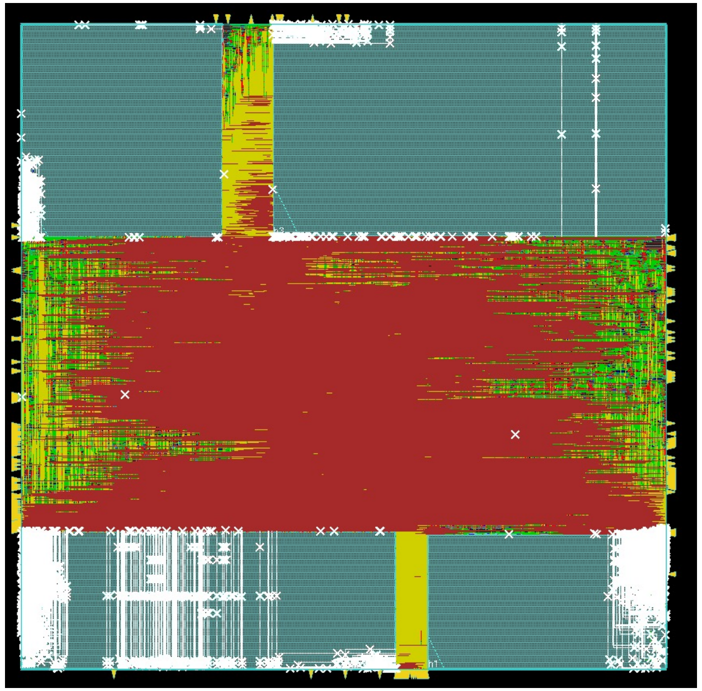
🟧 Covered in this talk

Case Study 1:

Routability Challenges

Routability Background

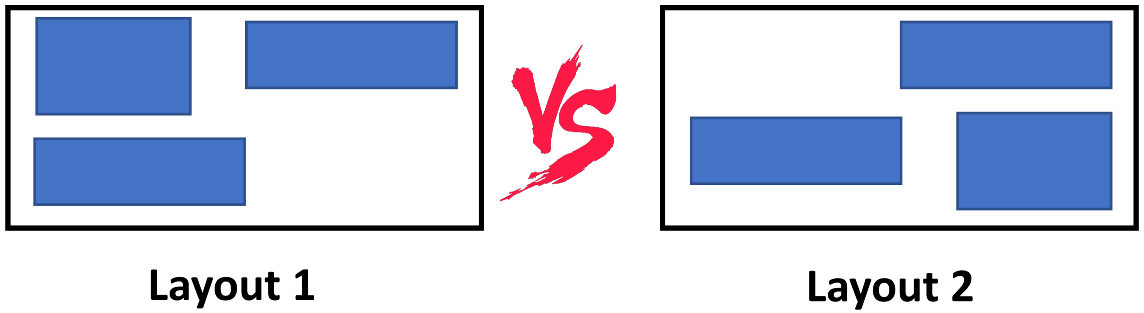
- Design Rule Checking (DRC)
  - ➢ Meeting manufacturing requirements
  - ➢ Less DRC violations (DRV) -> better routability
- DRV mitigation at early stages
  - ➢ Requires routability prediction/estimation
- Previous routability (DRV) estimations
  - ➢ Inaccurate or not fast enough

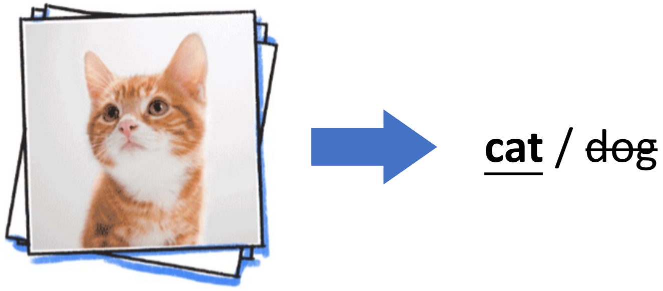


DRC violations (white) on circuit layout

First Deep Learning Method for Routability Prediction

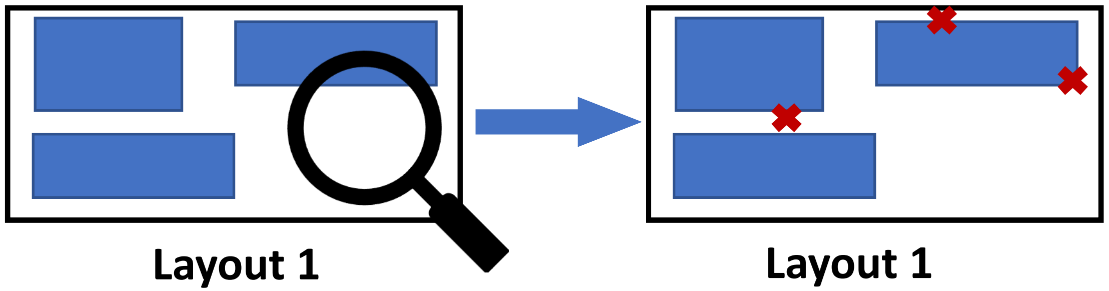
- Task 1: which one will result in less DRV count?

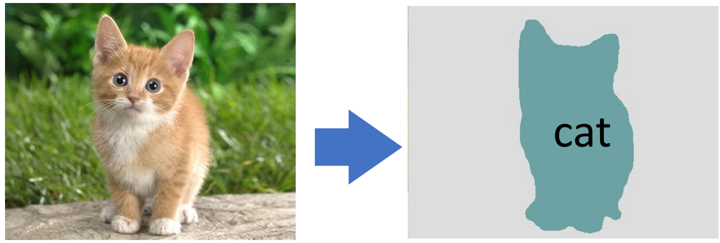




Customized CNN methods

- Task 2: where are DRC violations?



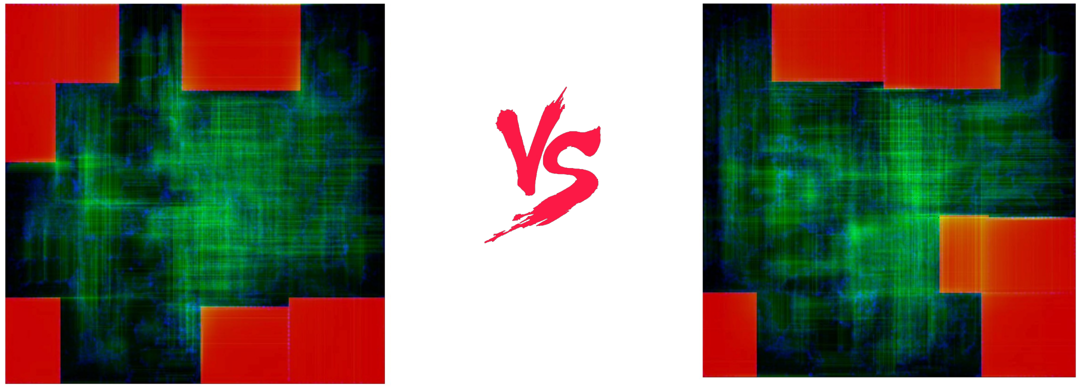


Customized FCN methods

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

First Deep Learning Method for Routability Prediction

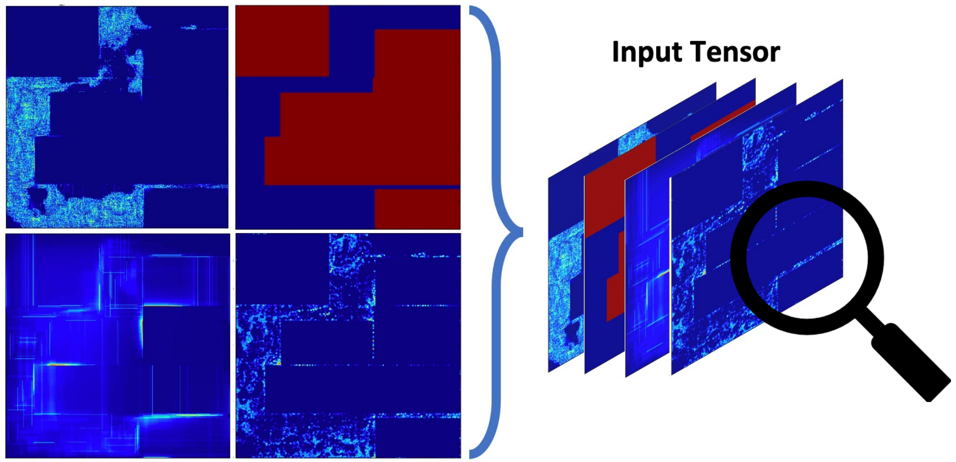
- Task 1: which one will result in less DRV count?



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

In seconds, with similar accuracy

- Task 2: where are DRC violations?

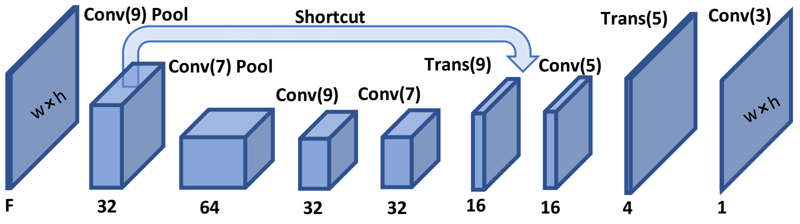


- Requires detailed routing
  ~~More hours~~ * Iterations

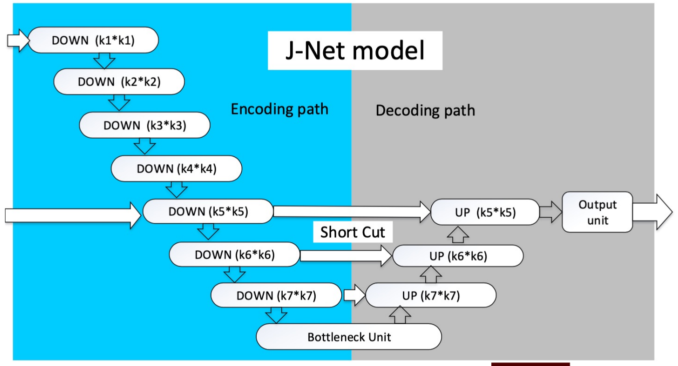
In seconds, outperform previous works

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

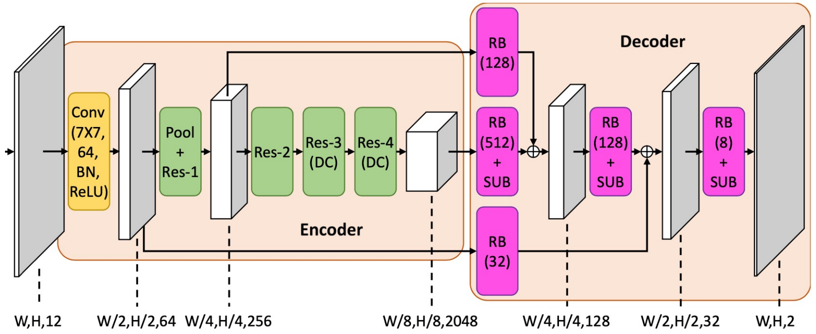
Many Excellent Deep Learning Methods



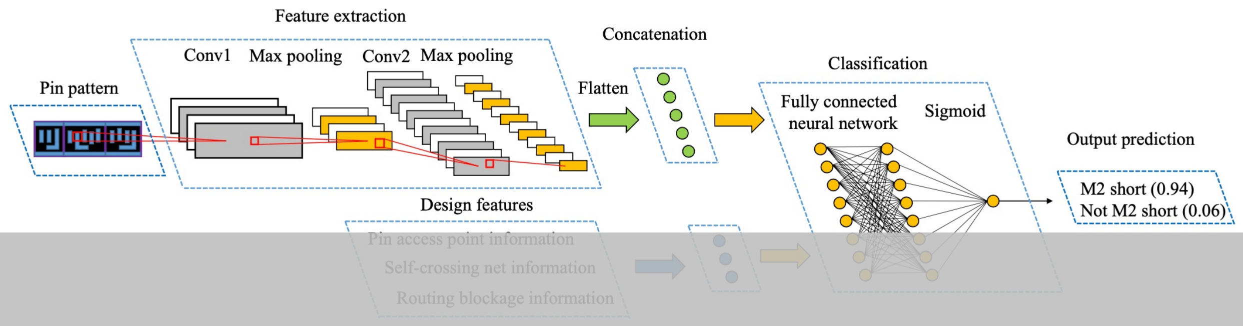
RouteNet [Xie, et al., ICCAD'18]



