arm Research

APOLLO: An Automated Power Modeling Framework for Runtime Power Introspection in High-Volume Commercial Microprocessors

Zhiyao Xie[1,2], Xiaoqing Xu[1], Matt Walker[1], Joshua Knebel[1], Kumaraguru Palaniswamy[1], Nicolas Hebert[1], Jiang Hu[3], Huanrui Yang[2], Yiran Chen[2], Shidhartha Das[1]

[1]*Arm Limited* [2]*Department of ECE, Duke University* [3]*Department of EE, Texas A&M University*

MICRO 2021 (Session 1 Best Paper Session)

TEXAS A&M UNIVERSITY

Duke UNIVERSITY

Executive Summary

Problems: High-performance features create power-delivery challenges in CPUs

Key Idea: 0.05% of RTL signals can provide enough information for power estimation in single-cycle temporal resolution

Contributions:

1. **Fast** and **accurate** design-time power model handling millions-of-cycles benchmarks in minutes
2. An unprecedented **low-cost** runtime OPM (on-chip power meter) supporting **per-cycle** power tracing
3. Fully **automated** development process for any given design

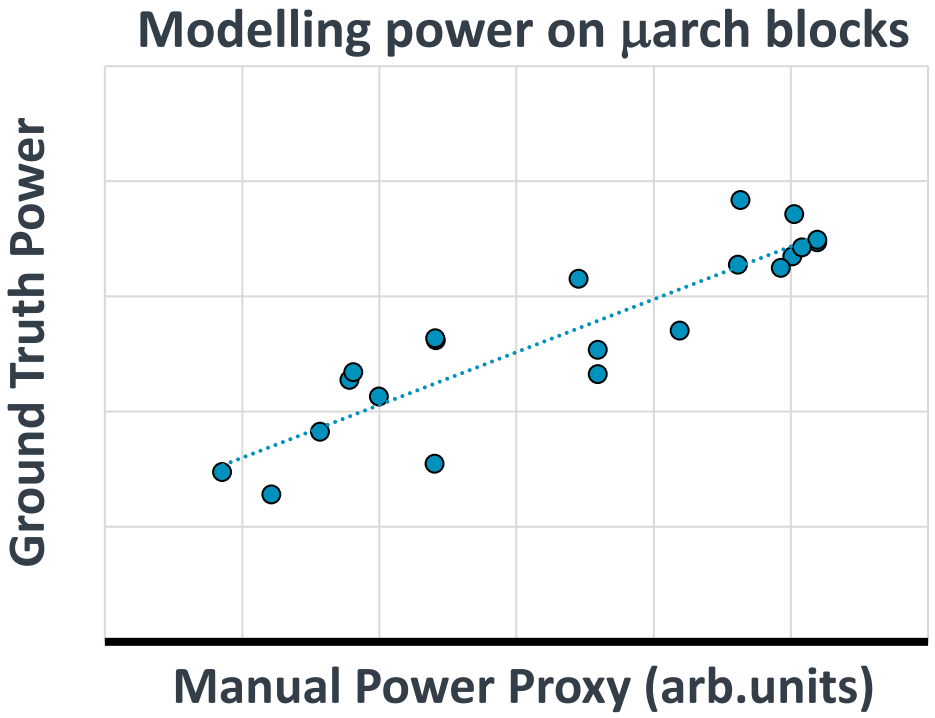
Evaluations:

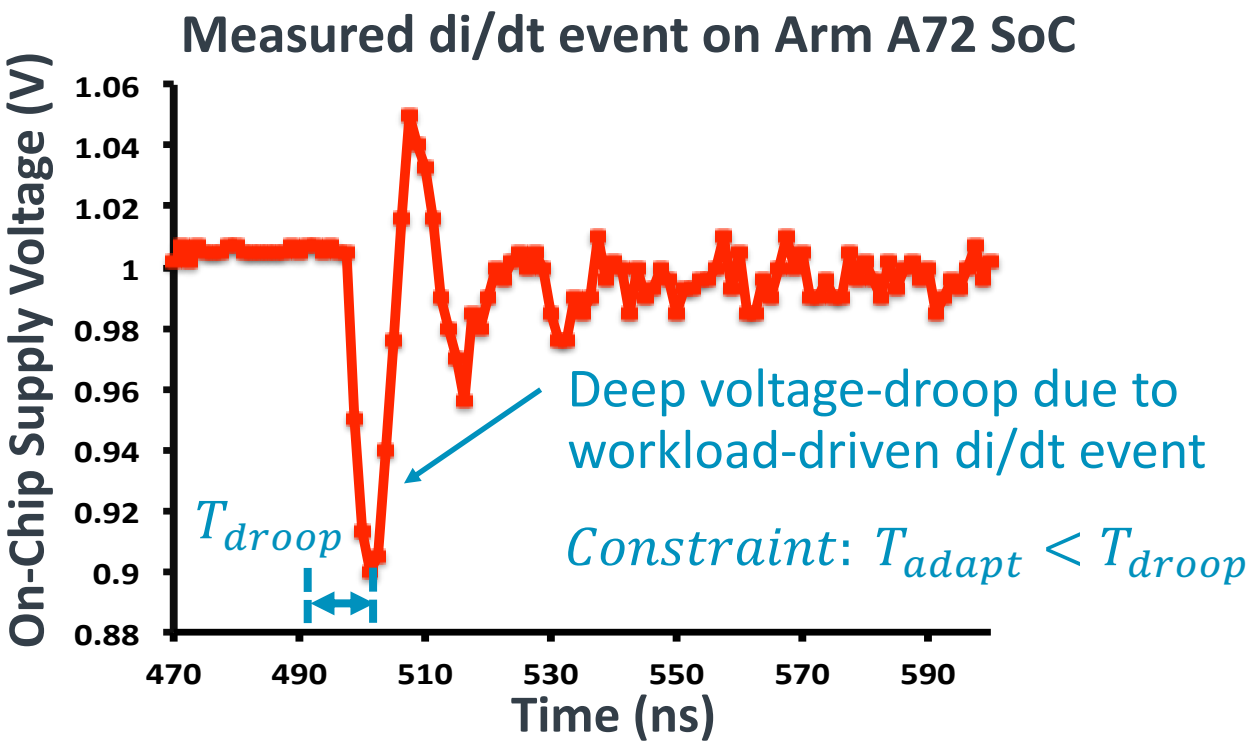
- Two high-volume 3GHz 7nm Arm microprocessors Neoverse N1, Cortex-A77
- Proven with multiple Arm power-indicative workloads

Problem 1 – Design-time CPU Power Introspection

- **Delivering generational gains in IPC and FMAX adversely impacts CPU power**
  - Diminishing returns with speculation, wide-issue and vectored execution
- **Power consumption is adversely impacted and trends upwards**
  - Efficiency gains through Moore's law scaling has effectively stalled
  - Parallel execution and greater transistor integration => increased switching activities
- **Power-delivery resources not keeping pace with CPU power demands**
  - Resistive interconnects at scaled technology nodes => greater sensitivity to peak-power
  - Package technology unable to sustain di/dt demands
- **Increasing power-sensitivity drives the need for design-time introspection**

Problem 2 – Run-time Introspection



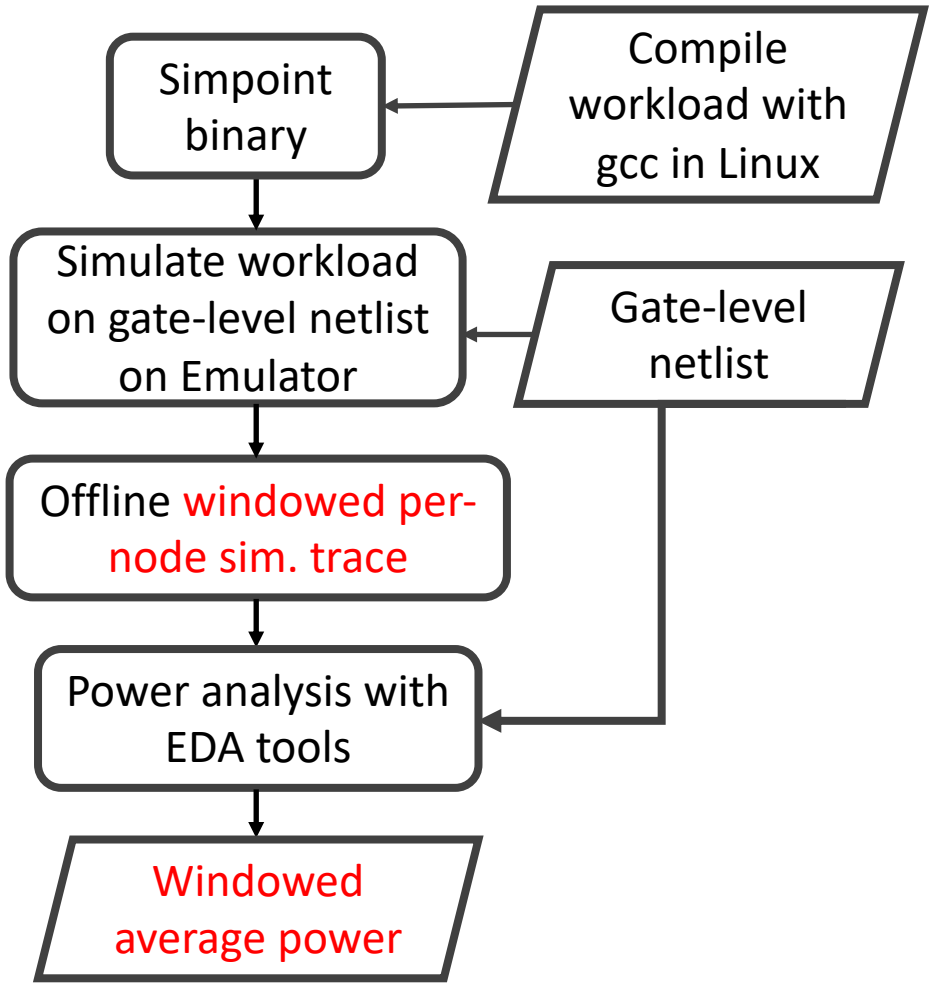


- **Peak-Power mitigation requires accurate power-estimation to drive throttling decisions**
  - Manually inferring proxies is difficult, particularly in modern CPUs with complex underlying µarch
- **Micro-architectural interactions (branch-mispredicts, ROB issue, hit-after-miss) trigger abrupt changes in CPU current-demand leading voltage-droop due to di/dt events**