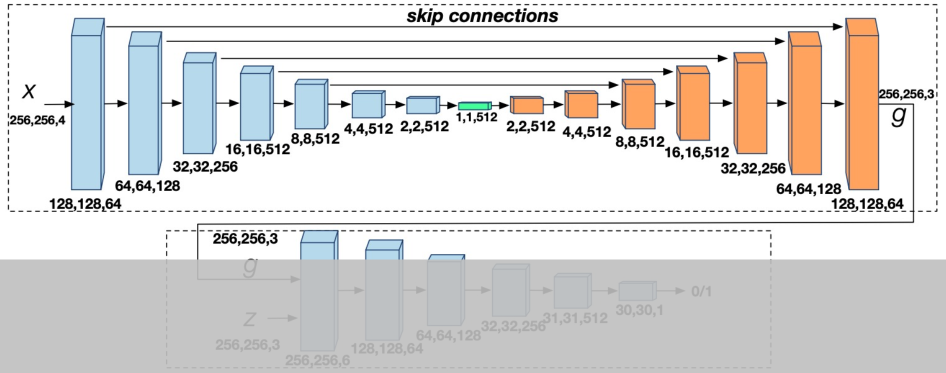
J-Net [Liang, et al., ISPD'20]



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



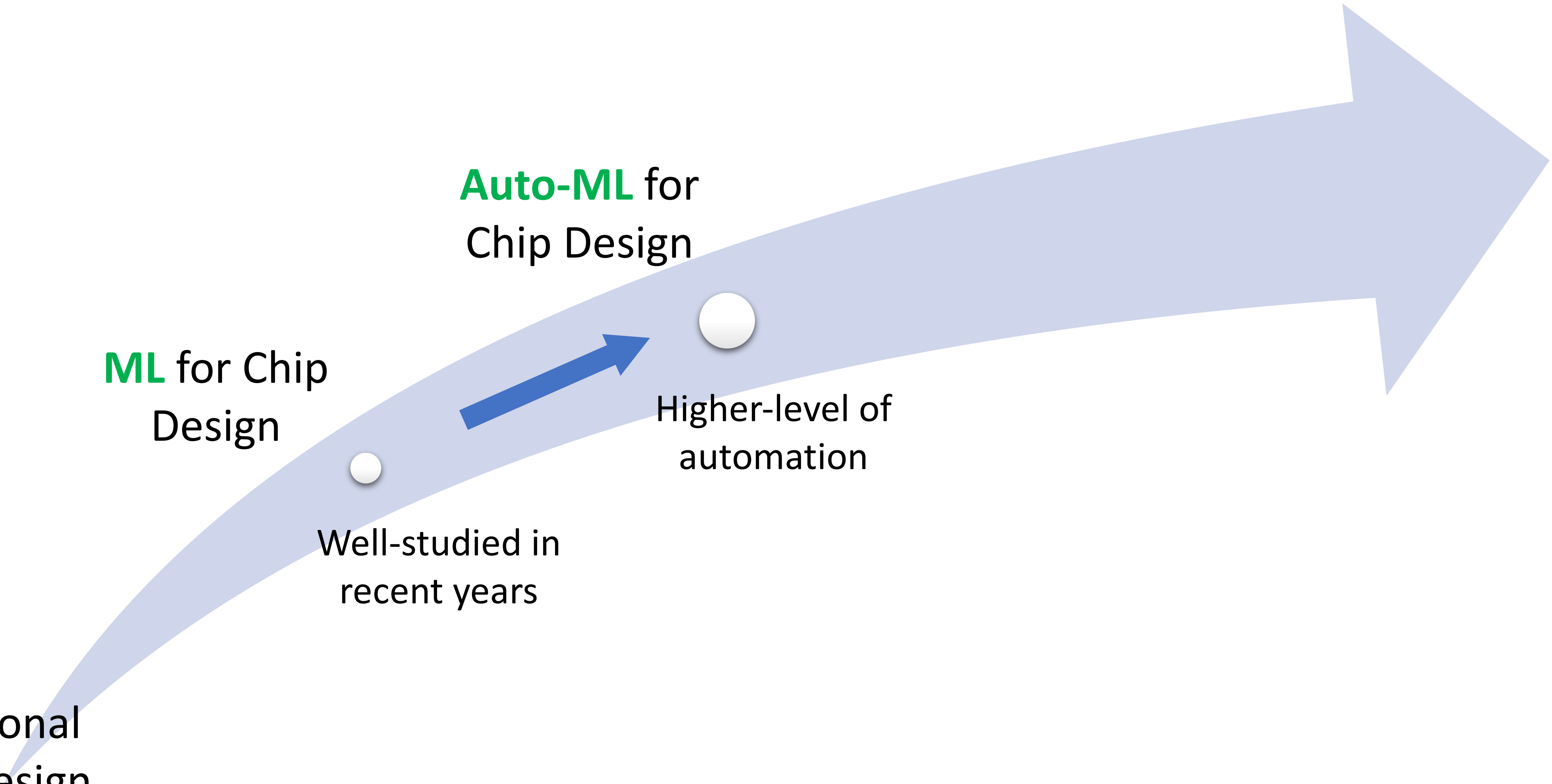
[Yu, et al., DAC'19]



Painting, Yu, et al., DAC'19

Tremendous Engineering Efforts Required!

What I Believe We Should Target



Traditional Chip Design

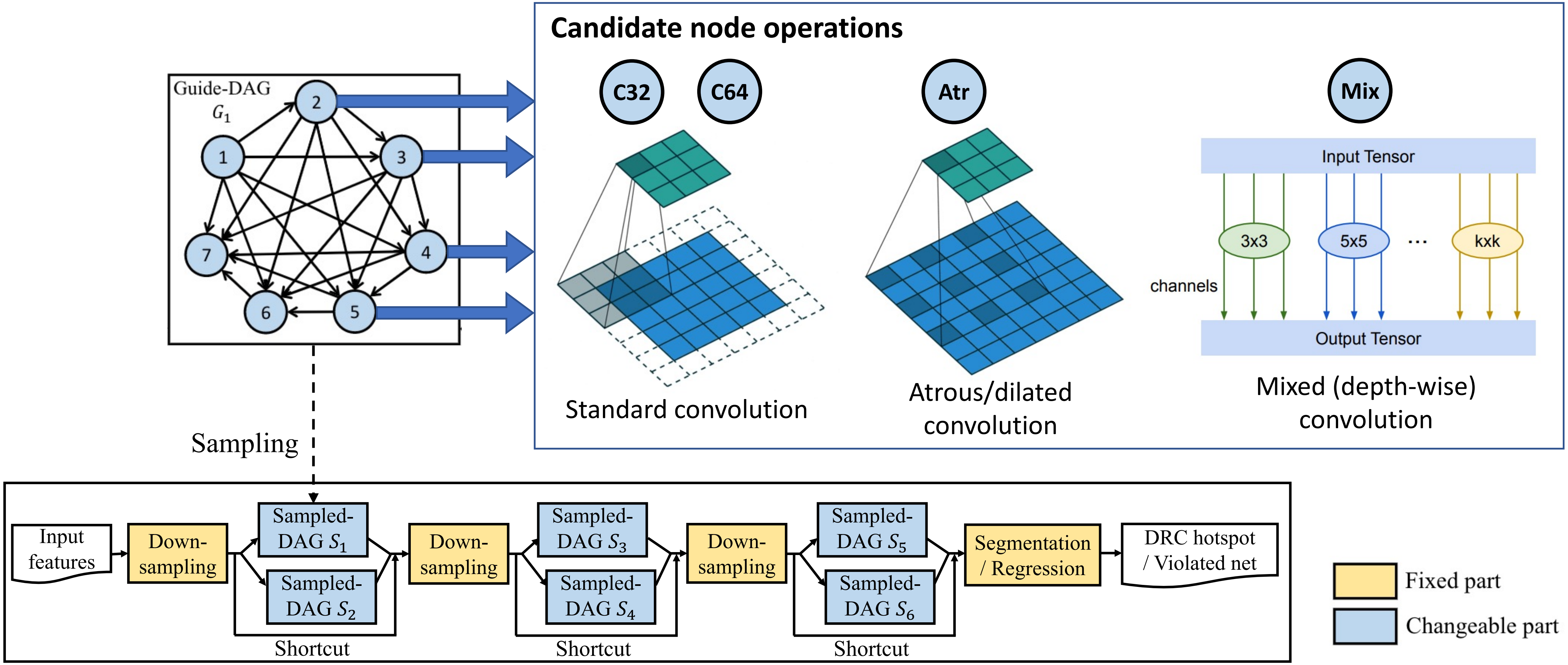
ML for Chip Design

Well-studied in recent years

Auto-ML for Chip Design

Higher-level of automation

Automatic Estimator Development – Search Space



Candidate node operations

Standard convolution

Atrous/dilated convolution

Mixed (depth-wise) convolution

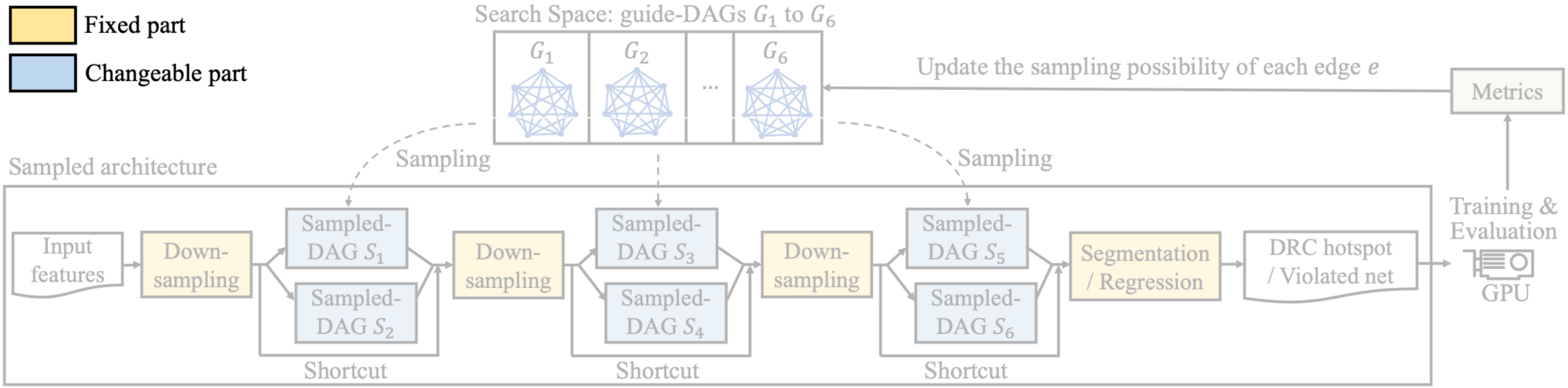
Sampling

Fixed part

Changeable part

[Chang, et al., ICCAD’21]

Automatic Estimator Development – Searching Algorithm

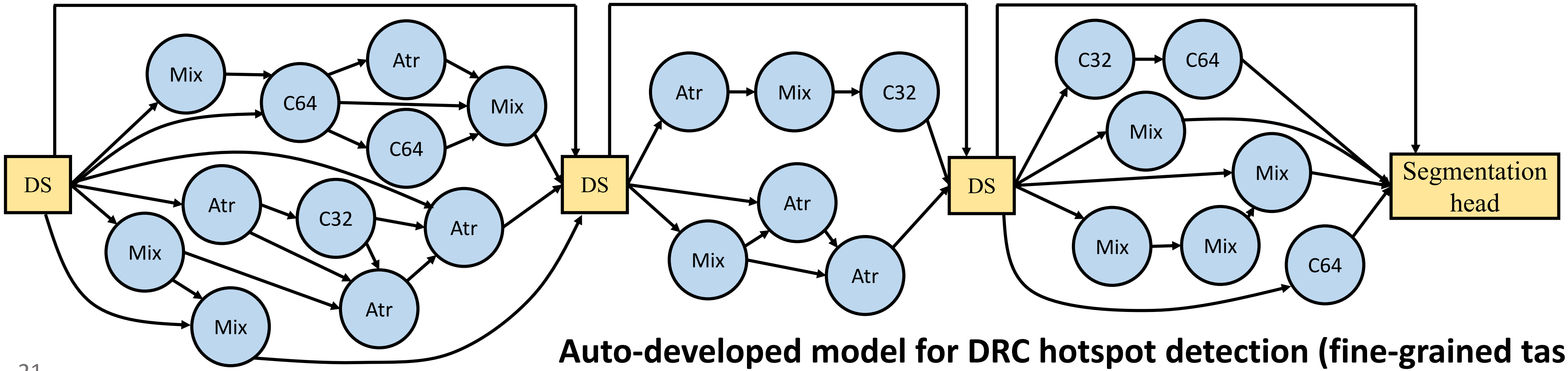


1. Sample from the completely-ordered graph (G_i) to get (S_i)
2. Evaluate the sampled model by training and testing
3. Update the sampling probability by evaluation result

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

Auto-developed Model Structures

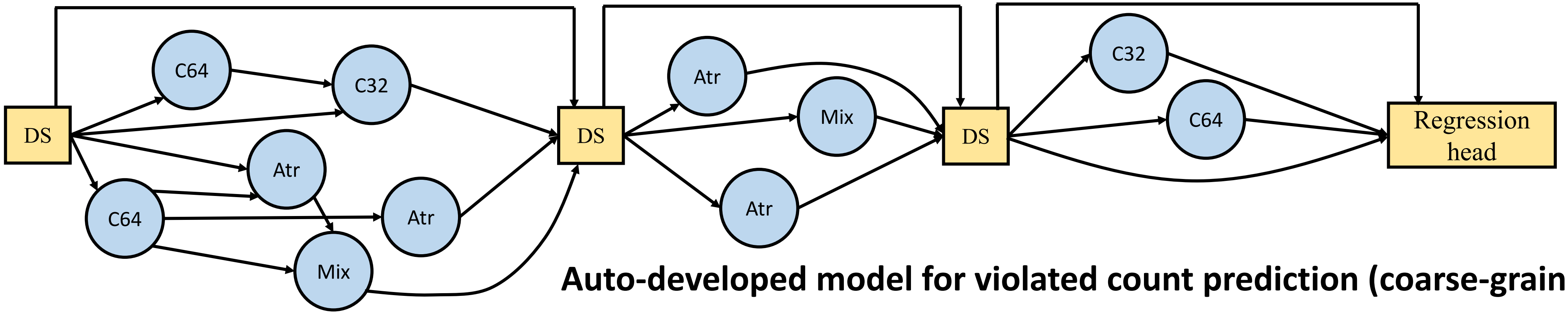
- Human-designed models:
  - Highly hierarchical and organized architecture
  - Limited operation types
- Auto-developed model:
  - Construct parallel branches and flexible interactions
  - Supports different operators



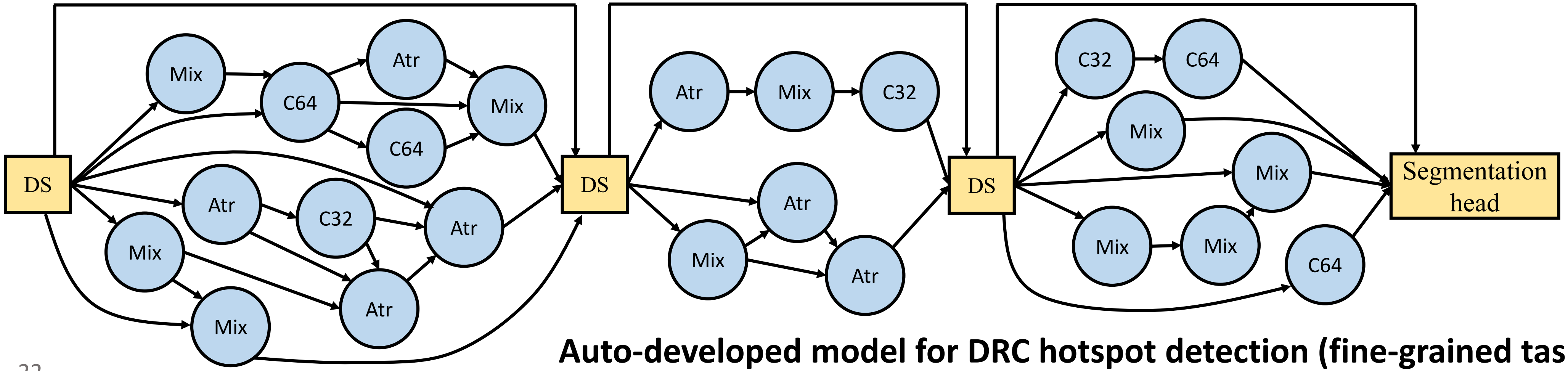
Auto-developed model for DRC hotspot detection (fine-grained task 2)

Auto-developed Model Structures

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



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

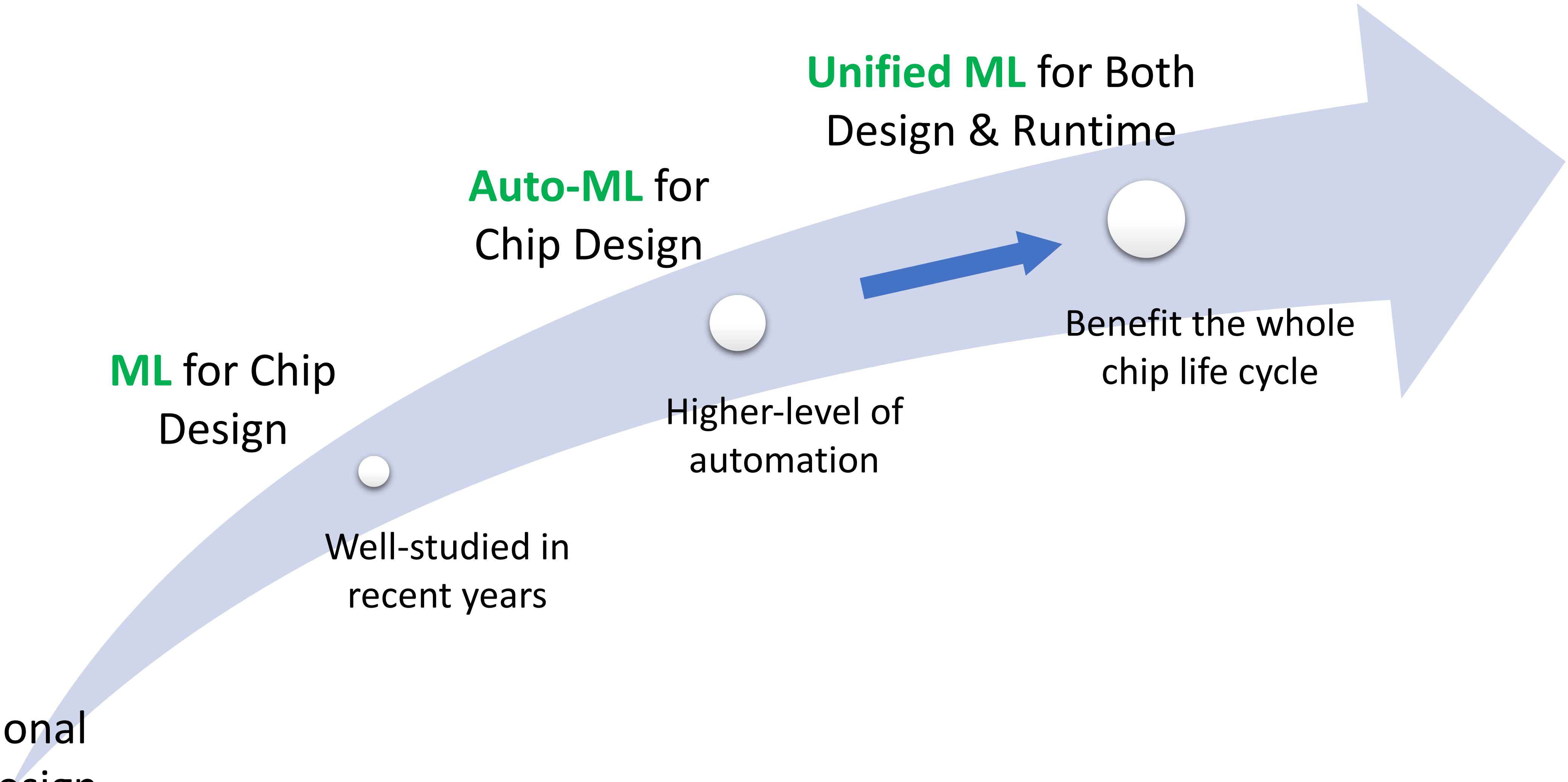


Auto-developed model for DRC hotspot detection (fine-grained task 2)

Case Study 2:

Power & Power Delivery Challenges

What I Believe We Should Target



Traditional Chip Design

ML for Chip Design

Well-studied in recent years

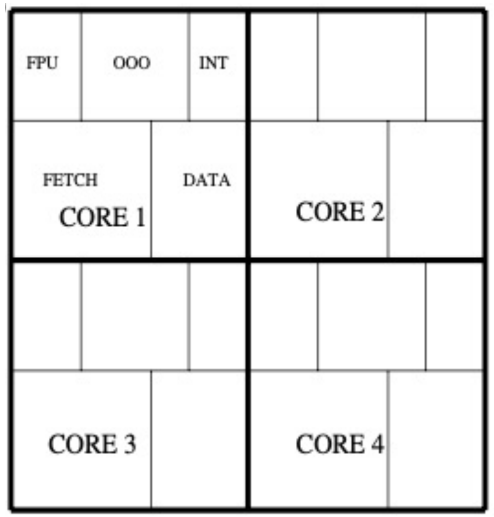
Auto-ML for Chip Design

Higher-level of automation

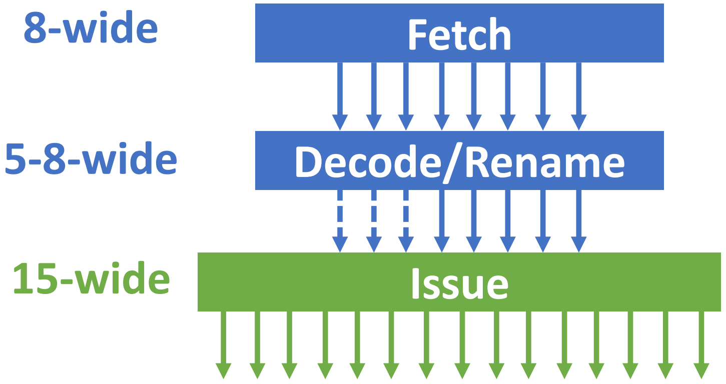
Unified ML for Both Design & Runtime

Benefit the whole chip life cycle

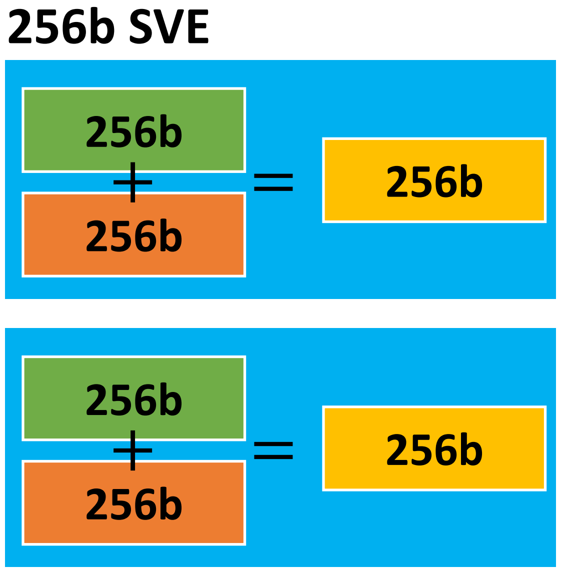
Challenge 1 – Design-time Power Introspection