arm

Problem 3 - Workload Power Characterization

- **Dimensioning power-delivery networks requires power-characterization of real-world workloads**
  - Simple micro-benchmarks no longer sufficient
- **Single SPEC simpoint executed on CPU can take weeks on emulator – an expensive resource**
  - Signoff-level power-measurement quality is expensive
- **Extends only to windowed-average power consumption**
  - Impossible to scale to Ldi/dt analysis

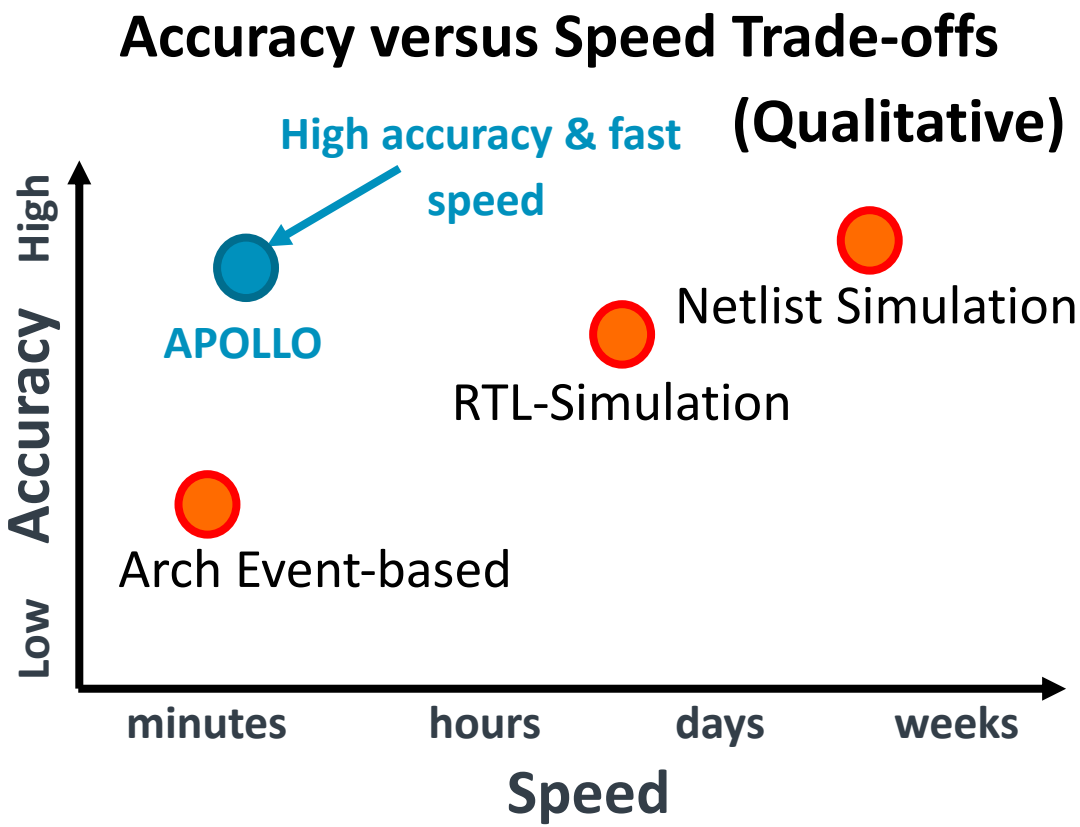


Industry-Standard Emulator-Driven Power Flow

arm

APOLLO – Key Objectives and Attributes

Problem: Prior art suffers from stark trade-offs between accuracy and speed



Automated Power-Proxy Extraction

- Use ML techniques to identify correlated events

Fast, yet accurate on-chip metering

- Proven on commercial CPUs with >95% accuracy
- 0.2% area overhead over Neoverse N1 core

Per-cycle temporal resolution

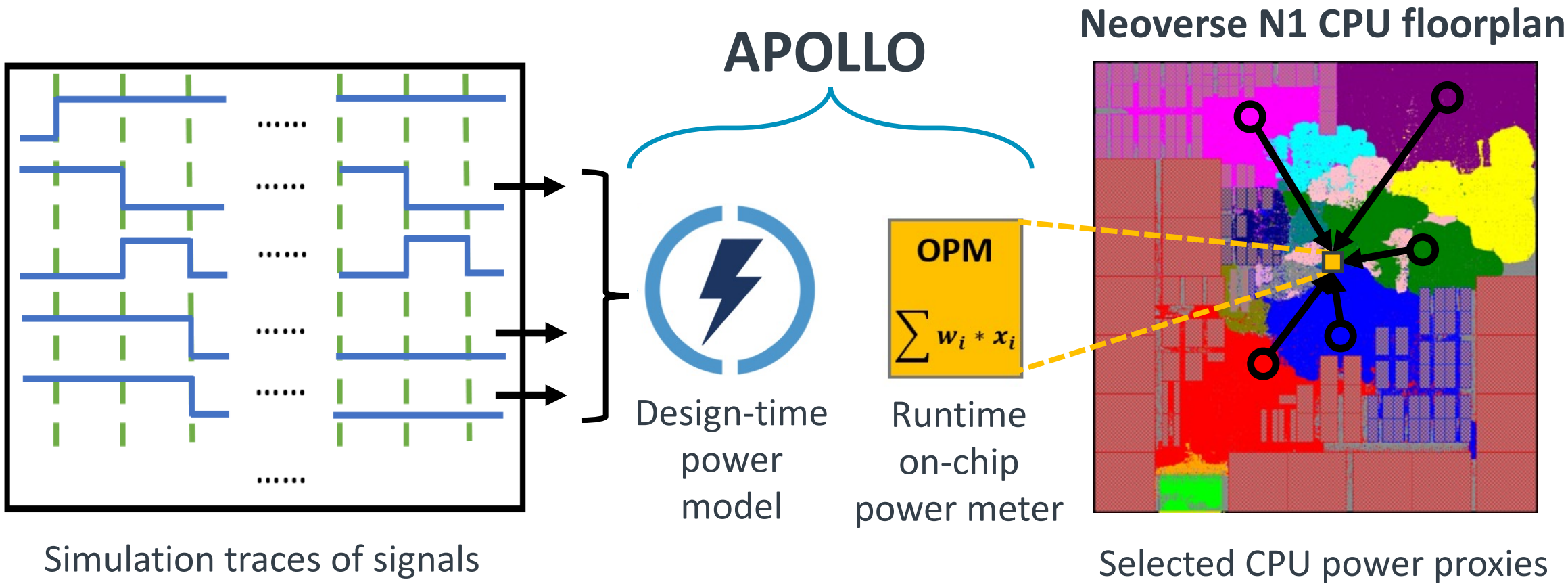
- Unify simulation, Ldi/dt mitigation, emulator-tracing within the same framework