Many-core CPU with more transistors



Wider issue



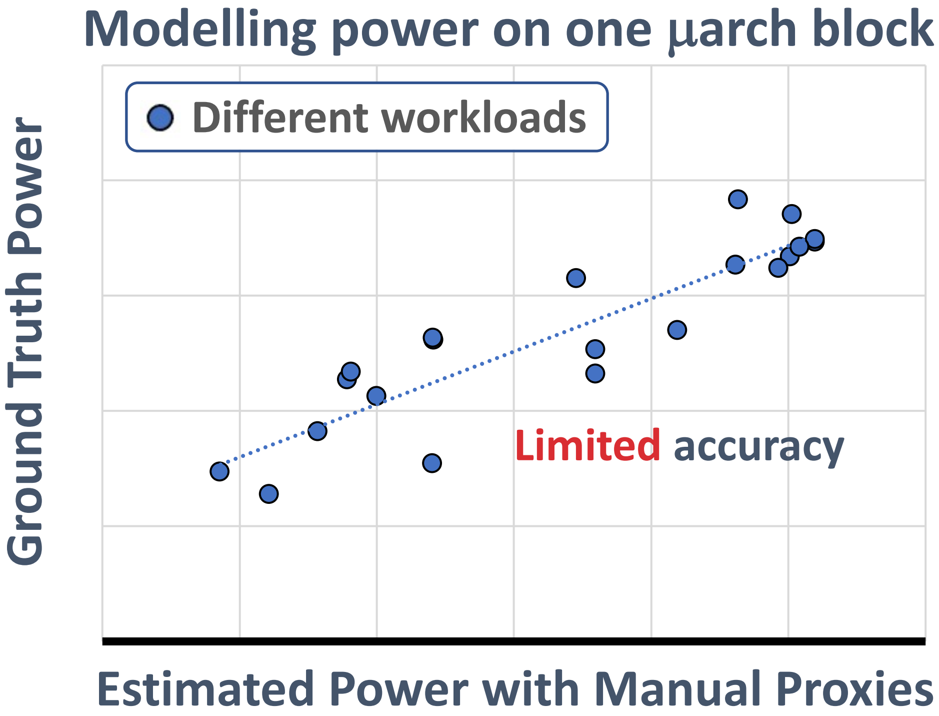
More vectored execution

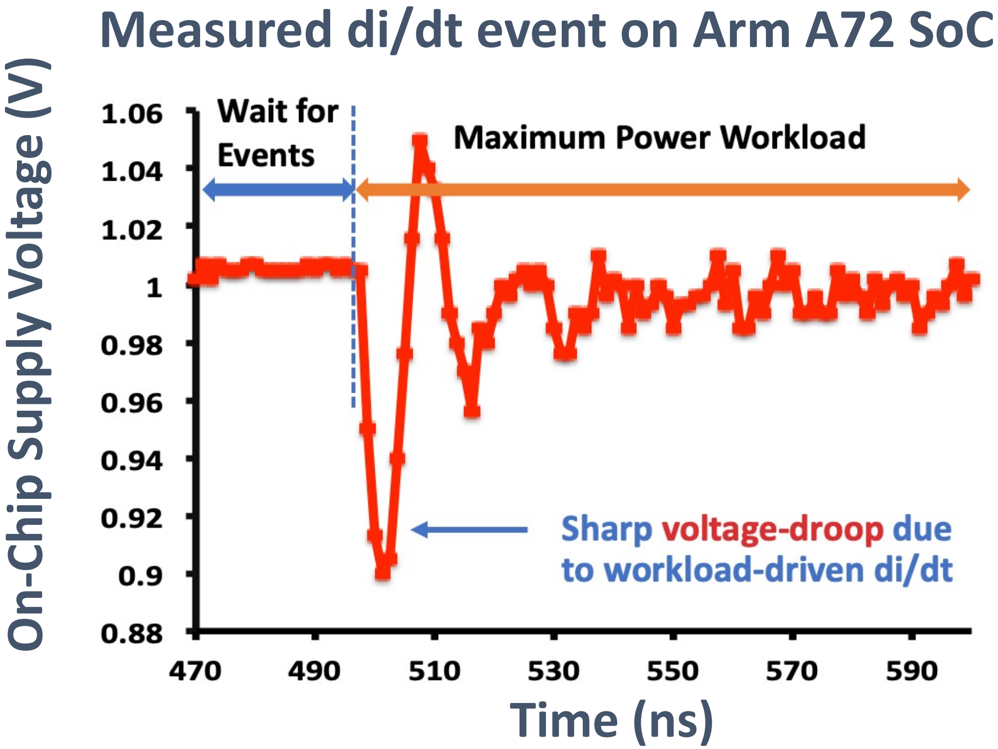
- Delivering generational performance gains **adversely impacts** CPU power
- Power-delivery resources **not keeping pace** with CPU power demands
- **Increasing power-sensitivity drives the need for design-time introspection**

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

Source: Arm Neoverse V1, 2021

Challenge 2 – Run-time Power Introspection

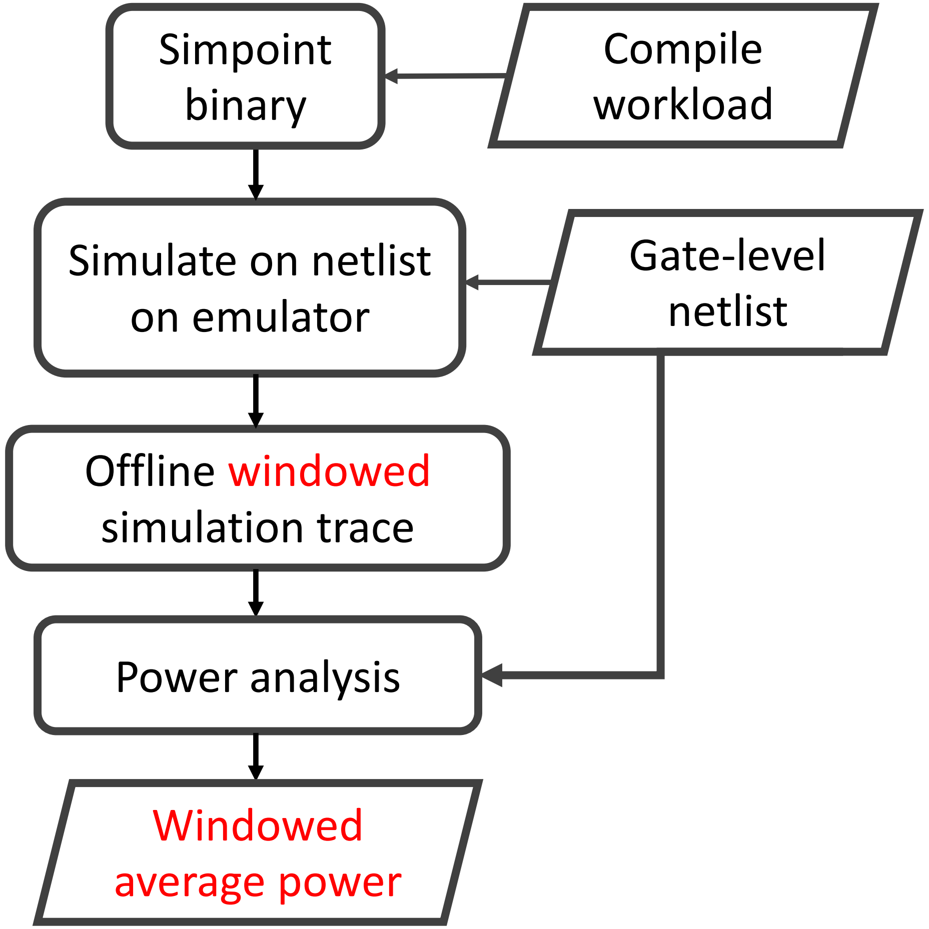




- **Peak-Power mitigation requires accurate power estimation to drive throttling**
    - Manually inferring proxies is very difficult in complex modern CPUs
- **Abrupt changes in CPU current-demand (di/dt event) leading to deep voltage-droop**

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

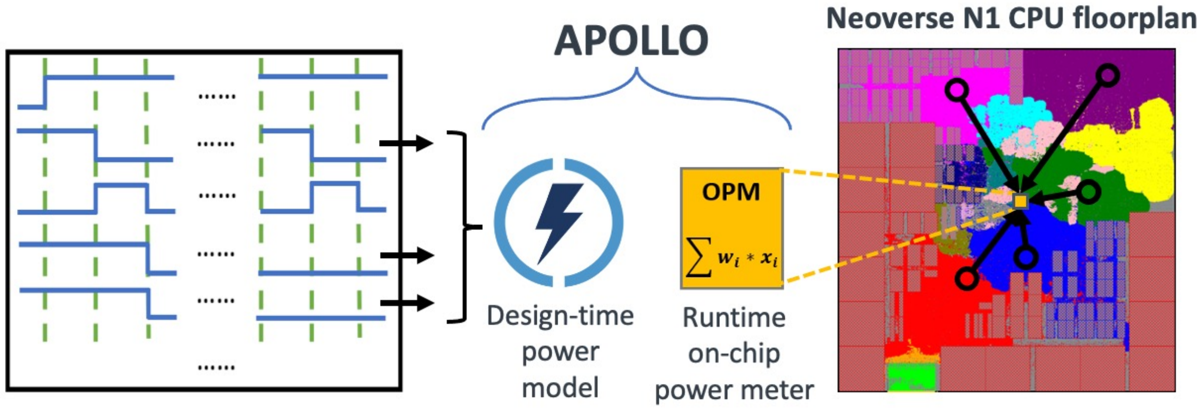
- Peak power mitigation
  - **Difficult to manually** infer proxies
- Voltage droop (Ldi/dt) mitigation
  - Require very **low** response latency

Design-time Challenges Summary

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

APOLLO: A Unified Power Modeling Framework

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



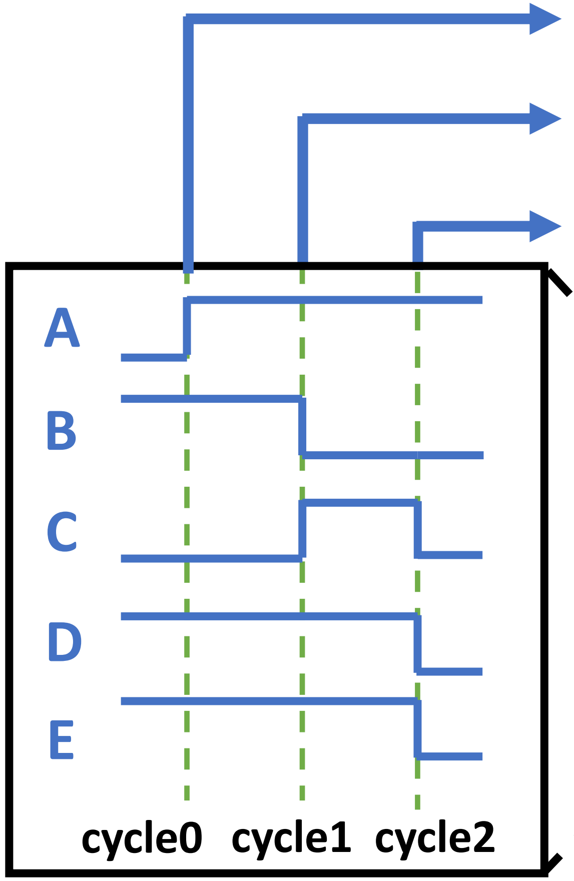
APOLLO

Neoverse N1 CPU floorplan

Design-time power model

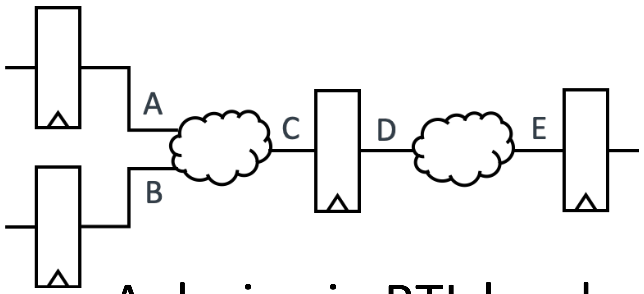
Runtime on-chip power meter

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

$$F\left(\begin{array}{c|ccccc} & A & B & C & D & E \\ \text{cycle0} & 1 & 0 & 0 & 0 & 0 \\ \text{cycle1} & 0 & 1 & 1 & 0 & 0 & \cdots\cdots \\ \text{cycle2} & 0 & 0 & 1 & 1 & 1 \\ & & & \vdots & & \end{array}\right)_{M\ \text{RTL signals}} = \begin{bmatrix} P0 \\ P1 \\ P2 \\ \vdots \end{bmatrix}$$

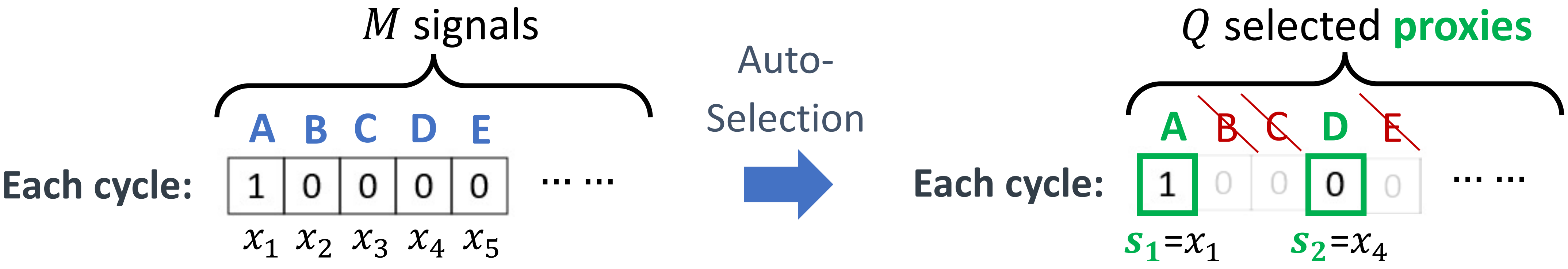
Train the ML model: $F(X) = y$

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

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

Simple Key Ideas

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



M signals

A B C D E

Each cycle: 1 0 0 0 0 … …

x_1 x_2 x_3 x_4 x_5

Auto-Selection

Q selected **proxies**

A ~~B~~ ~~C~~ D ~~E~~

Each cycle: 1 0 0 0 0 … …

$s_1 = x_1$ $s_2 = x_4$

Linear model with M RTL signals

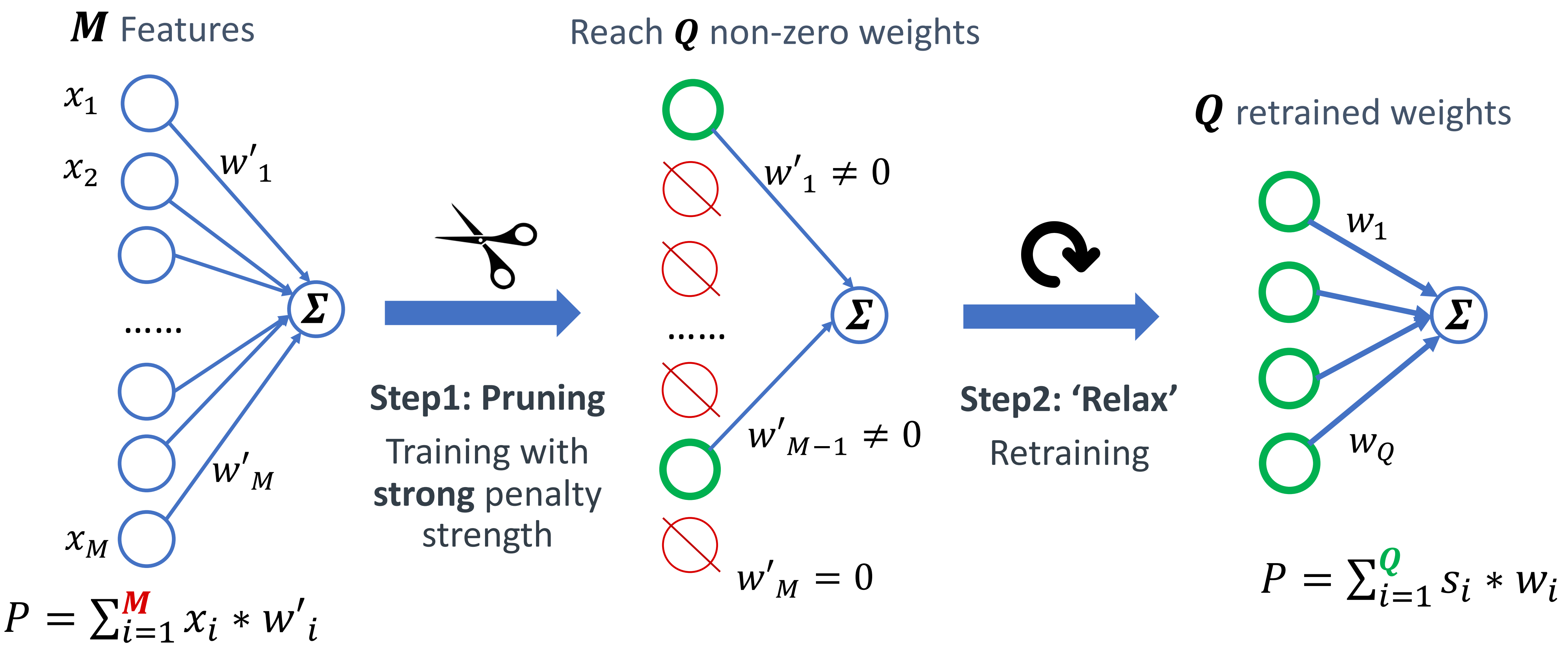
$$P = \sum_{i=1}^{M} x_i * w'_i$$

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



M Features

Reach ***Q*** non-zero weights

Q retrained weights

Step1: Pruning

Training with **strong** penalty strength

Step2: 'Relax'

Retraining

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

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

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

Model Training and Testing

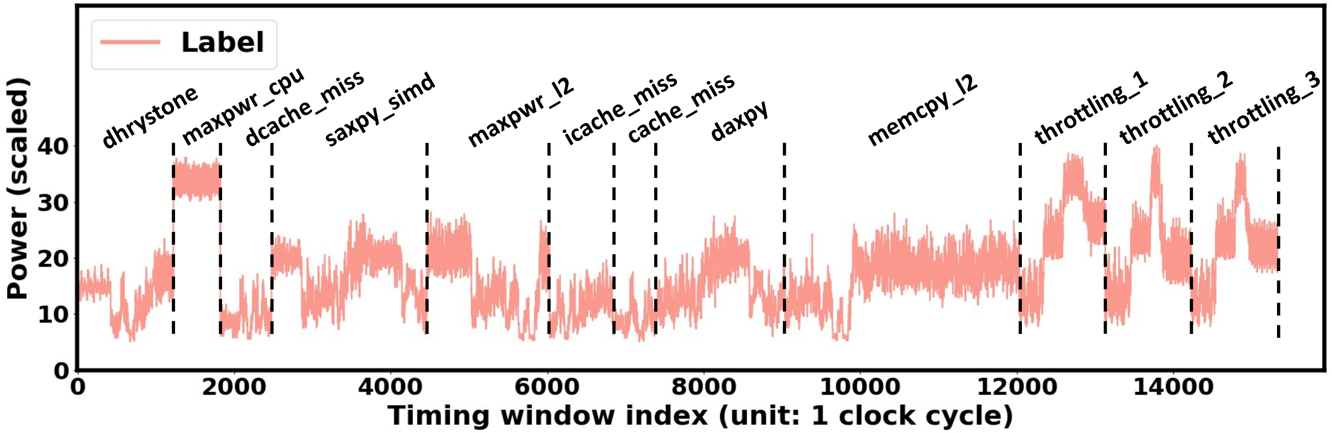
C6g

Best price performance for compute-intensive workloads

aws

Neoverse N1 (infra)
Deployed in AWS Graviton

Cortex A77 (mobile)
Deployed In Snapdragon 865

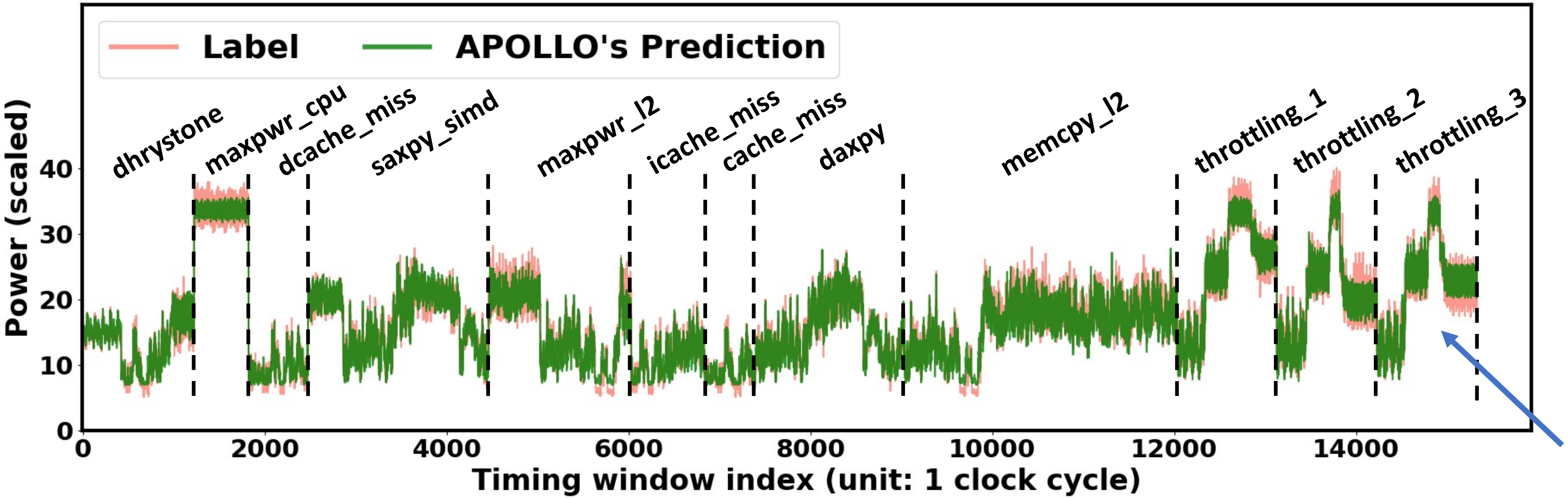


- Experiments on 3GHz 7nm Arm **commercial** microprocessors **Neoverse N1** and **Cortex A77**
- **Automatically** generate a “diverse” set of random micro-benchmarks for training
- Testing on **various** Arm power-indicative workloads