Extensible to higher abstraction simulation

- Trade-off accuracy for pre-identified events

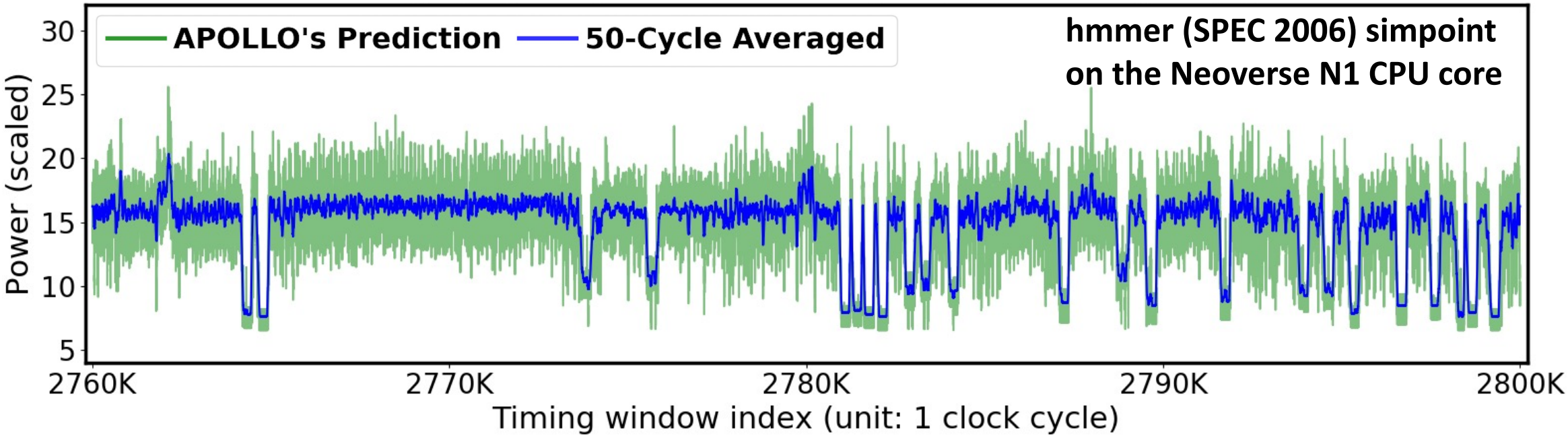
arm

APOLLO Includes Design-time Model and Runtime OPM



A Workload Execution Preview of APOLLO

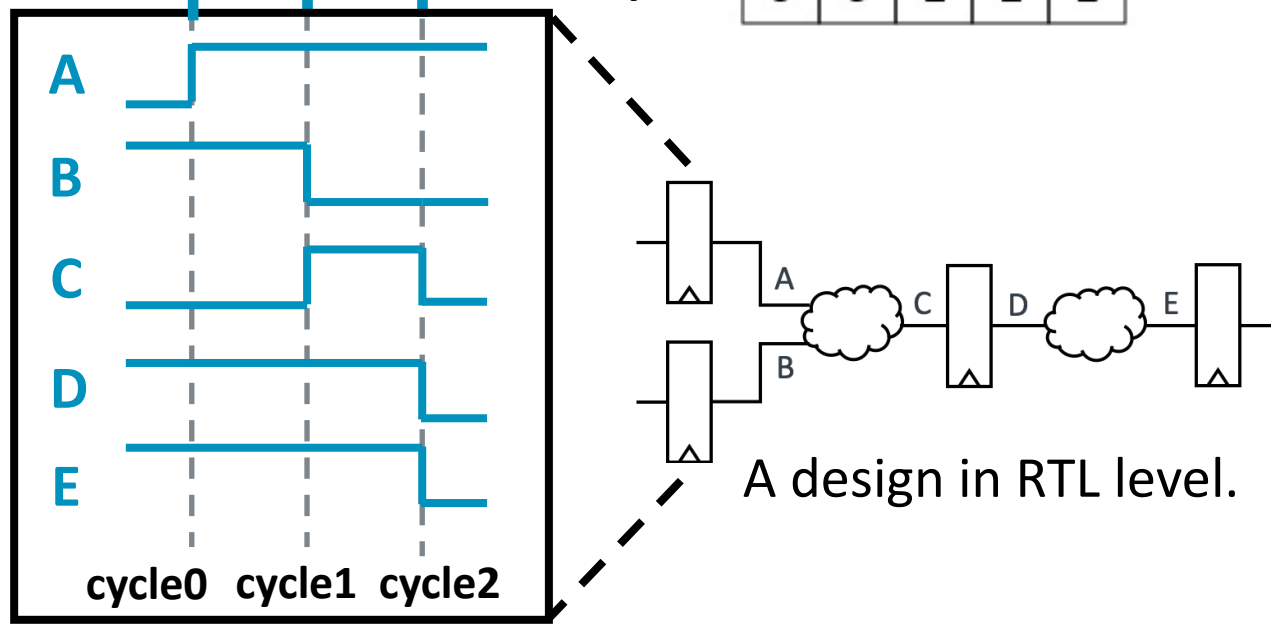
40K cycles of APOLLO Power Estimation out of a trace of 17M cycles



- **~2 weeks execution time reduced to few minutes on the emulator**
  - Instead of full netlist emulation, only the RTL is emulated
  - Storage requirement reduced 100x to proxies only (150 in this example)
- **Unprecedented power-introspection due to single-cycle temporal resolution**

arm

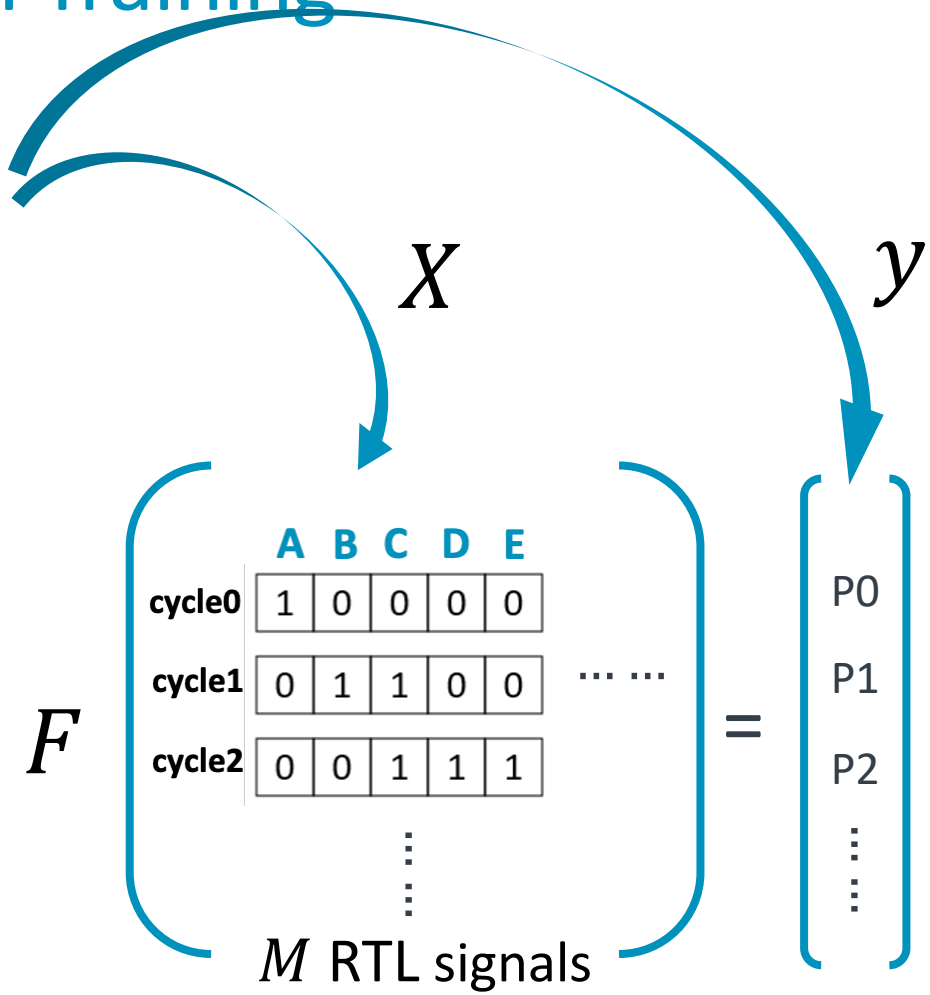
APOLLO Feature Generation & Model Training



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.



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

M > 500,000 in Neoverse N1

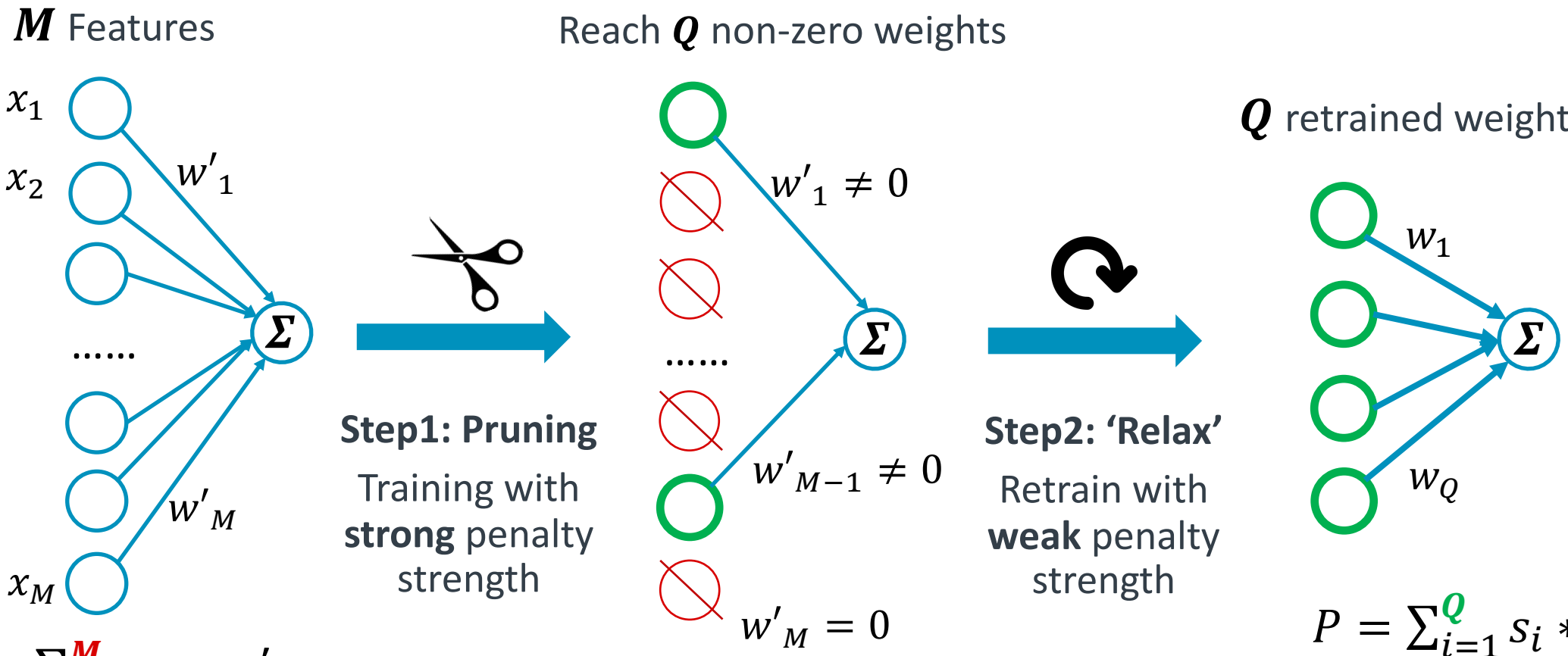
M > 1,000,000 in Cortex-A77

arm

ML-Based Power Proxies Selection

Model construction in two steps

Please check our paper for detailed discussion on MCP method



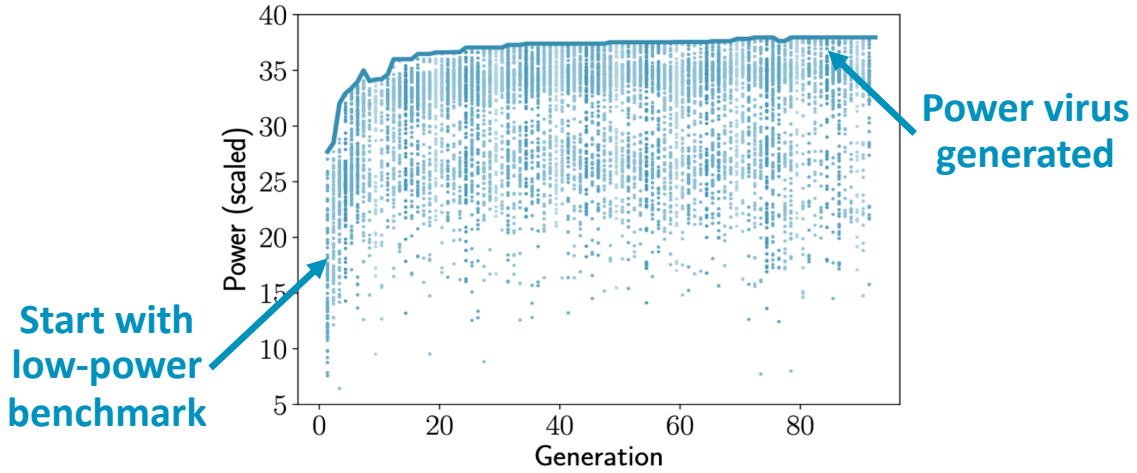
Minimax concave penalty (MCP) for pruning