Duke
UNIVERSITY

Prediction Accuracy as Design-Time Power Model

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

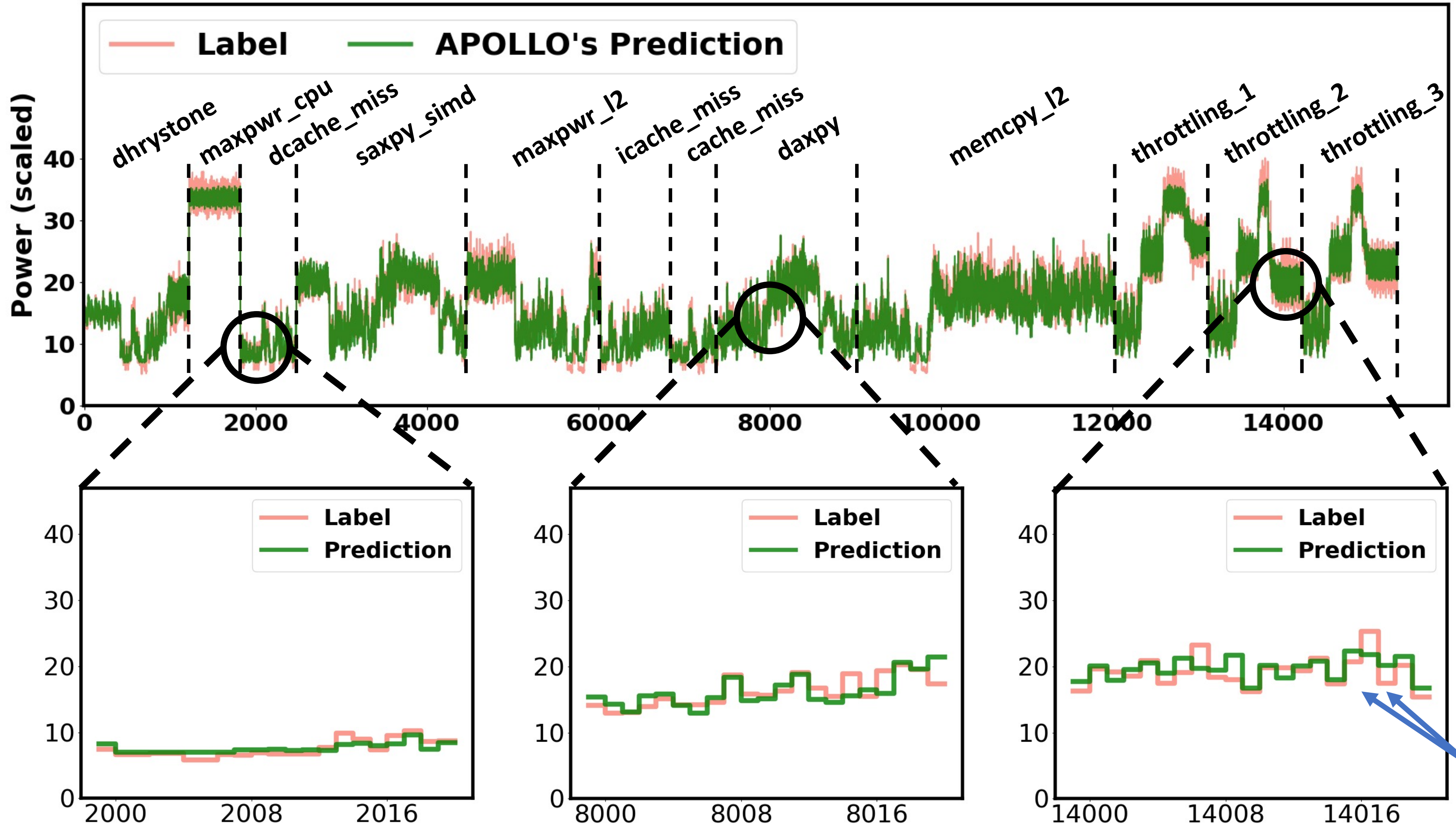


- MAE = 7.19%
- R^2 = 0.953

Prediction trace shows great agreement with ground-truth

Prediction Accuracy as Design-Time Power Model

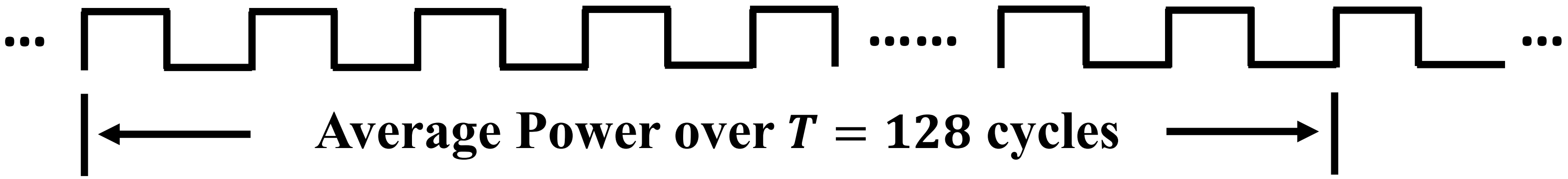
Per-cycle prediction from APOLLO with Q=159 proxies



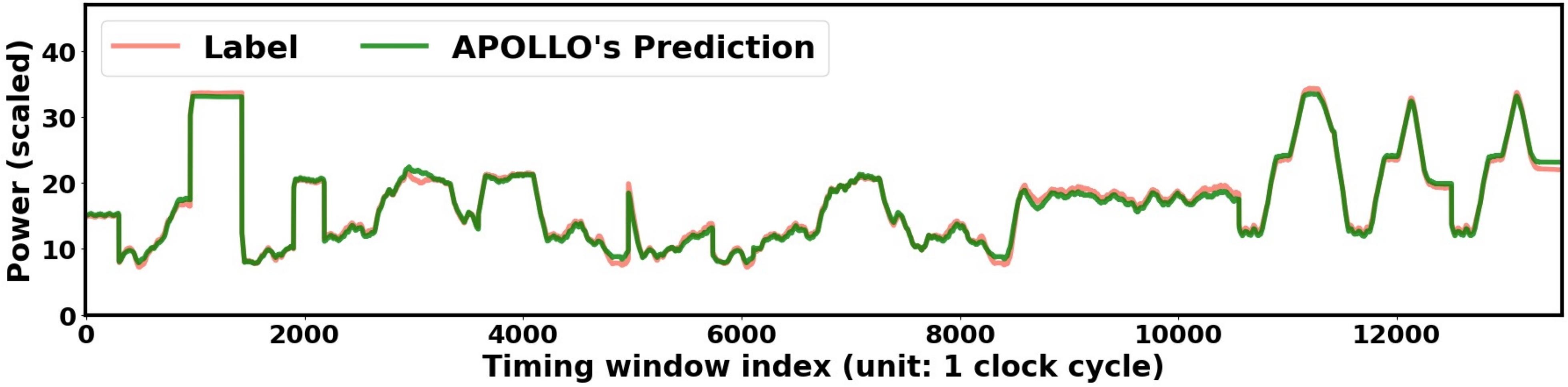
- MAE = 7.19%
- R^2 = 0.953

Per-cycle error can be averaged

Accuracy on Multi-Cycle Power Estimation



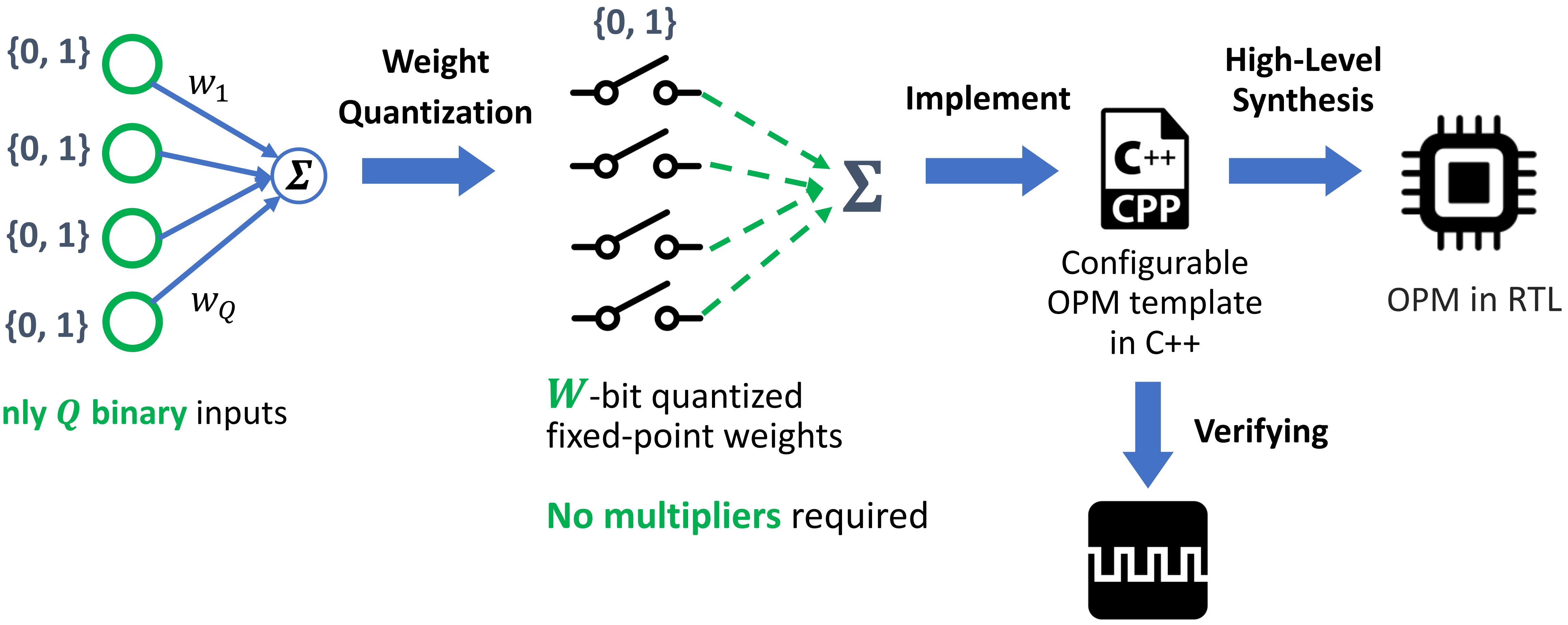
128-cycle prediction from APOLLO with ***Q*=70** proxies