Our Proposed Power Modeling Approach

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

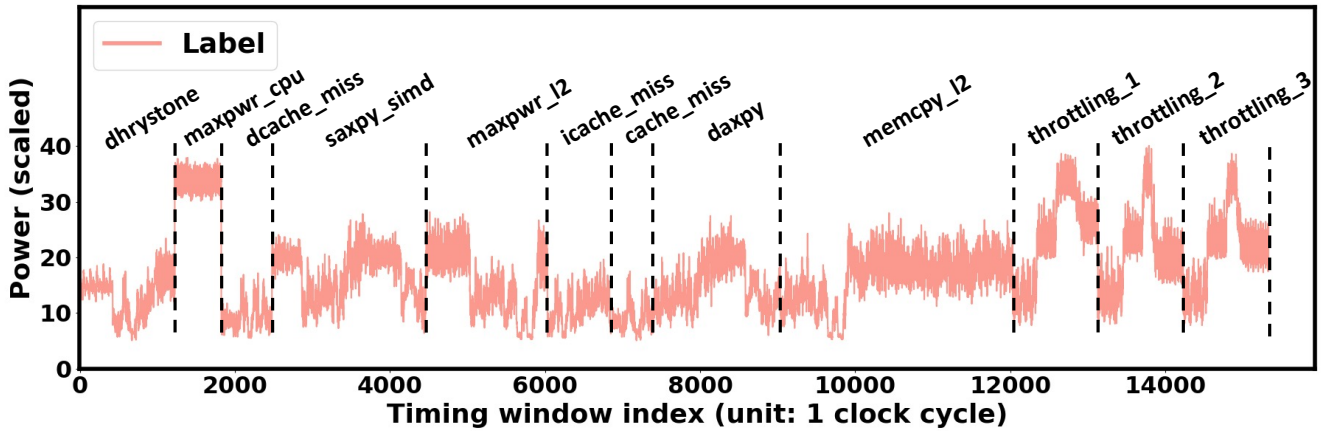
Training data automatically generated

- 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



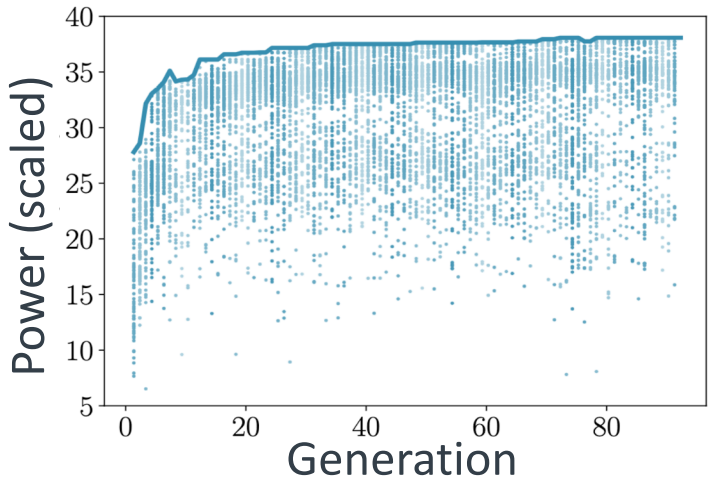
Model training & testing

- 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



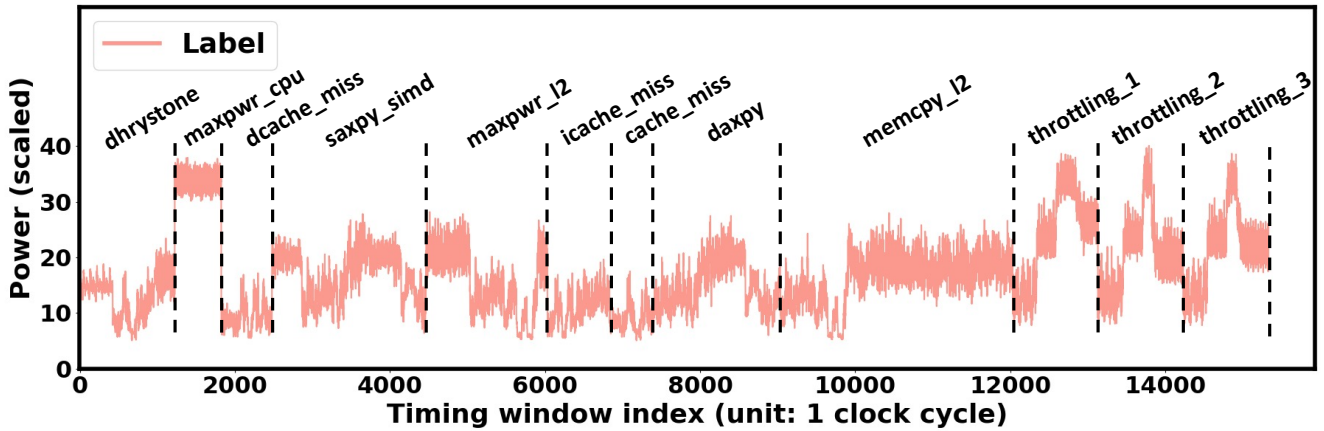
Our Proposed Power Modeling Approach

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



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

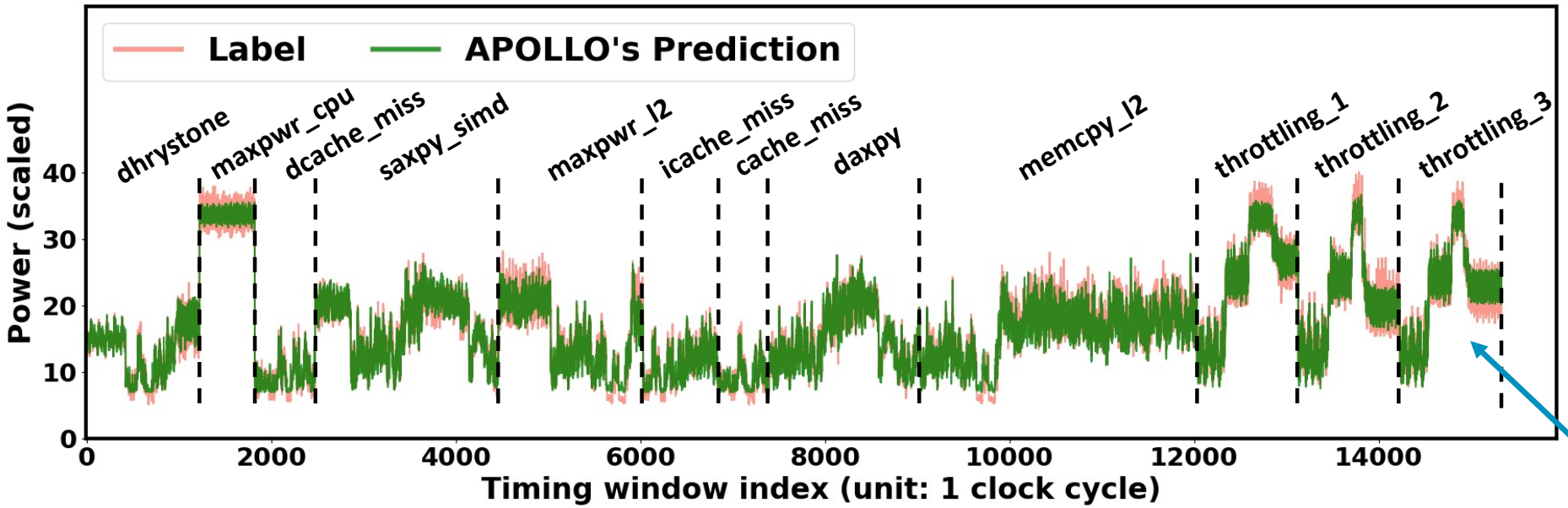


arm

Prediction Accuracy as Design-Time Power Model

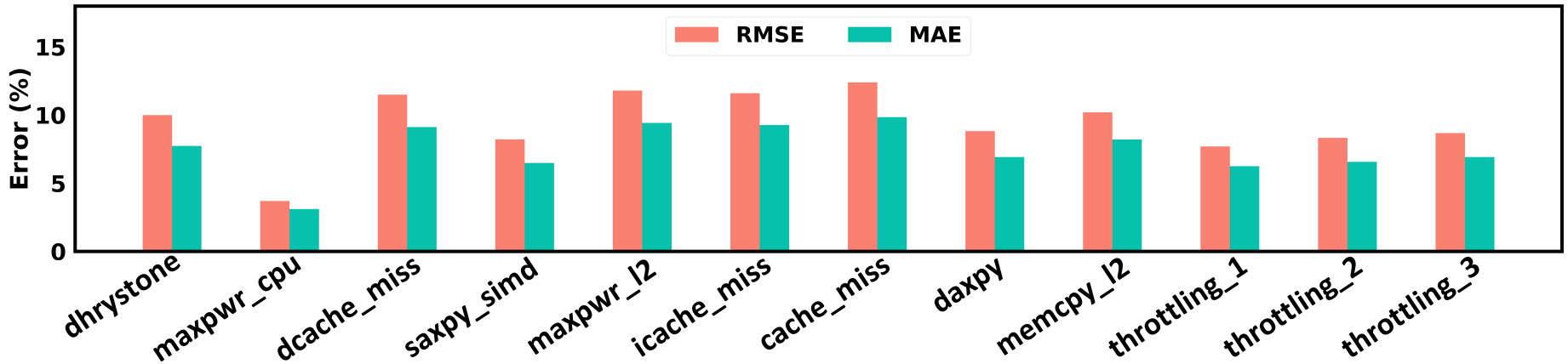
Please check our paper for detailed comparisons with baselines