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

Automated Low-Cost Runtime OPM Implementation

APOLLO is designed to be hardware-friendly



{0, 1}

{0, 1}

{0, 1}

{0, 1}

w_1

w_Q

Σ

Weight Quantization

{0, 1}

Σ

Implement

Configurable OPM template in C++

High-Level Synthesis

OPM in RTL

Verifying

Only Q binary inputs

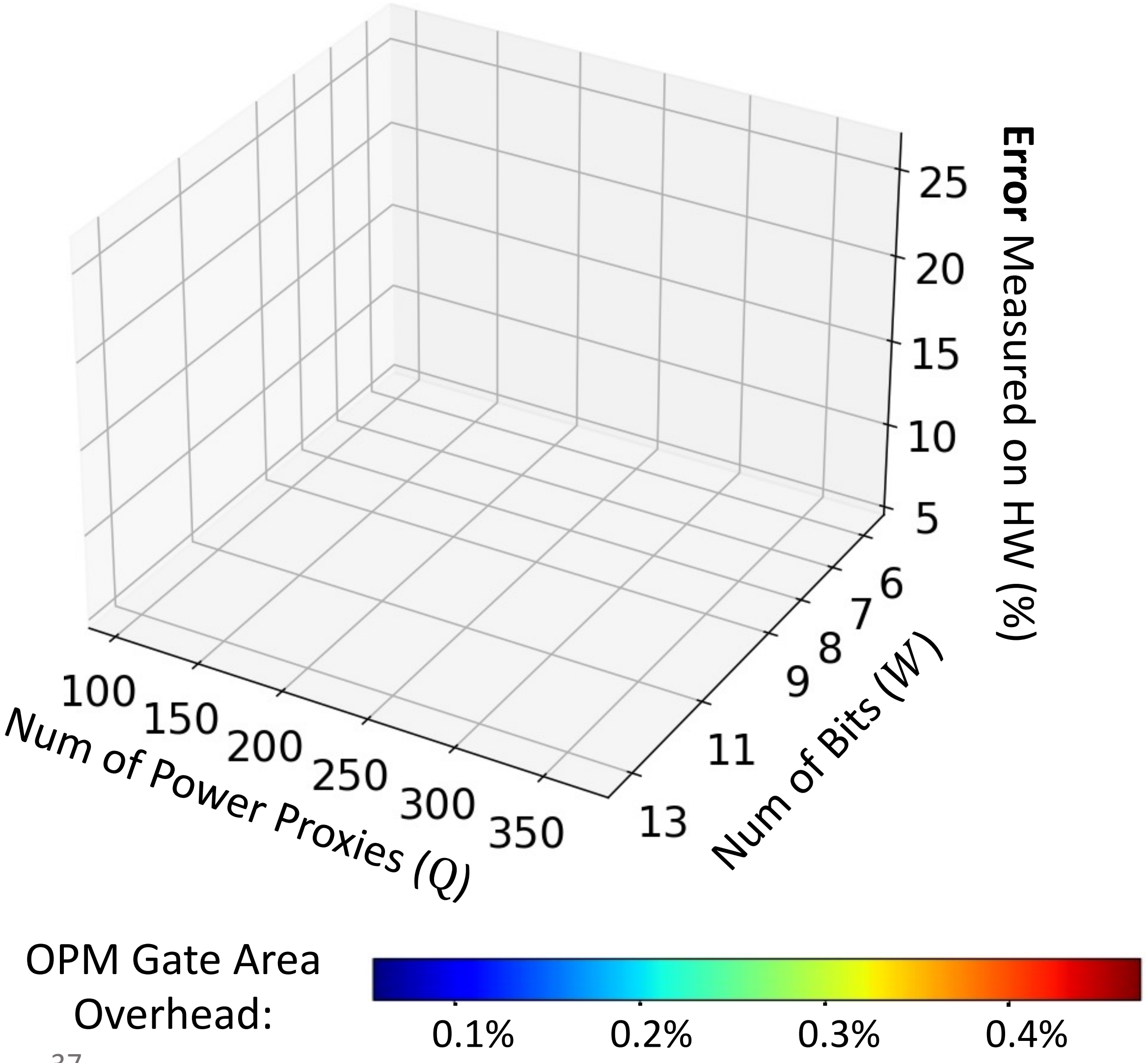
W-bit quantized fixed-point weights

No multipliers required

Verify OPM accuracy

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

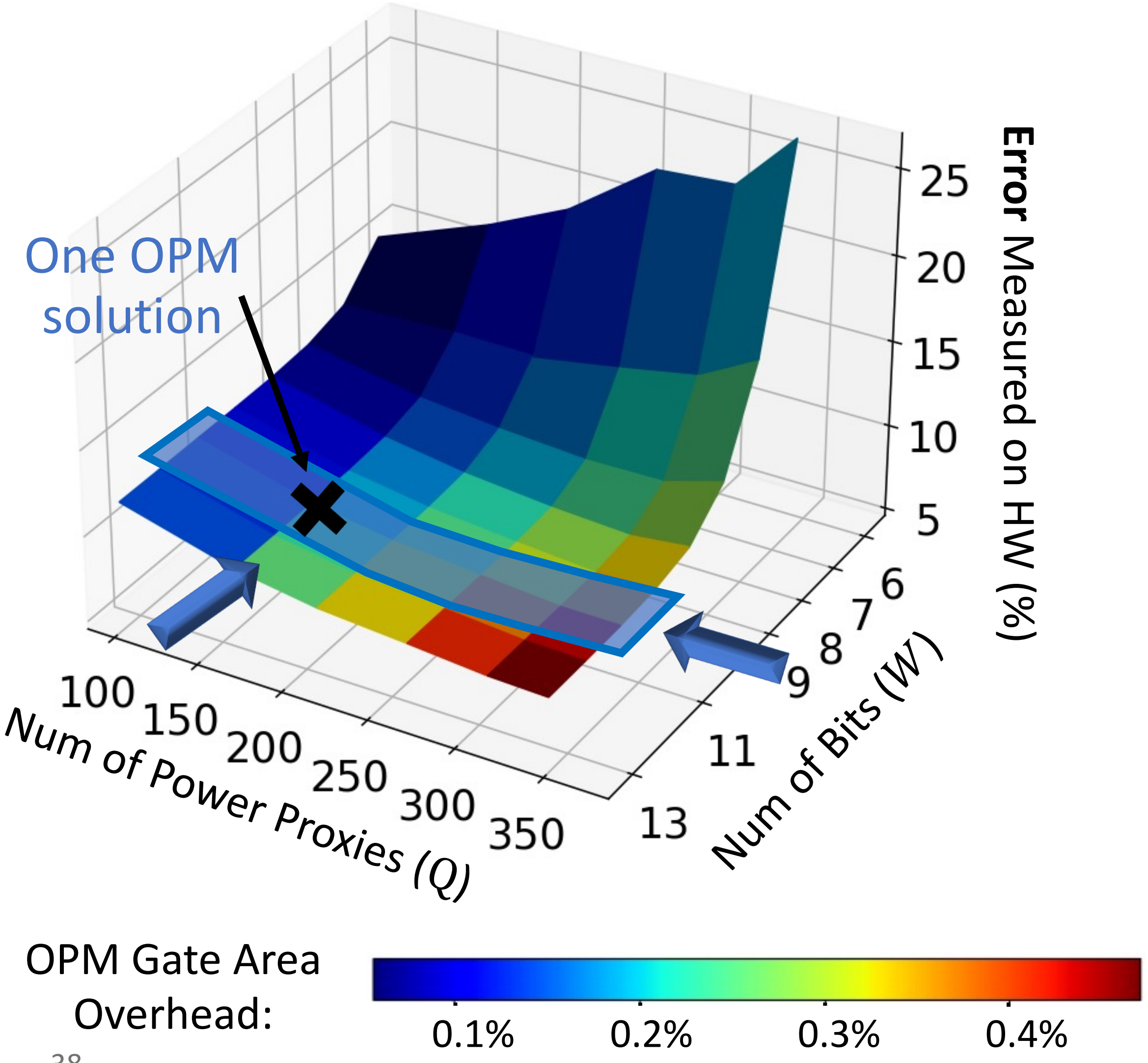
Runtime OPM implementation on Neoverse N1



- 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

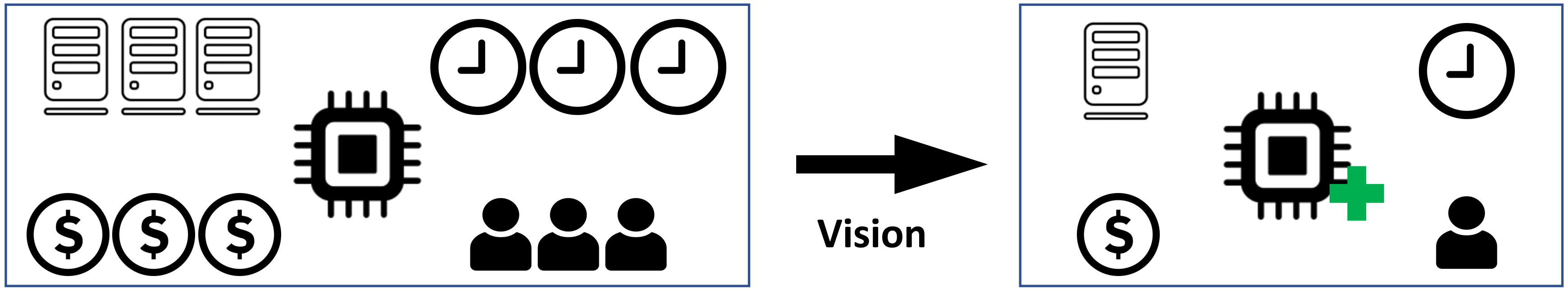
Runtime OPM implementation on Neoverse N1



- 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

Summary and Takeaway

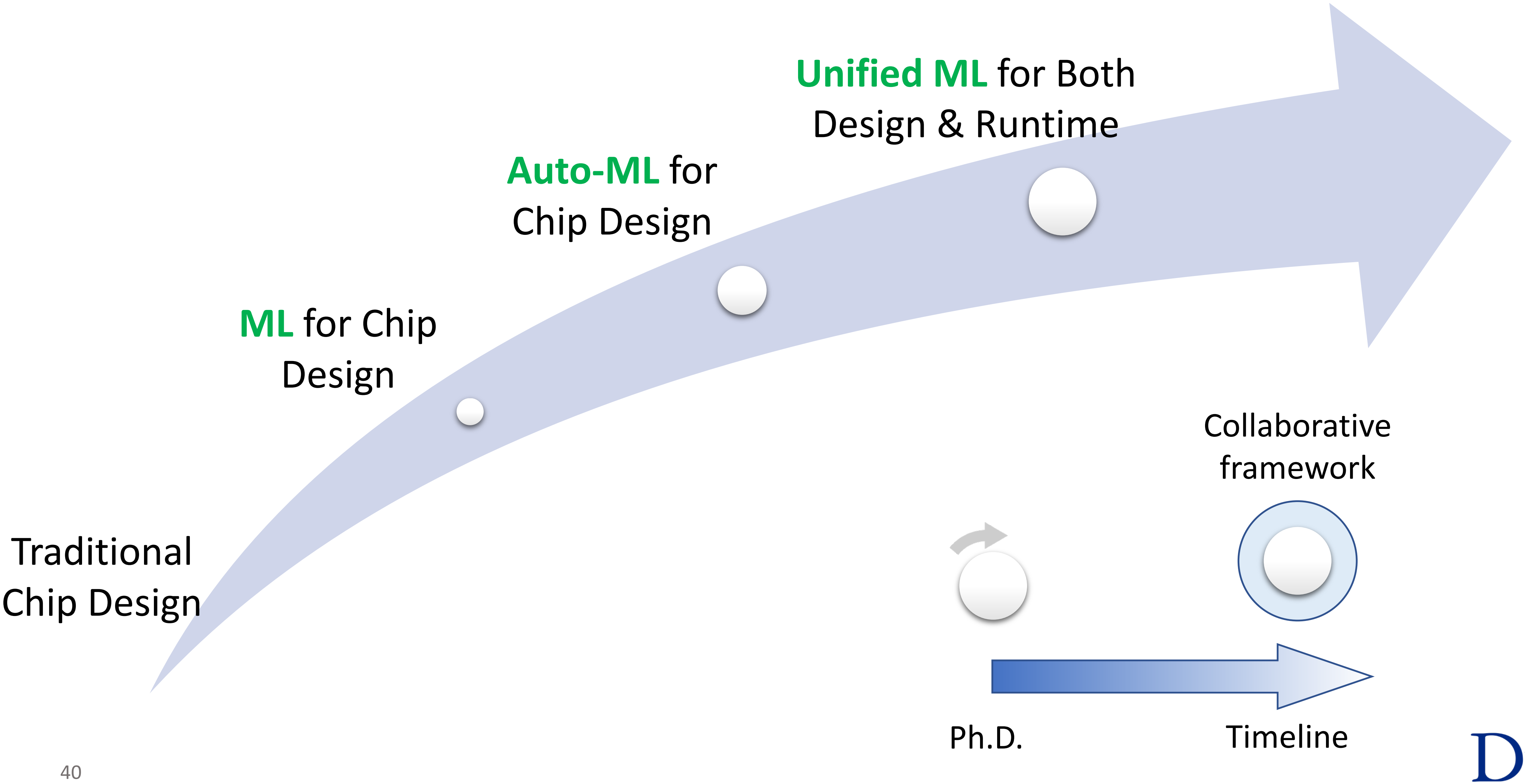
- Problem: Increasing Challenges in Chip Design
  - Cost, time-to-market, reliance on designers, diminishing performance return, ……
- **ML** in chip design
  - Less simulation time, faster feedback, less designer effort
- **AutoML** in chip design
  - Reduces months of model development to hours, no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



Chip design & implementation

Truly Intelligent Solutions

Future Research Plan



Traditional Chip Design

ML for Chip Design

Auto-ML for Chip Design

Unified ML for Both Design & Runtime

Collaborative framework

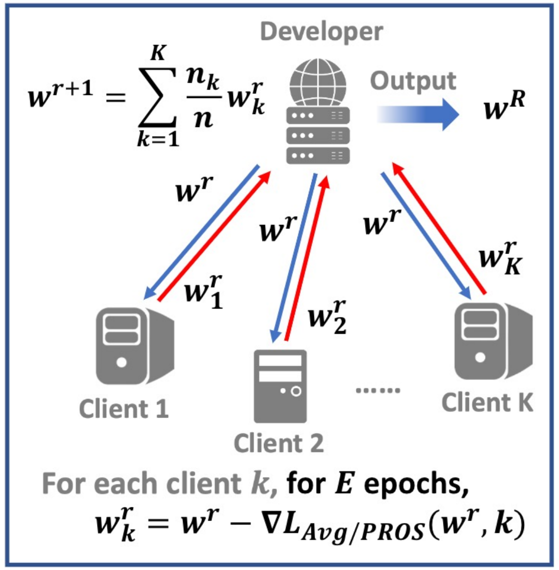
Ph.D.