Per-cycle prediction from APOLLO with Q=159 proxies



- MAE = 7.19%
- RMSE = 9.13%
- R^2 = 0.953

Prediction trace has excellent agreement with ground-truth envelope

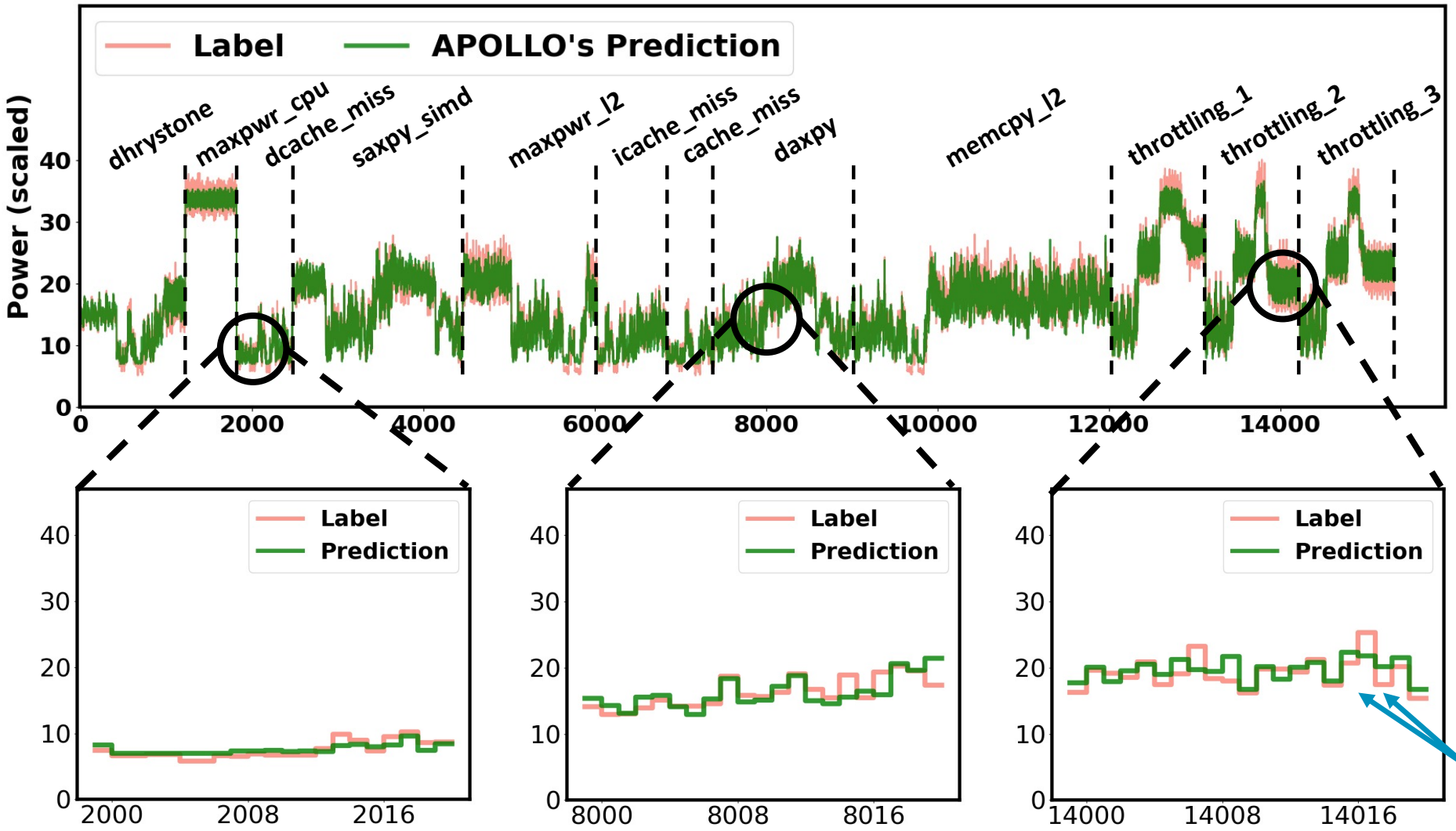


- MAE < 10% for all workloads

arm

Prediction Accuracy as Design-Time Power Model

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



- MAE = 7.19%
- RMSE = 9.13%
- R^2 = 0.953

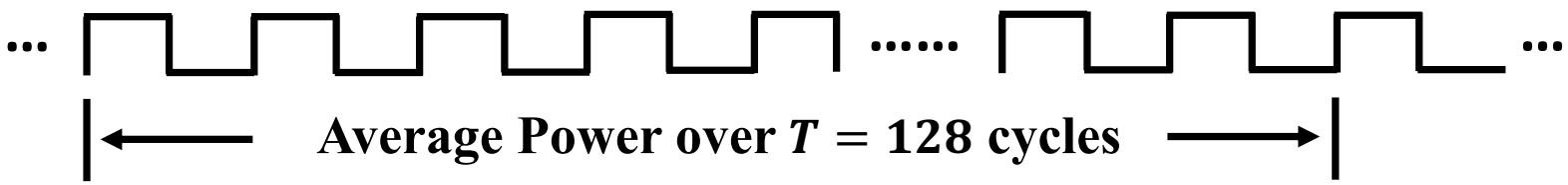
Per-cycle error can be averaged

arm

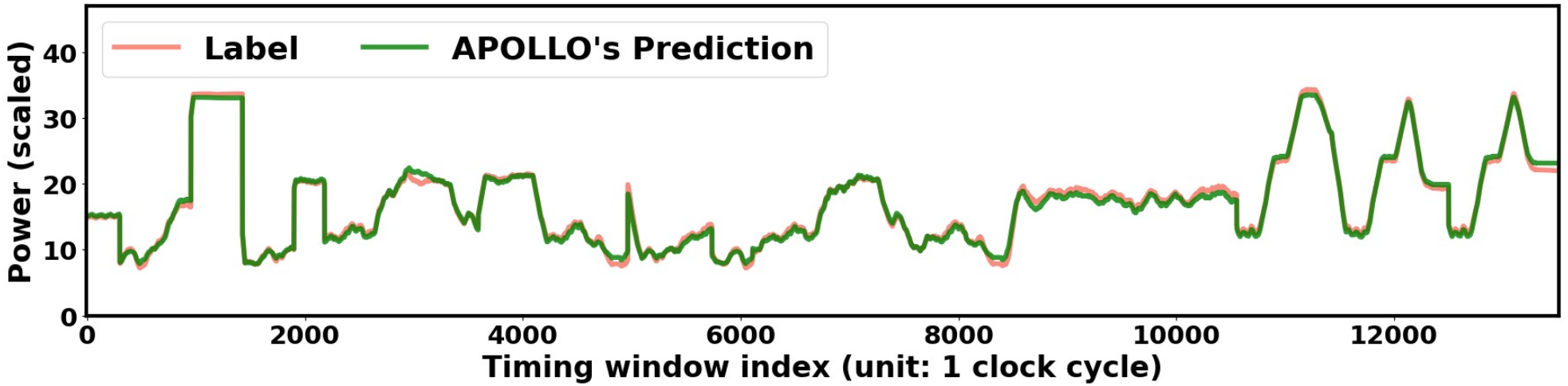
Accuracy on Multi-Cycle Power Estimation

Please check our paper for detailed multi-cycle APOLLO methods & evaluations

APOLLO accommodates any measurement window



128-cycle prediction from APOLLO with Q=70 proxies

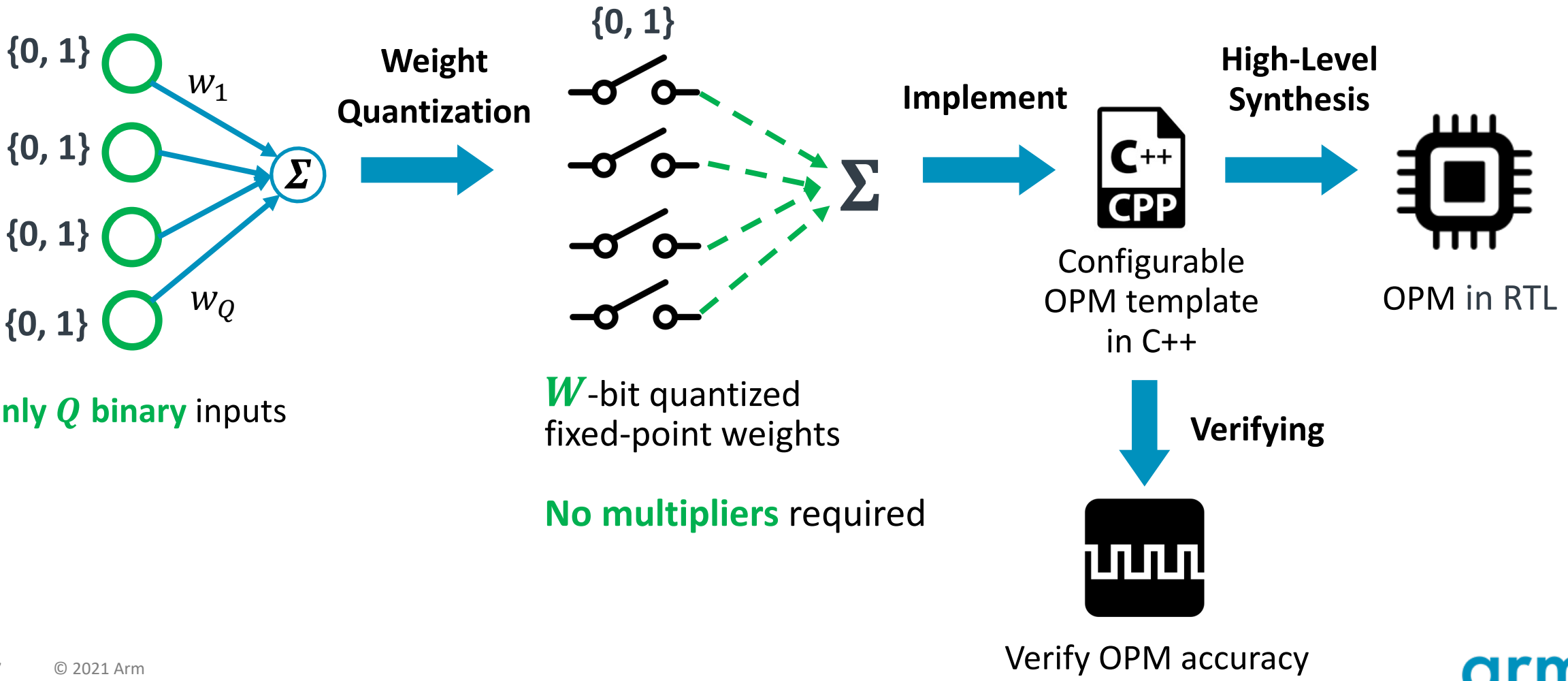


- MAE = 2.82%
- RMSE = 3.93%
- R^2 = 0.993
- Higher accuracy

Automated Low-Cost Runtime OPM Implementation

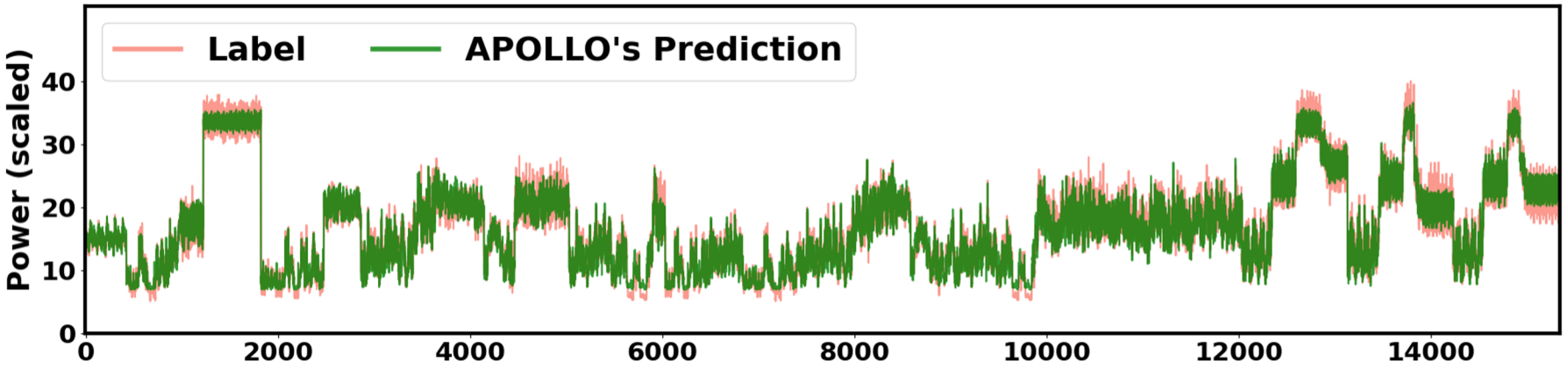
APOLLO is designed to be hardware-friendly

Please check our paper for detailed OPM implementation



Prediction Accuracy from Design-time Model & OPM

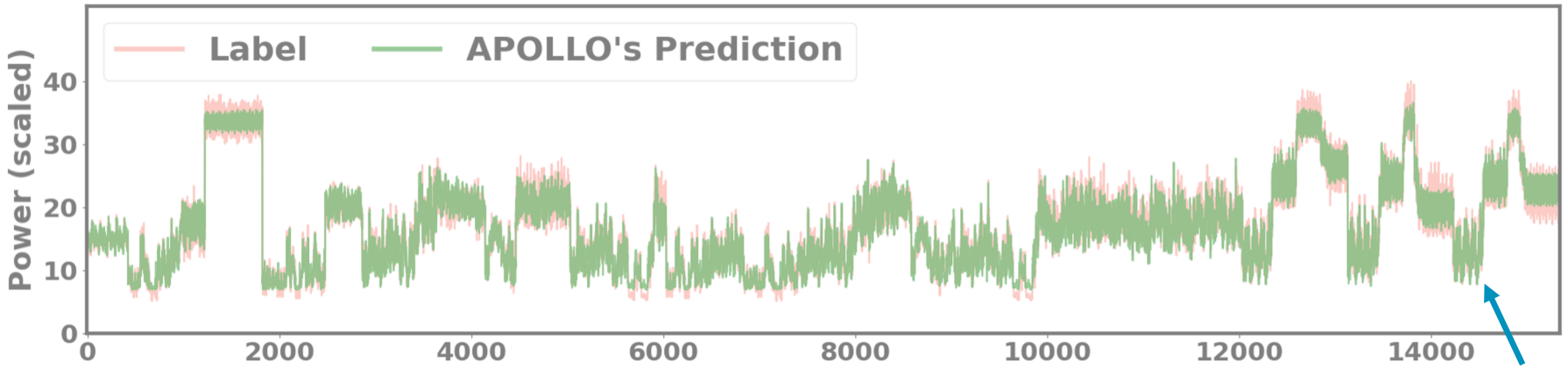
Per-cycle prediction from APOLLO with $\boldsymbol{Q}$=159 proxies