Timeline

Future Works: Collaborative Framework

Collaborative ML in Chip Design

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

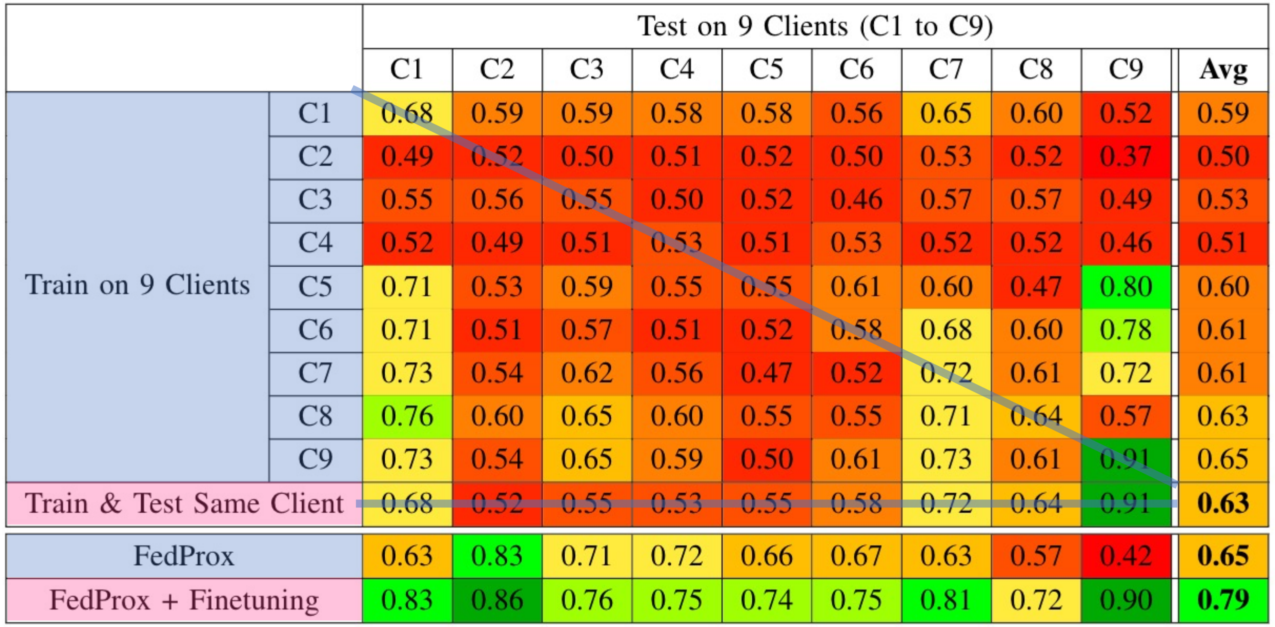

Developer
$w^{r+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^r$
Output
w^R
w^r
w^r
w^r
w_1^r
w_2^r
w_K^r
Client 1
Client 2
......
Client K
For each client k, for E epochs,
$w_k^r = w^r - \nabla L_{Avg/PROS}(w^r, k)$

Federated Learning:

Train on local data

Communicate weights

Example — Collaborative Training

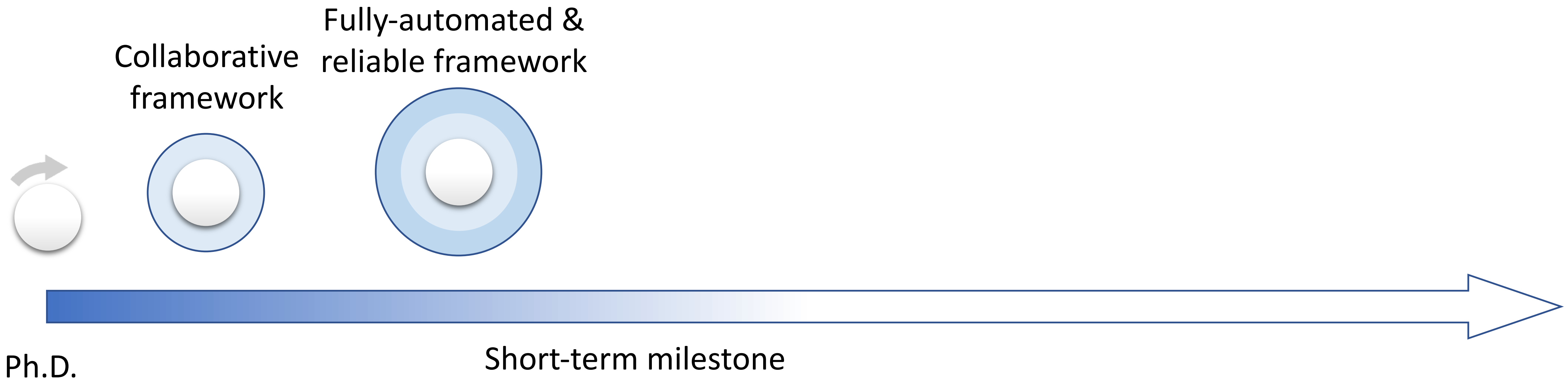


		Test on 9 Clients (C1 to C9)									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	**Avg**
Train on 9 Clients	C1	0.68	0.59	0.59	0.58	0.58	0.56	0.65	0.60	0.52	0.59
	C2	0.49	0.52	0.50	0.51	0.52	0.50	0.53	0.52	0.37	0.50
	C3	0.55	0.56	0.55	0.50	0.52	0.46	0.57	0.57	0.49	0.53
	C4	0.52	0.49	0.51	0.53	0.51	0.53	0.52	0.52	0.46	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)

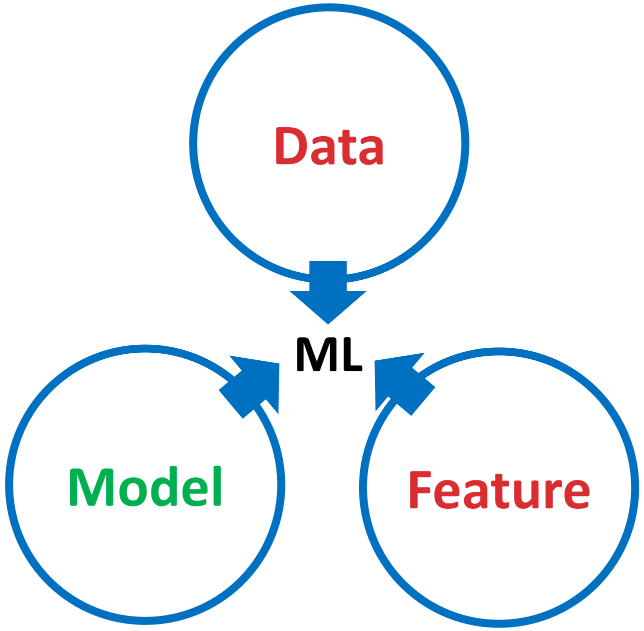
Future Research Plan



Future Works: Fully-Automated & Reliable Framework

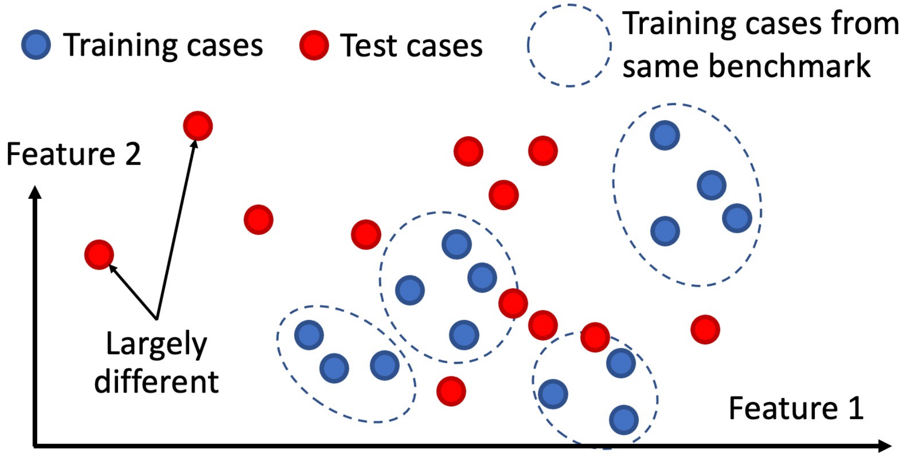
Fully-Auto ML in Chip Design

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

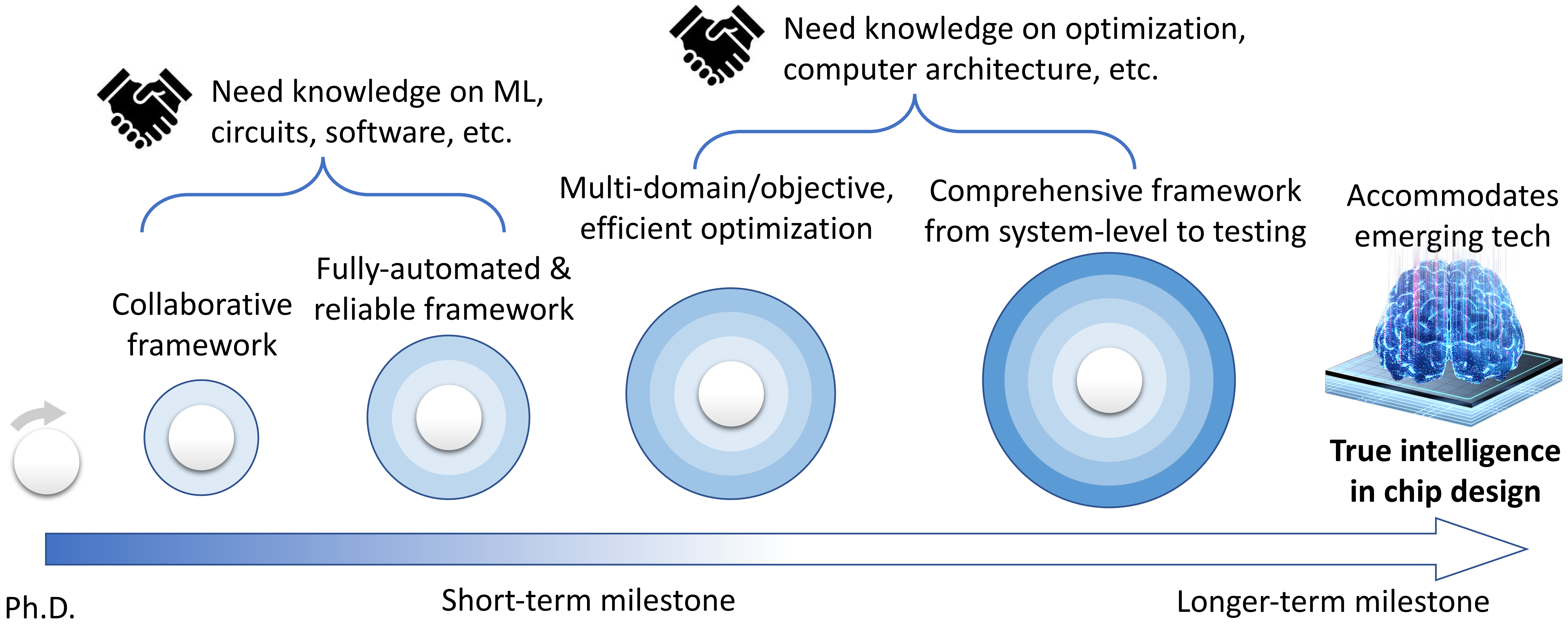


Reliable ML in Chip Design

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



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

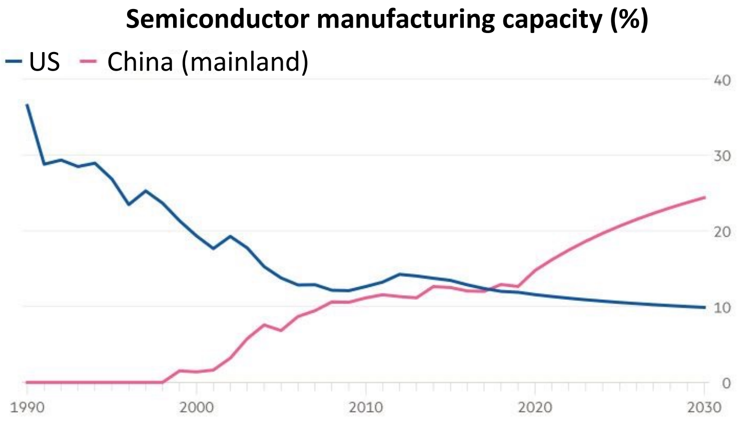
Future Funding and Collaboration Opportunities

- 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

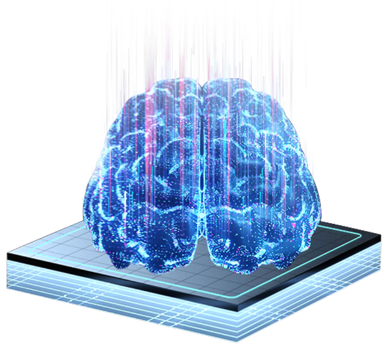
Congressional Research Service
Informing the legislative debate since 1914

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

- Computer Architecture and Circuit Courses
  - Digital VLSI design, digital integrated circuits
  - 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