- MAE = 7.19%
- RMSE = 9.13%
- R^2 = 0.953

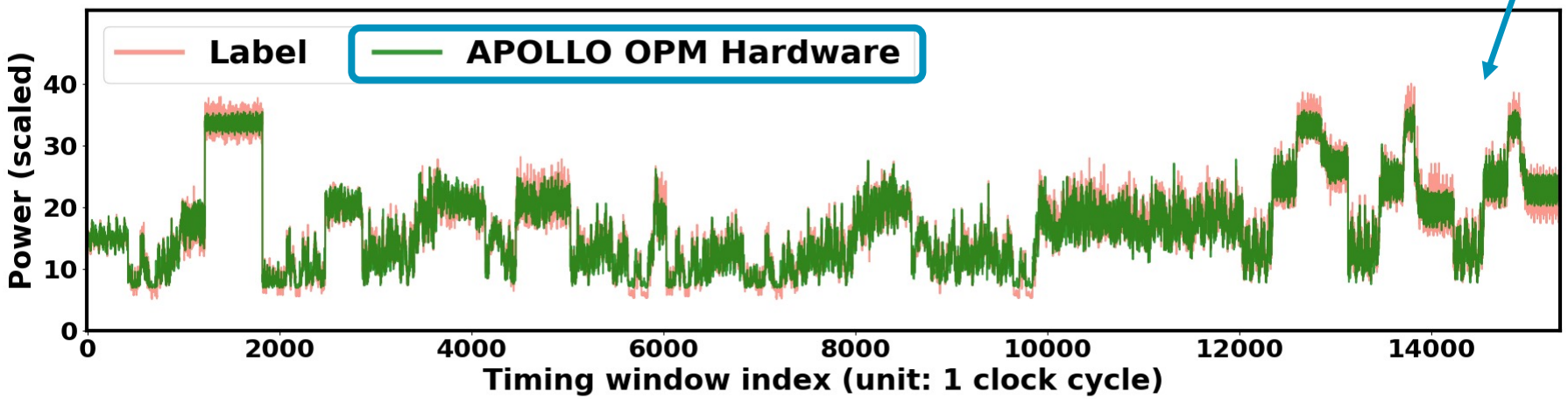
Prediction Accuracy from Design-time Model & OPM

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



Negligible difference

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



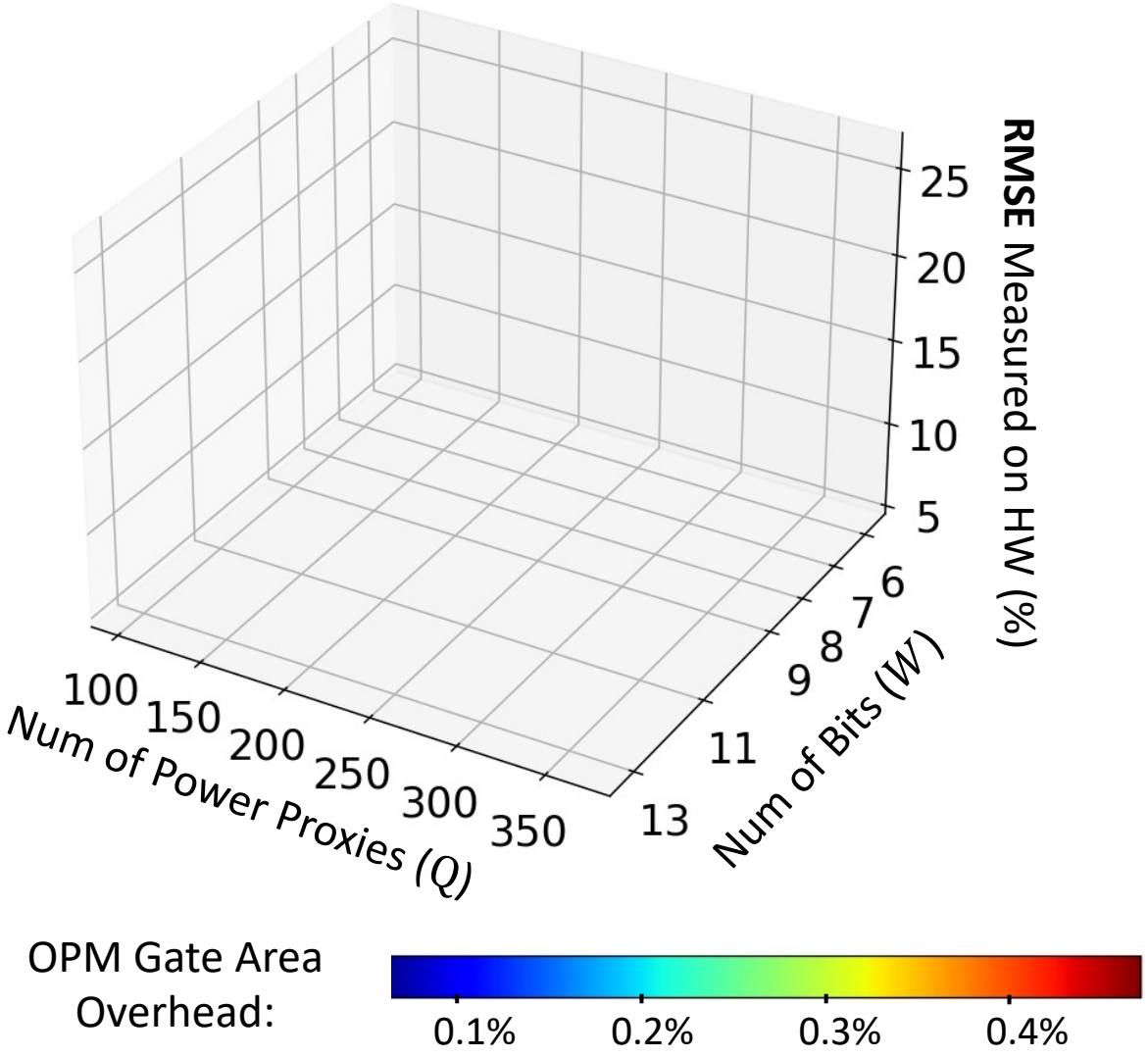
- MAE = 7.19%
- RMSE = 9.13%
- R^2 = 0.953

< 0.02% difference

- MAE = 7.19%
- RMSE = 9.15%
- R^2 = 0.953
- **W=11 bits after quantization**

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

Runtime OPM implementation on Neoverse N1

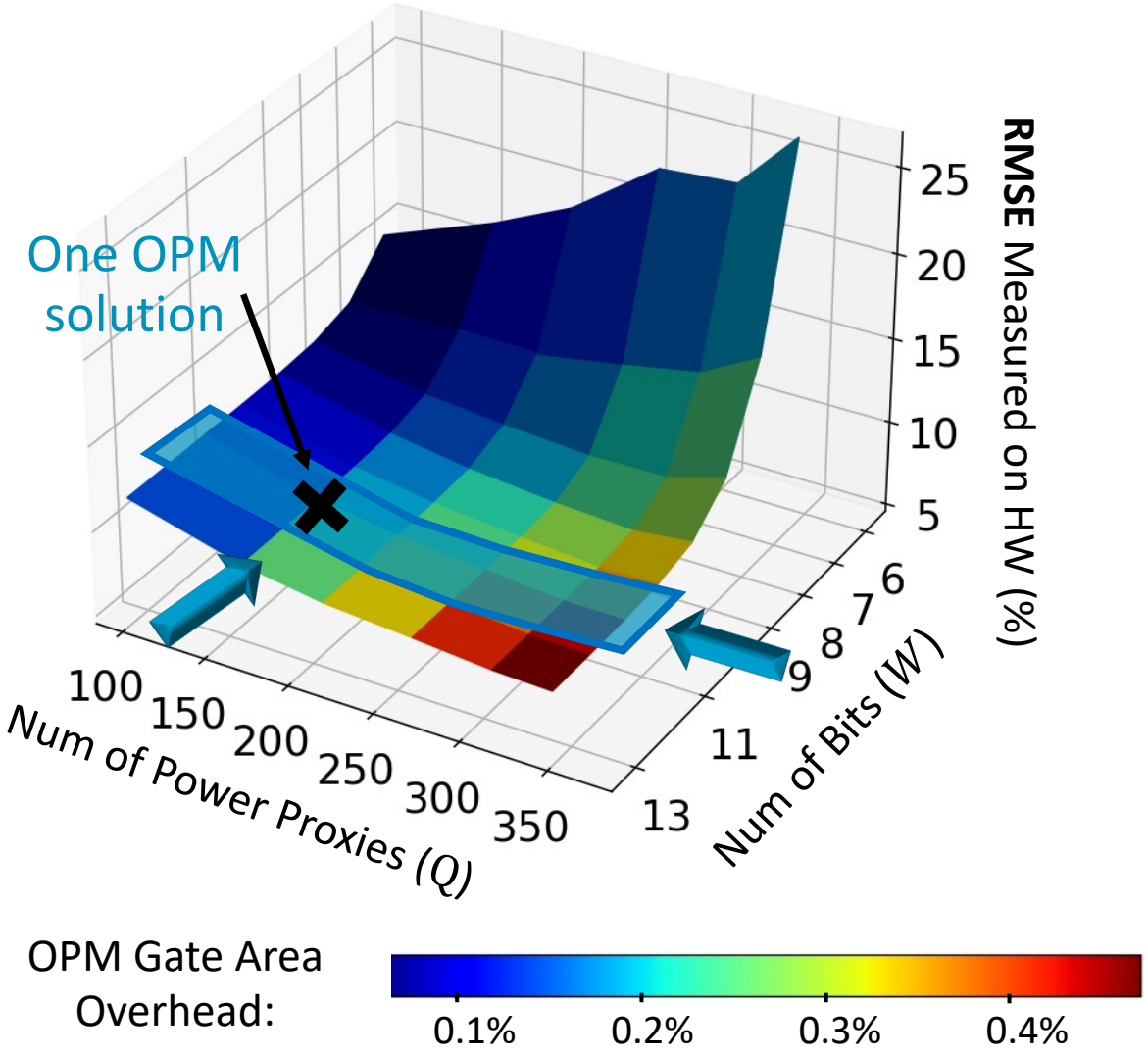


- Trade-off accuracy and hardware cost
- Sweep proxy num $\boldsymbol{Q}$ and quantization bits $\boldsymbol{W}$

arm

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

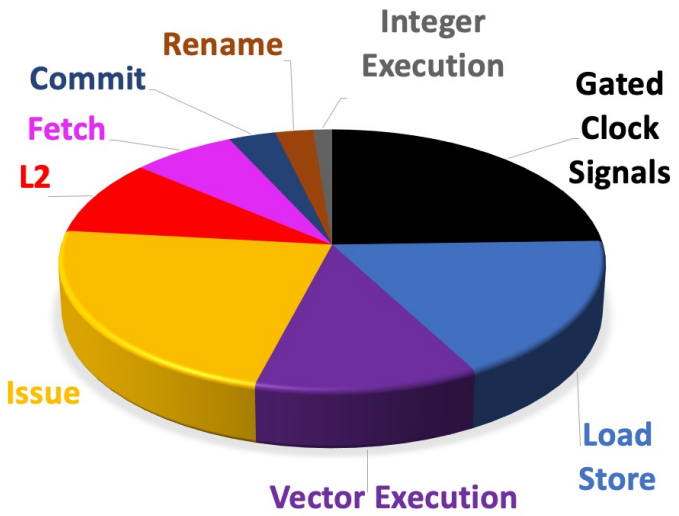
Runtime OPM implementation on Neoverse N1



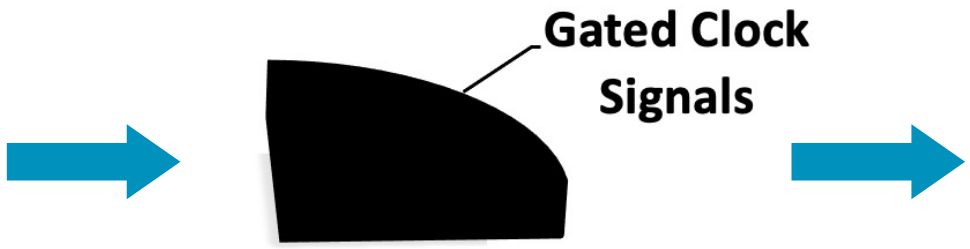
- Trade-off accuracy and hardware cost
- Sweep proxy num $\boldsymbol{Q}$ and quantization bits $\boldsymbol{W}$
- **Strategy**
  - Keep $\boldsymbol{W}$= 10 to 12
  - Vary $\boldsymbol{Q}$ for different solutions
- **For an OPM with $\boldsymbol{Q}$=159, $\boldsymbol{W}$=11**
  - **< 0.2%** area overhead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

arm

Future Work (1) : Design-time Power Introspection



Distribution of power proxies on Neoverse N1.



Trained **only** with **transaction-qualifiers** as raw input feature-list

- Q=118 proxies
- MAE = 13%
- **Sufficient for design decisions**

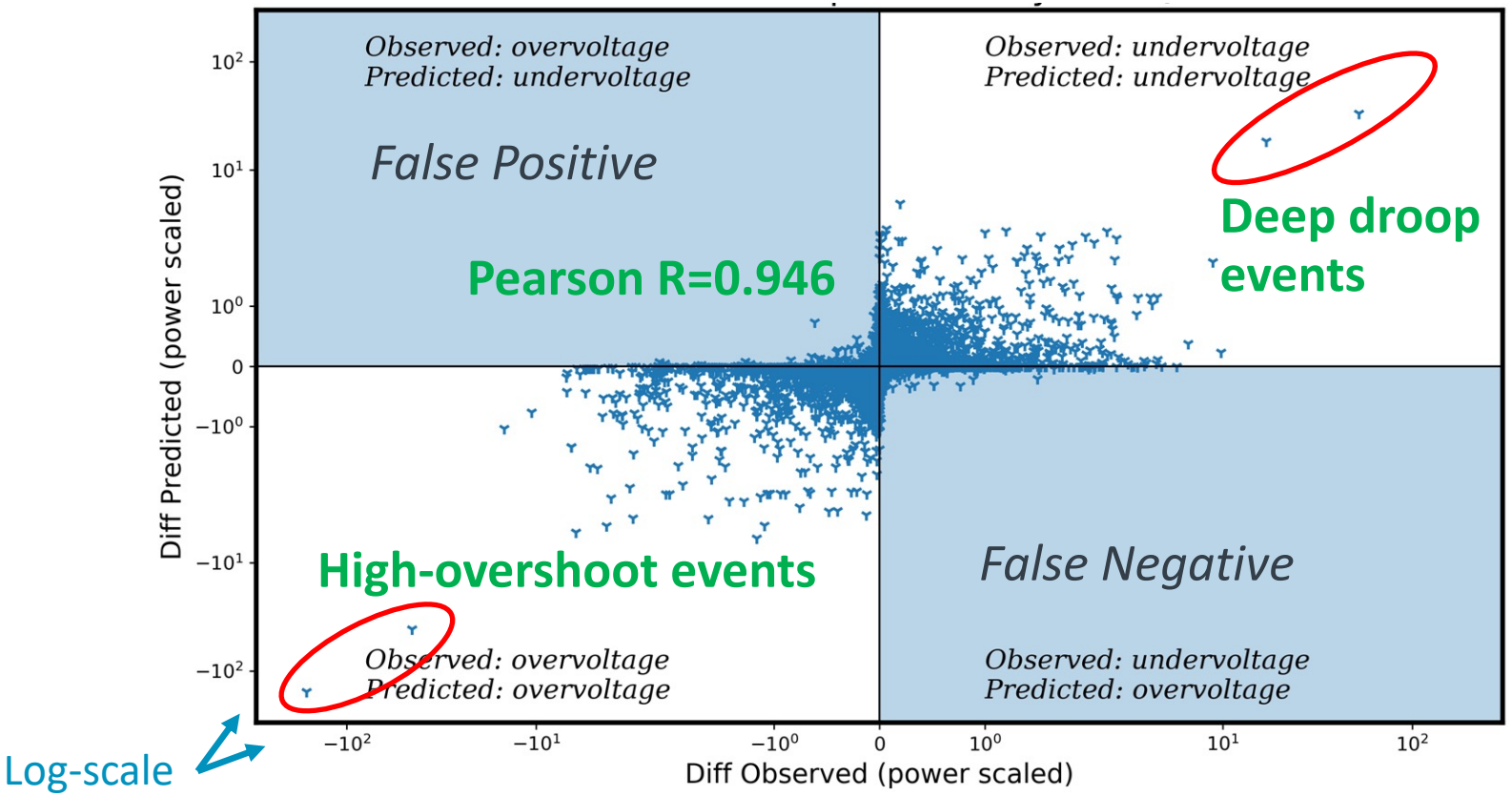
Micro-architects require accurate handle on power contributors during RTL development

Restrict the proxy-search to an enumerated list of architectural transaction-qualifiers

- Can be “valid-signals” or “clock-gating enables” or designer-identified control-plane qualifiers
- Modify the fundamental APOLLO algorithm to train for signal “levels” instead of “toggles”
- Trade-off model accuracy for much greater **interpretablity**, benefiting design decisions

arm

Future Work (2): CPU-driven Proactive di/dt Mitigation



Please check our paper for details on proactive droop mitigation using the OPM

- OPM-generated current readings are differentiated to obtain di/dt events
- Excellent correlation is obtained for deep droop and deep overshoot events

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**

arm Research

This presentation and recording belong to the authors.
No distribution is allowed without the authors' permission.

Contact us:
zhiyao.xie@duke.edu, Shidhartha.Das@arm.com

Thank You
Danke
Gracias
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה