A Collaborative Machine Learning Approach to Fast and High-Fidelity Design Prediction

Task 2810.021
Jiang Hu
Dept of ECE
Texas A&M University

Task 2810.022
Yiran Chen
Dept of ECE
Duke University

Task Overview

- Center: Texas Analog Center of Excellence (TxACE)
- Thrust: Computer-Aided Design and Test (CADT)
- Subthrust: System, Logic and Physical Design (SLPD)
- Task leaders
  - Jiang Hu, Texas A&M Univ, Task 2810.021
  - Yiran Chen, Duke Univ, Task 2810.022
- Start date: January 1, 2019
- Industrial liaisons
  - Gi-Joon Nam, IBM
  - Xiaoqing Xu, ARM
  - Divya Prasad, ARM
  - Savithri Sundareswaran, NXP

Anticipated Results

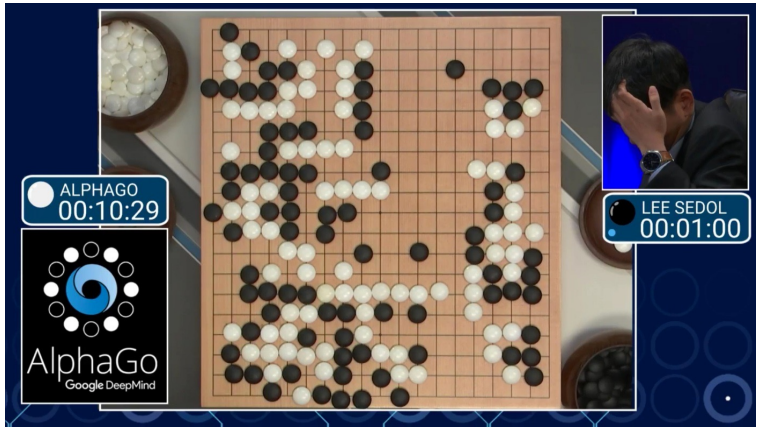
- Machine learning-based techniques for fast and high-fidelity prediction of
  - circuit routability
  - timing
  - net-length
  - power
  - crosstalk noise
- Routability/crosstalk predictions will be useful for analog and mixed-signal designs
- Machine learning guided synthesis parameter tuning

Planned Deliverables

- Machine learning-based early routability prediction for digital and analog IC designs. (12/2019)
- Machine learning-based early timing prediction for digital IC designs. (12/2019)
- Machine learning-based early crosstalk noise prediction for digital and analog IC designs. (12/2020)
- Machine learning-based early power prediction for digital IC designs. (12/2020)
- Machine learning-based pre-layout net-length prediction. (12/2021)
- Machine learning-based synthesis parameter tuning. (12/2021)

Background and Motivation

- Decisions in early design steps have large impact
- Need fast and high-fidelity predictions
- Existing techniques
  - Analytical: fast but inaccurate
  - Trial design: accurate but very slow
- Machine learning
  - Extracting design knowledge from data
  - Emulating design experience



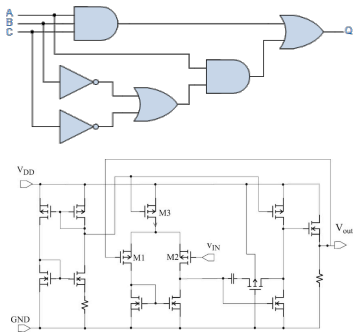
Pre-layout Net-length Prediction

- Net-length: a deciding factor for
  - Power
  - Performance
    - Digital timing
    - Analog performance by parasitic
- Prediction before time-consuming layout
  - Critical for estimating power-performance in synthesis
    - Digital logic synthesis
    - Analog schematic design

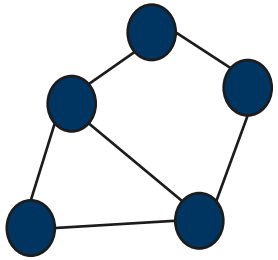
Previous Approaches

- Mutual contraction (MC) [DAC03] ~ net-degree/neighboring-net-degree
- Intrinsic shortest path length (ISPL) [ICCAD05] ~ shortest path between two nodes except the edge in between
- Polynomial model [TVLSI01, SLIP09]
- ANN for total FPGA wirelength [FPT12]
- Machine learning for path-length with virtual P&R [DATE19]

Our Approach: Net^2



Circuit netlist

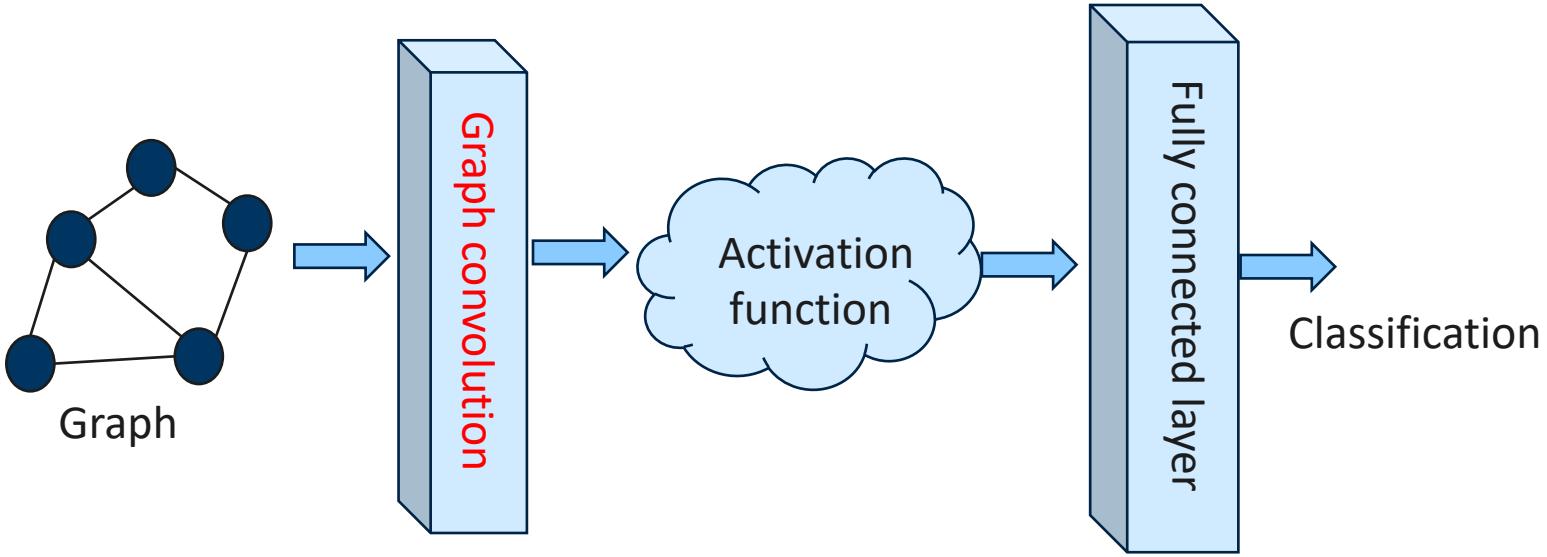


Graph

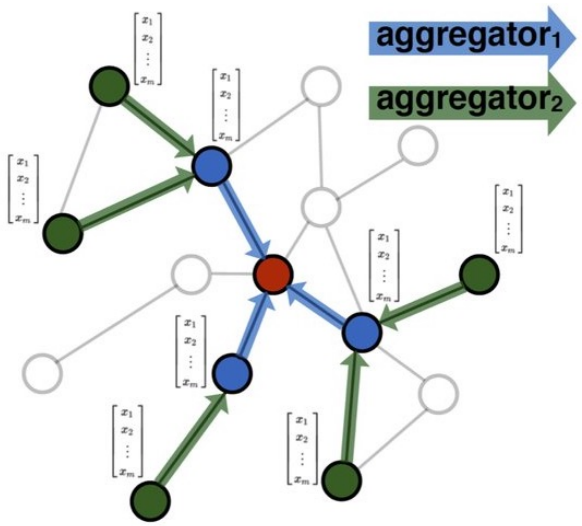
Graph Neural Network

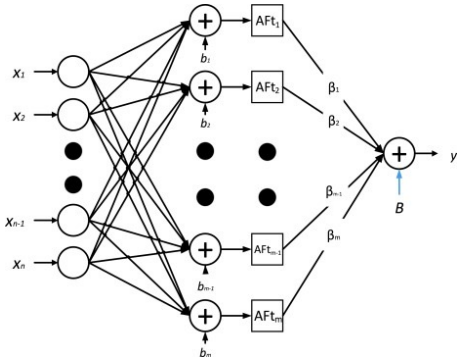
Net-length

Graph Neural Network



Graph convolution
≈ Feature aggregation



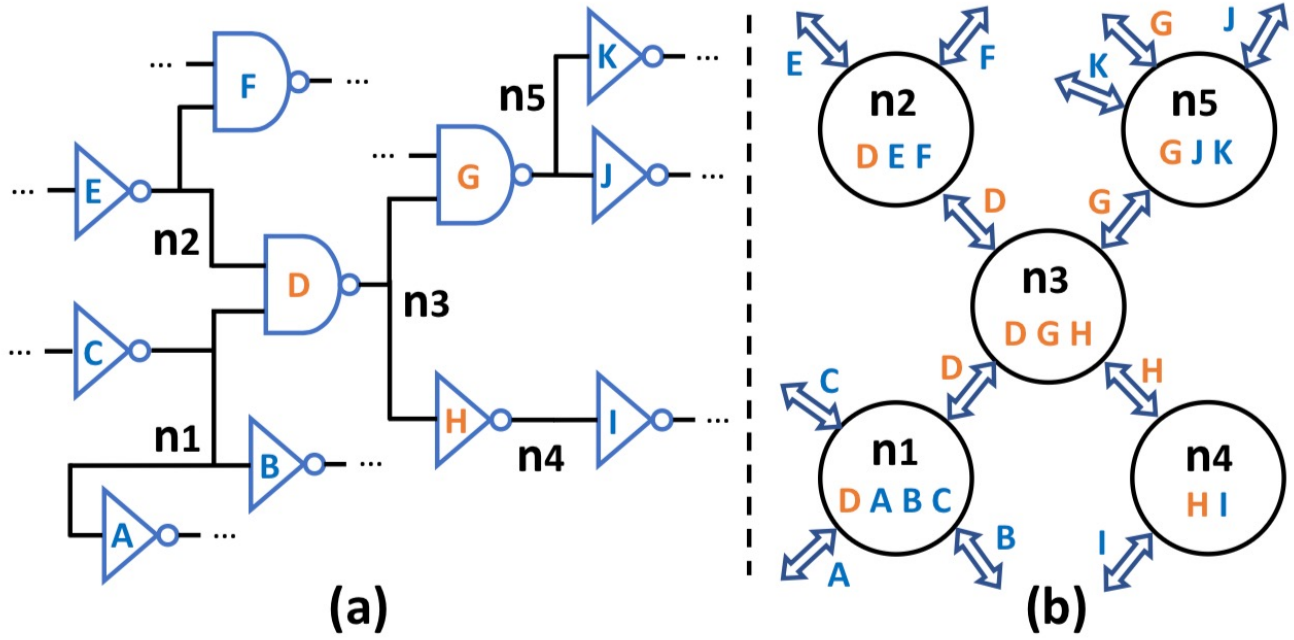


Existing GNN Techniques

- Graph convolutional network (GCN) [Kipf and Welling, ICLR 2017]
- GraphSage [Hamilton, Ying and Leskovec, NIPS 2017]
- Graph attention network (GAT) [Velickovic, et al., ICLR 2018]
- Edge features for GNN [Gong and Cheng, CPVR 2019]

Circuit Graph in Net^2

- Each net == a node,
- Label == HPWL (Half Perimeter Wire-Length) of each net

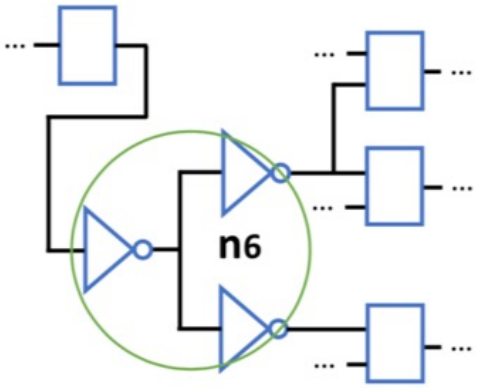


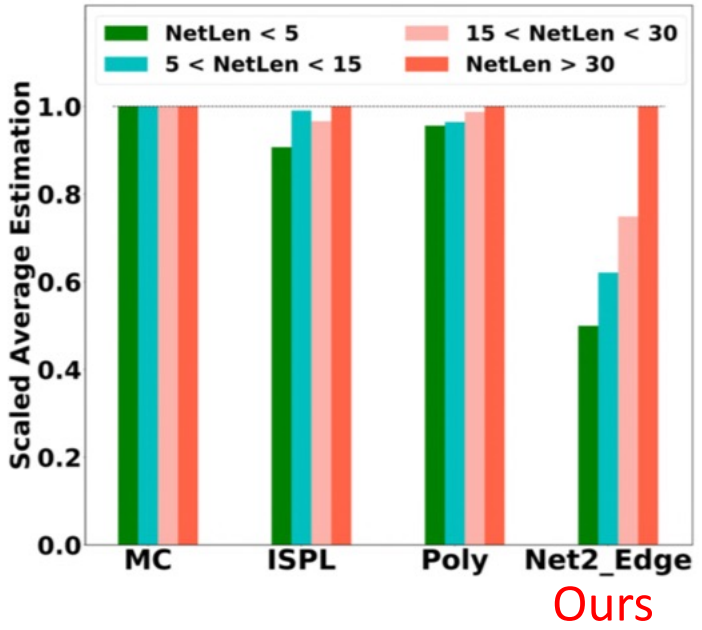
Node Features

- A node == a net
- Driver area
- Fan-in size
- Fan-out size
- Sum of cell/device area
- Fan-in and fan-out of neighbor nodes
- Sum and standard deviation of neighboring fan-in/fan-out

Local Information Insufficient

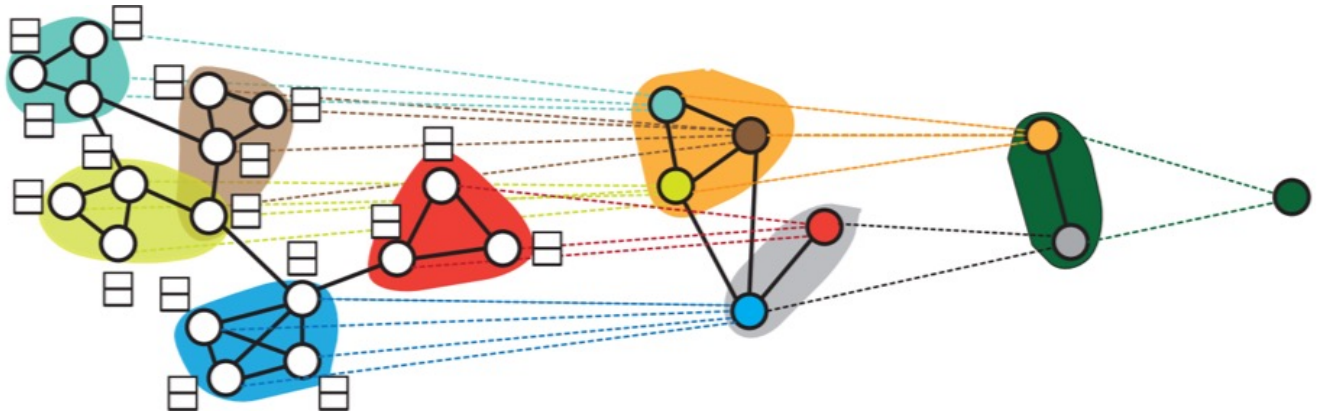
- Net n6 has one 2-pin fan-in, one 2-pin fan-out and one 3-pin fan-out
- In one circuit, 725 nets have the same fan-in and fan-out as n6
- The 725 nets are bucketed into 4 groups according to actual net-length
- Previous methods fail to differentiate the 4 groups



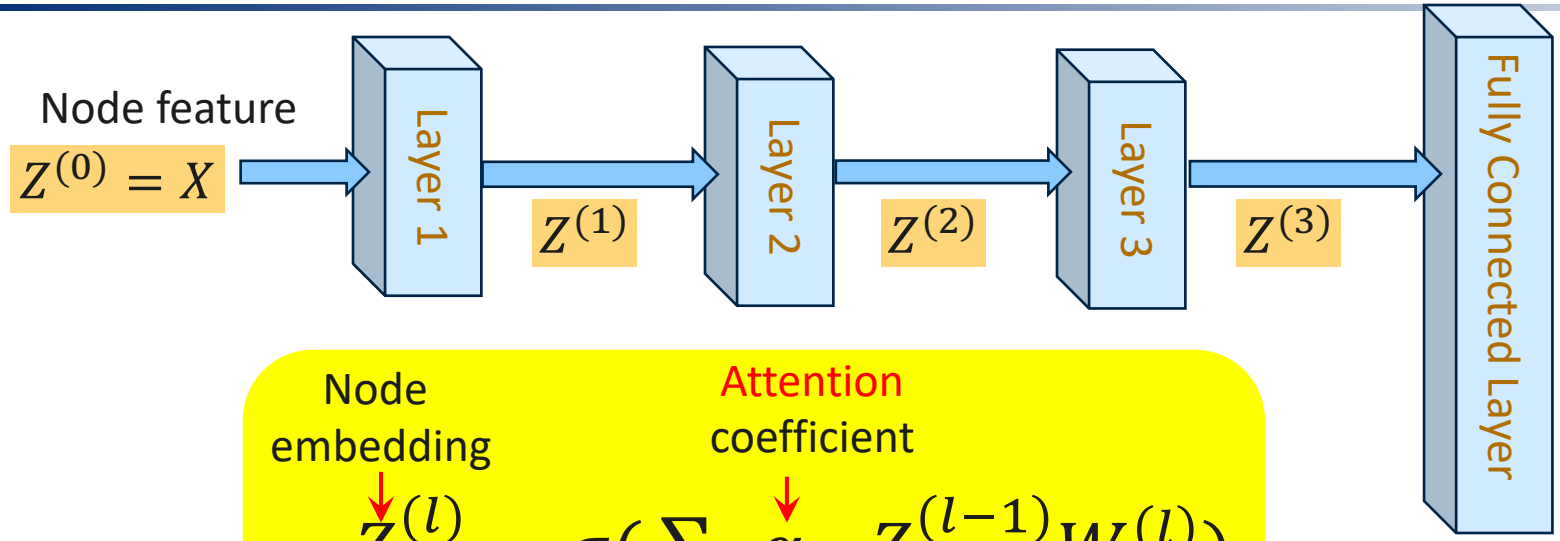


Clustering and Edge Features

- Multi-level clustering
- Each net/cell has a cluster ID at each level
- Two cells are closer if they have the same ID
- High (coarse) level clusters => global information
- Encode ID differences at multiple levels as edge features



Graph Attention Network (GAT)



Node embedding

Attention coefficient

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

Activation function

Trainable weight

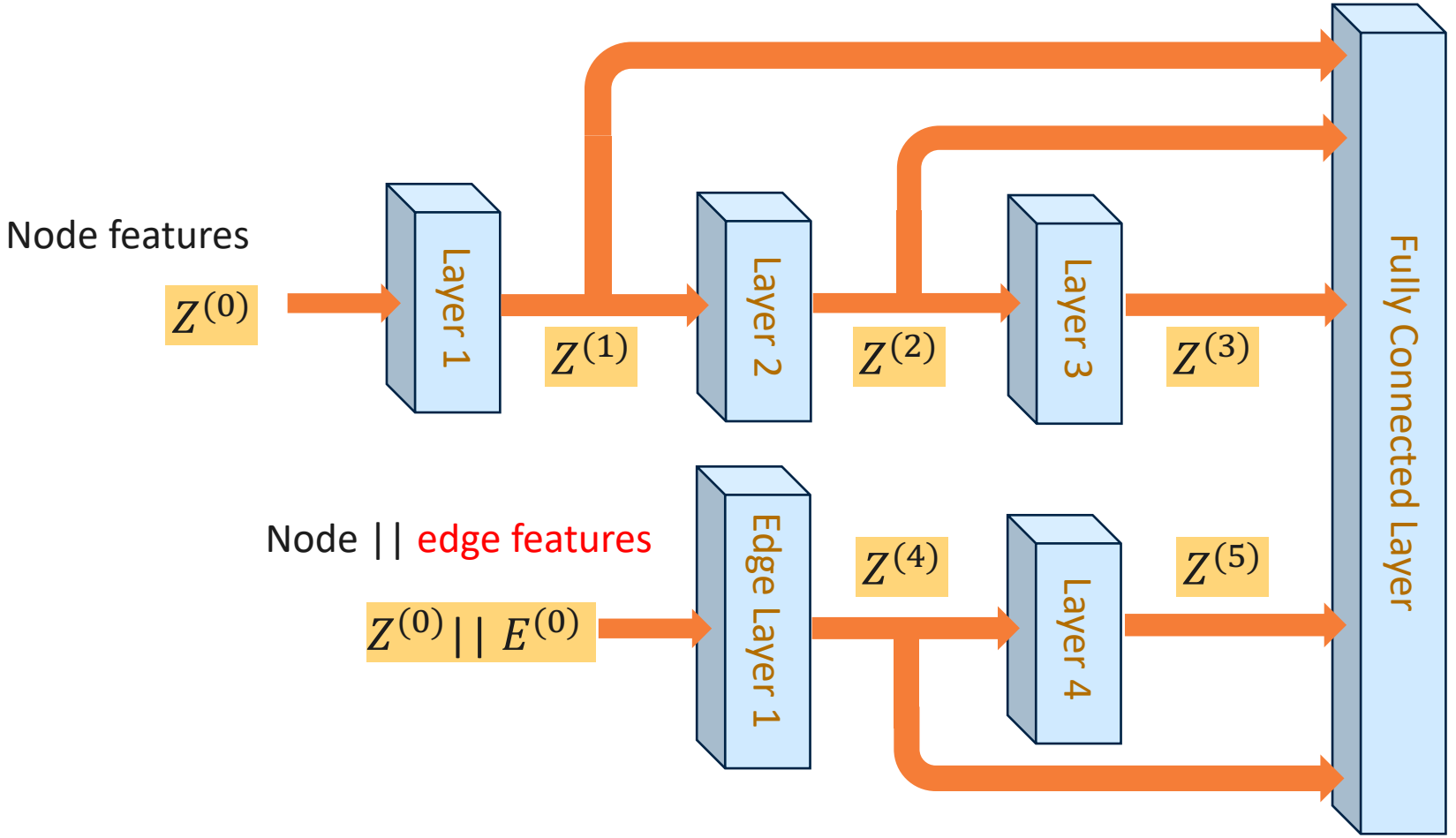
For node *i* and all its neighbors *j* :

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

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

Trainable vector

Customization to GAT

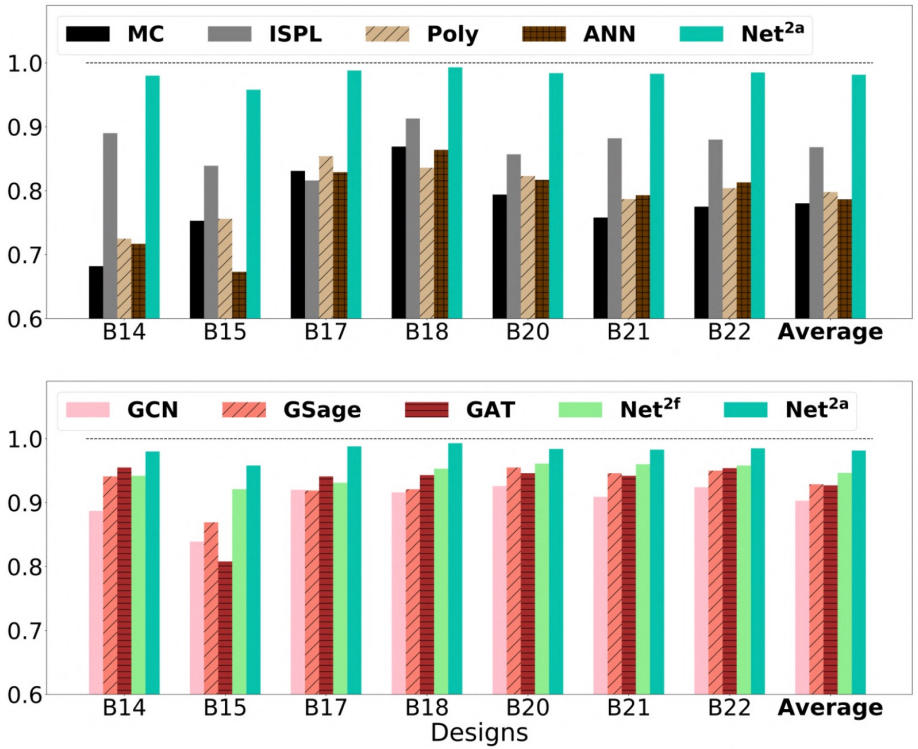


Experiments

- 7 designs from ITC99
- For each design, synthesize 10 different netlists
- Testing on one design, model trained on the other 6 designs

	B14	B15	B17	B18	B20	B21	B22
Smallest	13 K	5.3 K	18 K	54 K	26 K	26 K	39 K
Largest	34 K	15 K	49 K	138 K	67 K	66 K	99 K

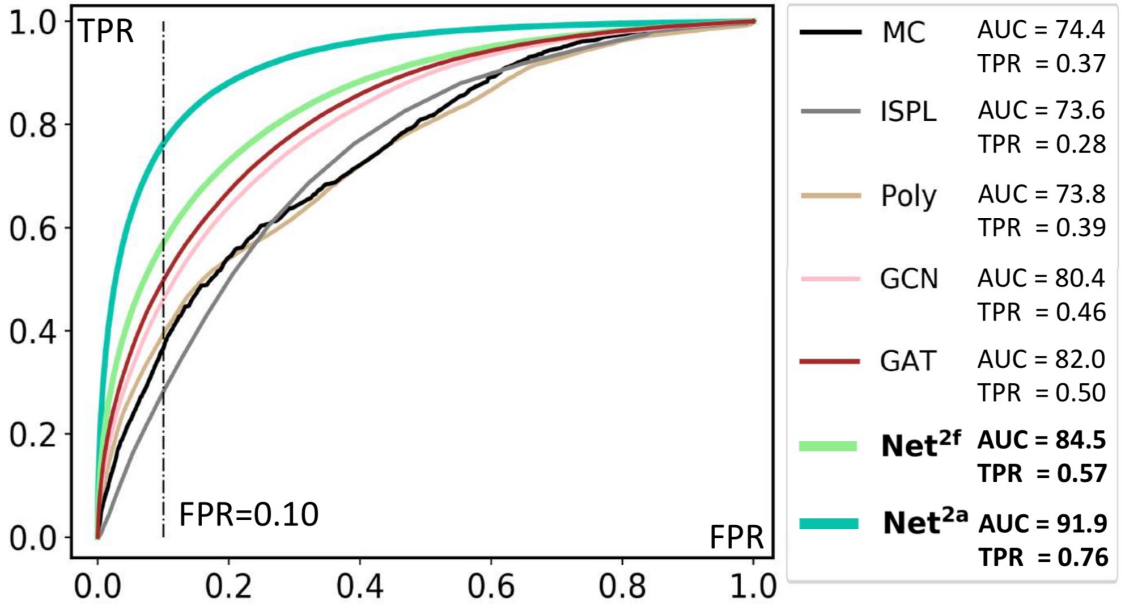
Net^2 Result for Net Size



Correlation coefficient R between prediction and label measured in 20 bins

Net^{2f}: fast

Net^{2a}: accurate



ROC curves in identifying 10% longest nets

Net^2 Result for Path Length

Identifying 10% Longest Paths in ROC AUC (%)

Methods	B14	B15	B17	B18	B20	B21	B22	Ave
ISPL	58.9	57.5	56.5	74.0	72.5	63.0	75.5	65.4
Poly	65.5	80.0	78.0	68.0	82.0	85.0	84.0	77.5
ANN	68.0	76.0	80.0	69.0	78.5	82.0	75.5	75.6
GCN	63.5	75.0	86.5	56.0	82.0	81.5	85.5	75.7
GSage	65.0	88.0	93.0	77.0	81.5	67.0	80.0	78.8
GAT	63.0	92.0	95.0	83.5	83.5	76.0	89.5	83.2
Net$^{\mathbf{2f}}$	79.0	88.5	97.5	84.0	75.5	83.0	92.0	85.6
Net$^{\mathbf{2a}}$	86.5	95.0	96.0	90.5	90.5	93.5	95.5	92.5

Comparing Pair of Paths by Lengths (%)

Methods	B14	B15	B17	B18	B20	B21	B22	Ave
ISPL	67.1	55.0	58.2	77.4	68.9	59.7	69.5	65.1
Poly	83.9	86.6	83.3	70.4	83.4	80.4	86.3	82.0
ANN	82.0	74.8	75.3	68.1	81.9	65.4	80.5	75.4
GCN	74.5	85.9	83.0	62.4	83.4	81.0	86.2	79.5
GSage	84.2	92.5	83.9	75.3	89.1	62.8	88.1	82.3
GAT	82.4	93.5	85.1	80.6	89.7	87.5	88.2	86.7
Net$^{\mathbf{2f}}$	87.3	92.7	87.6	93.1	91.1	91.2	86.9	90.0
Net$^{\mathbf{2a}}$	96.8	97.0	91.4	95.9	92.2	94.2	94.4	94.6

Net^2 Runtime

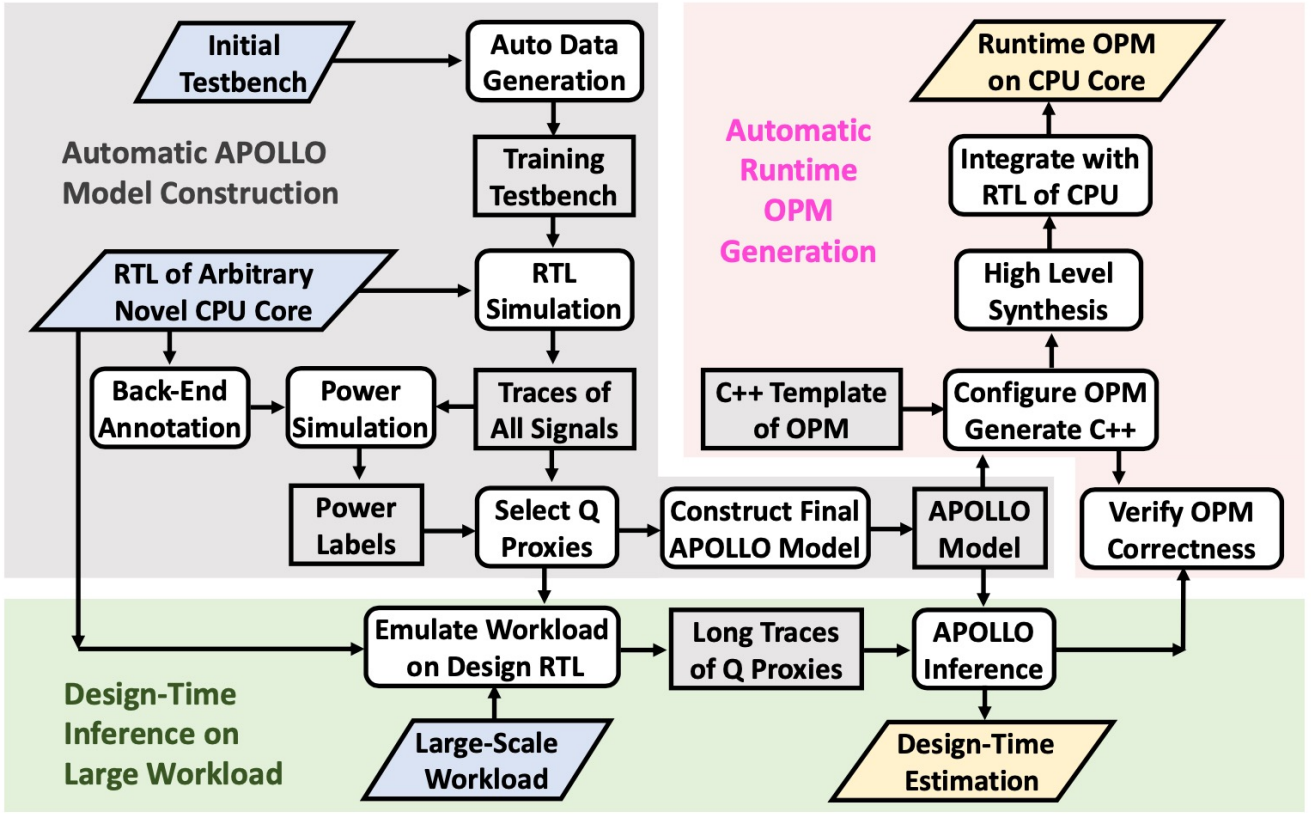
Design	Place	Partition	Net^{2f} Infer	Net^{2a} Infer	Net^{2f} Speedup	Net^{2a} Speedup
Ave	97.8	7.0	0.05	0.07	1.7K ×	14.3×

Design-Time and Runtime Power Prediction

- Design-time power prediction: critical for design decisions
  - Challenge: long simulation time, an hour simulation for dozens of clock cycles
  - FPGA-emulation: data for all signal toggles is too huge to be processed
- Runtime power prediction: power management
  - Challenge: simultaneously achieve fine temporal resolution and low area overhead
  - Performance-counter-based: low overhead, but resolution of thousands of cycles at the best, not sufficient for instruction throttling
  - OPM (On-chip Power Meter) on RTL signals (proxies): better resolution, but significant hardware overhead, at least 4%
- Many previous works, but the challenges have not been solved

Overview of Our Work

- A single framework named APOLLO
  - Enables simulating millions of cycles in several minutes
  - Runtime OPM with cycle-accuracy and <1% area overhead



The automated APOLLO framework

Centerpiece of Our Approach

Power prediction for cycle *i*

$$p[i] = \sum_{j=1}^{Q} w_j \cdot x_j[i]$$

Signal *j* toggles or not

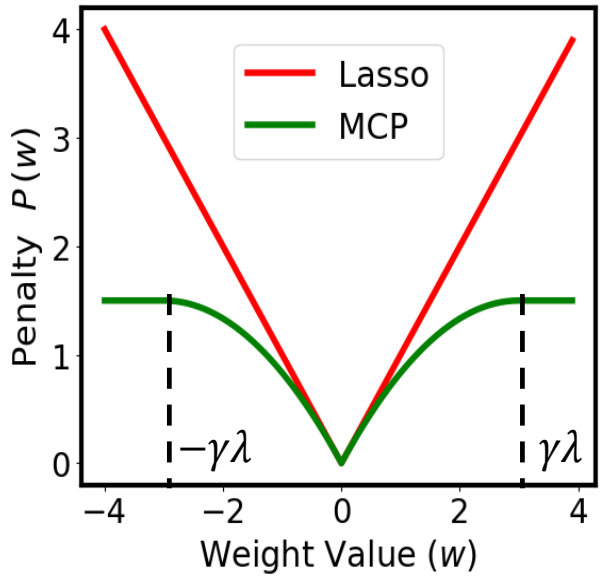
Trainable weight

- Select Q power proxies among all (M) RTL signals, $\frac{Q}{M} < 0.05\%$
- Linear model is common, but our proxy selection is novel
- Train weights as in machine learning
- Toggle x_j is binary, implemented with “AND” gates instead of multipliers

Machine Learning-Based Proxy Selection

$$\tilde{p}[i] = \sum_{j=1}^{M} \tilde{w}_j \cdot x_j[i]$$

- Train a linear model with ALL RTL signals
- Loss function penalizes weights besides errors
- Weights approach 0 by iterations



$$Loss = ||\underbrace{y}_{\text{Label}} - \tilde{p}||_2^2 + \sum_{j=1}^{M} \underbrace{P}_{\text{Penalty}}(\tilde{w}_j)$$

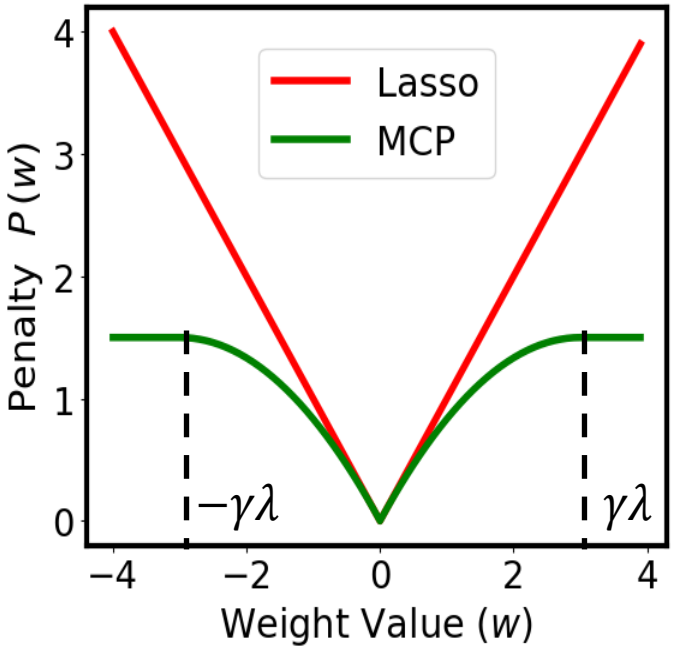
MCP(Minimax Concave Penalty)

$$P_{MCP}(\tilde{w}_j, \gamma > 1) = \begin{cases} \lambda|\tilde{w}_j| - \dfrac{\tilde{w}_j^2}{2\gamma} & \text{if } |\tilde{w}_j| \le \gamma\lambda \\ \dfrac{1}{2}\gamma\lambda^2 & \text{otherwise} \end{cases}$$

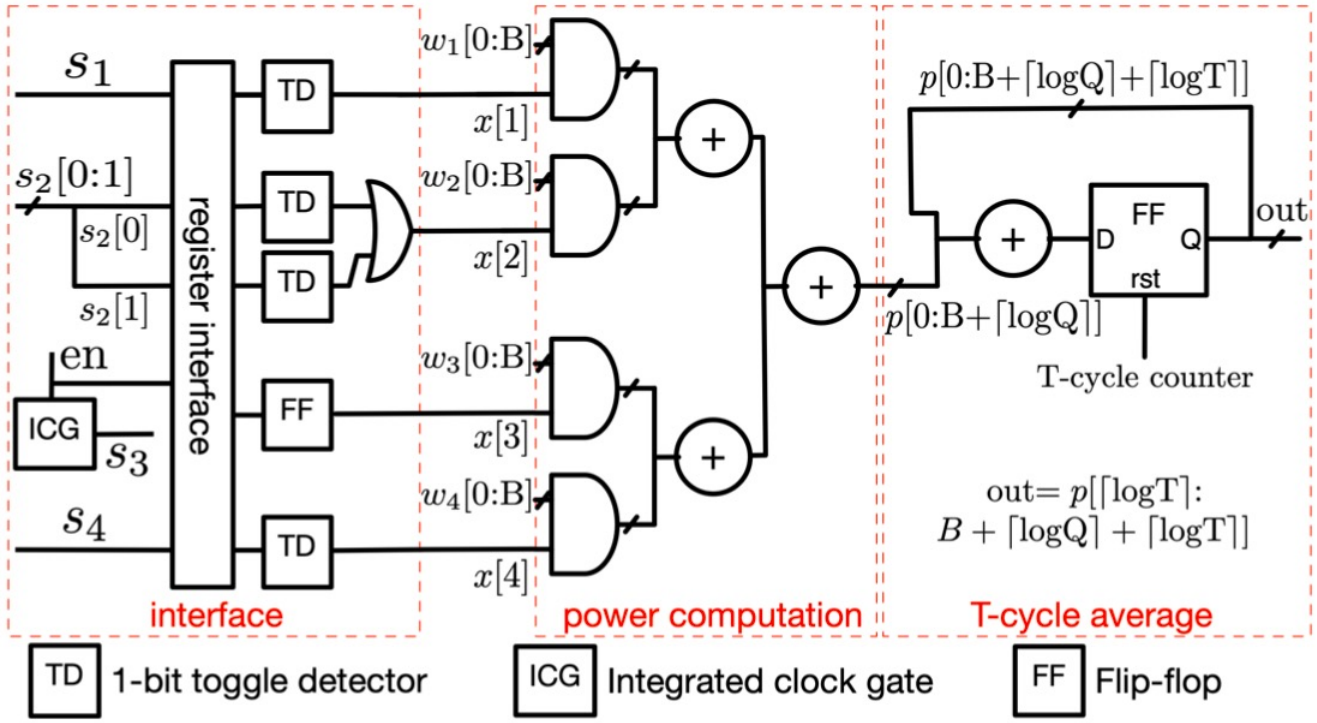
Machine Learning-Based Proxy Selection

- After iterations, many weights become 0
- Remove all 0-weight terms
- Keep only Q non-zero-weight terms (proxies)
- Retrain the model again from scratch

$$p[i] = \sum_{j=1}^{Q} w_j \cdot x_j[i]$$



Efficient APOLLO-OPM Implementation

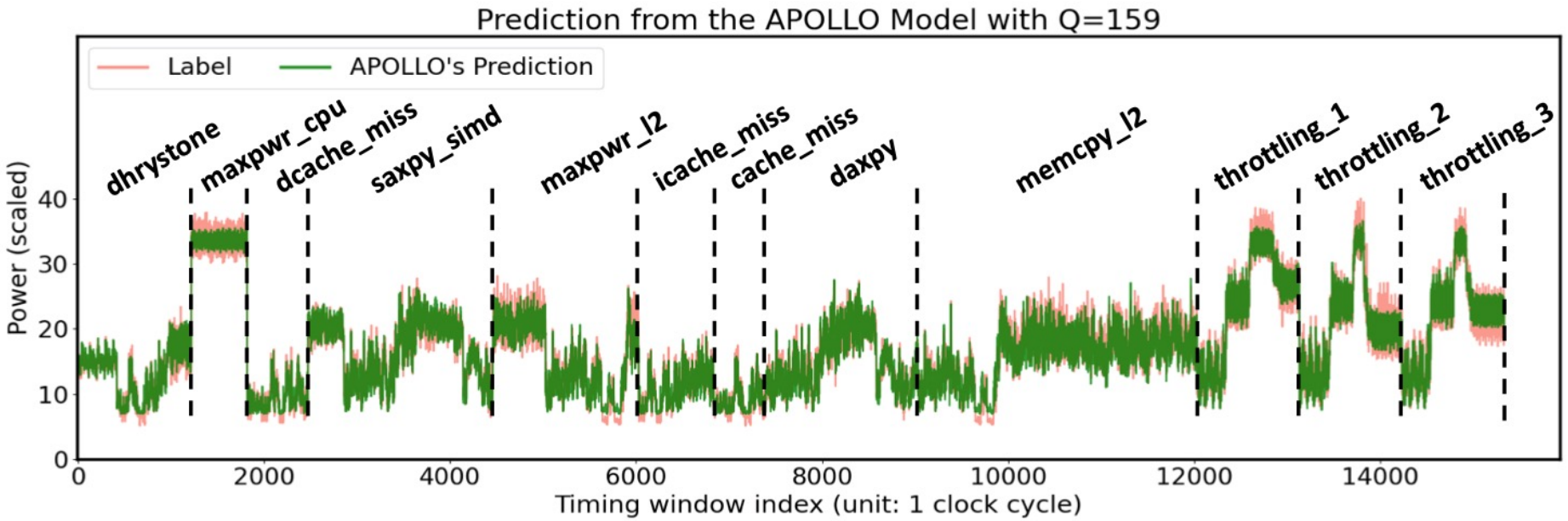


- Automated runtime OPM generation
  - OPM configured from generic C++ templates based on trained model
  - Integrated with commercial CPU
  - HLS -> synthesis -> layout
- Low-cost OPM
  - No multipliers
  - Only one counter
  - Small proxy number Q
  - Weights quantized to 10-bit

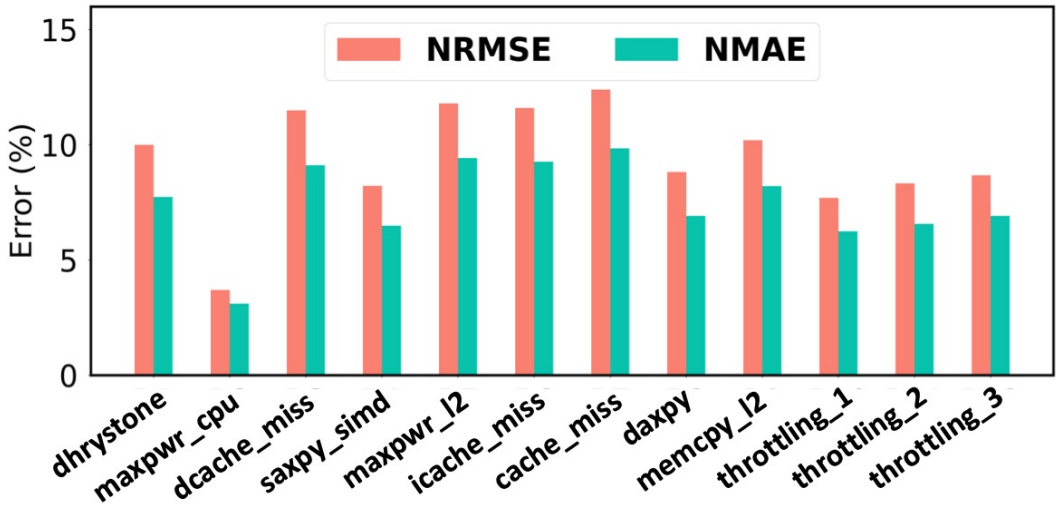
Experiment Setup

- High-volume commercial million-gate CPU designs
  - ARM Neoverse N1 microprocessor
  - ARM Cortex-A77 microprocessor
  - #RTL signals > half million
- 7nm technology
- Automatically generated random benchmarks for training
- 12 designer-handcrafted micro-benchmarks for testing

Power Prediction Demonstration

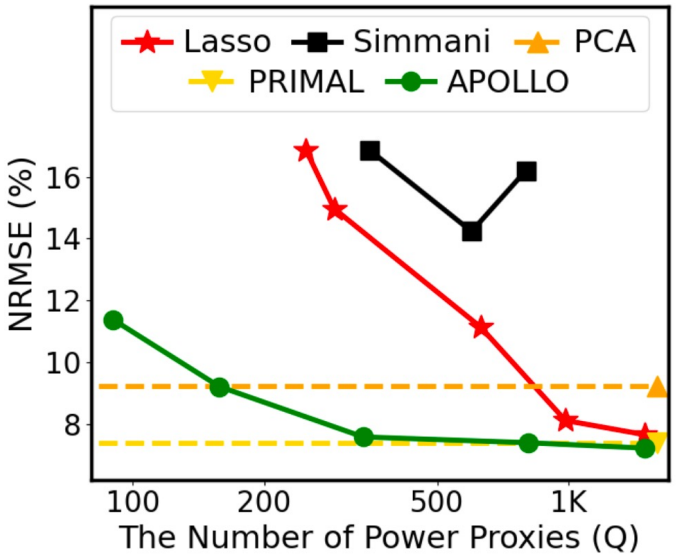


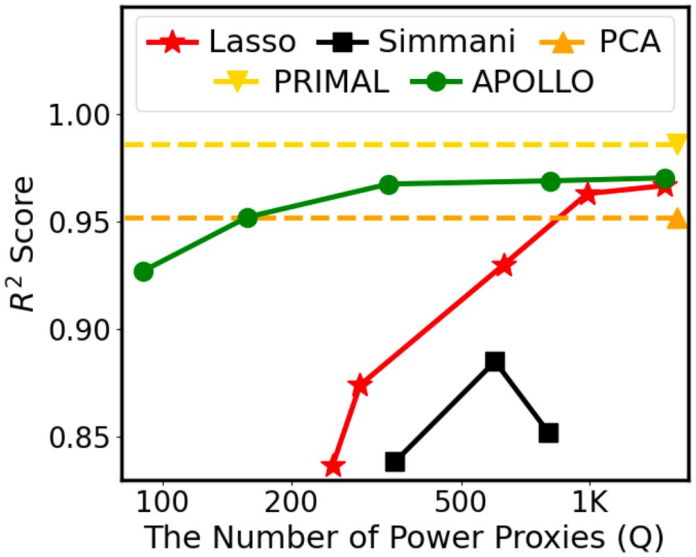
Neoverse N1



Per-cycle accuracy on each testing benchmark

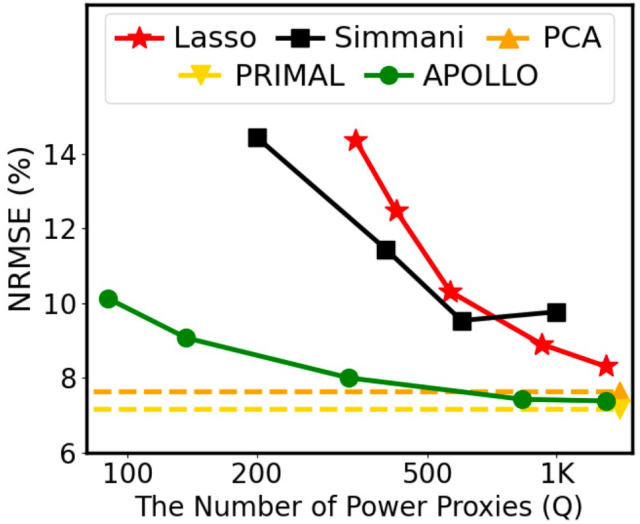
Accuracy vs #Proxies

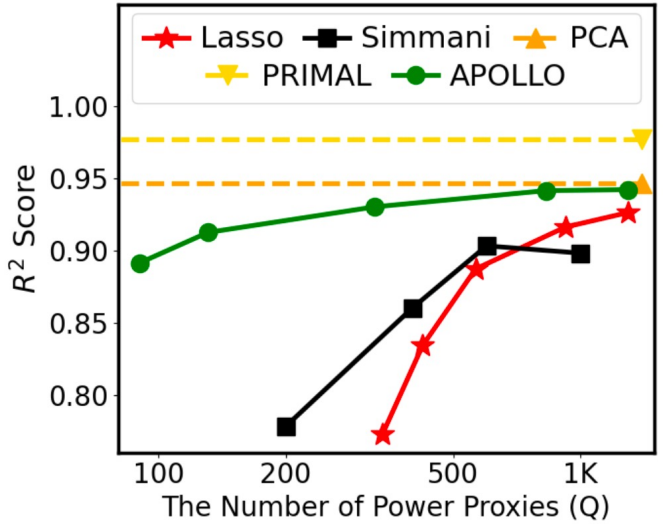




Neoverse N1

Cortex A-77

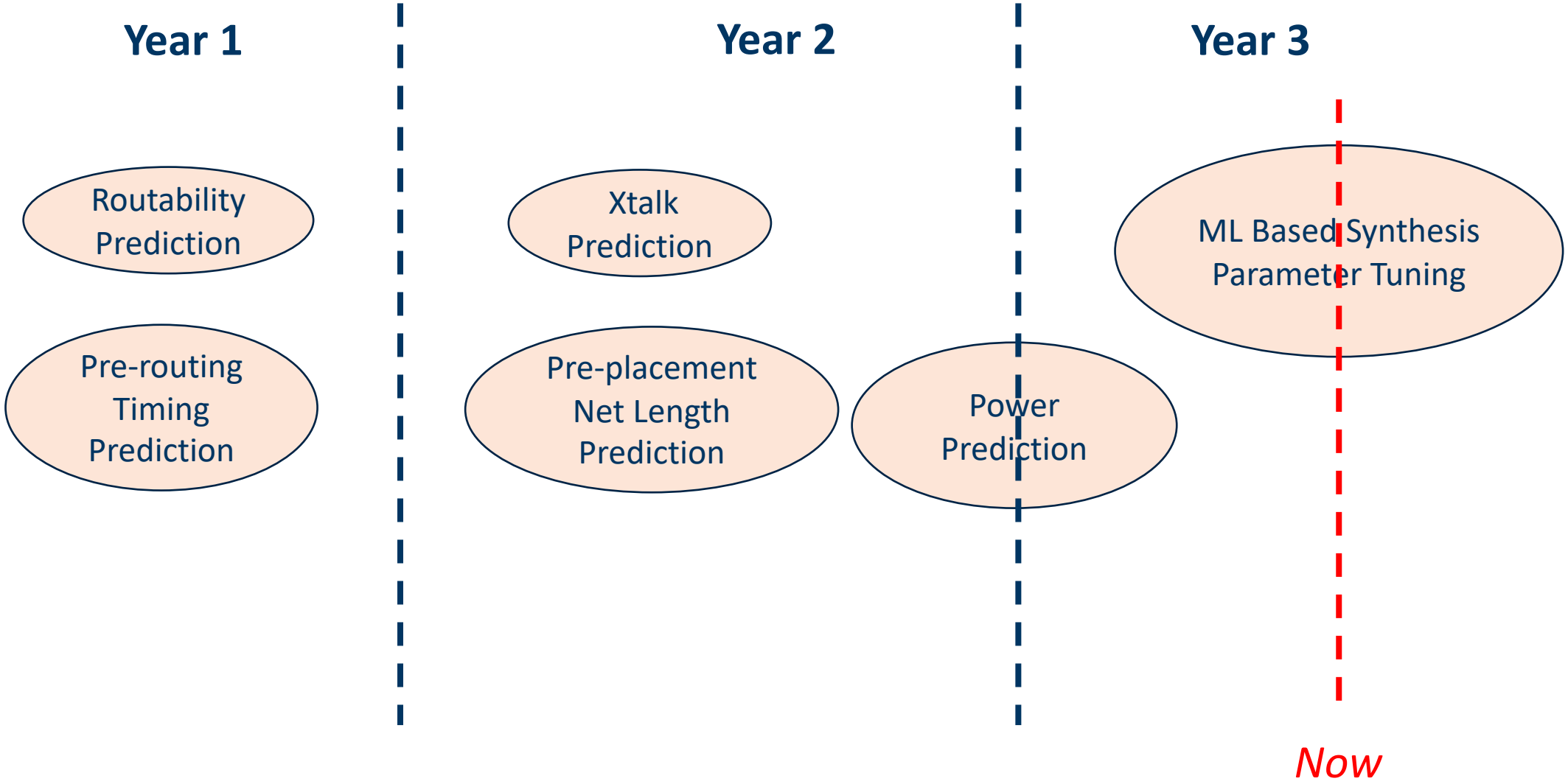




Achievements of APOLLO

- A highly automated framework for novel microprocessor designs
- Design-time power estimation
  - Software simulation, 1 hour simulates 20 clock cycles
  - APOLLO enables cycle-accurate million-cycle power estimation in several minutes
- Run-time power monitoring
  - APOLLO is the first to simultaneously achieve cycle-accuracy with < 0.2% hardware overhead

Progress and Next Steps



Publications

- Z. Xie, Y.-H. Huang, G.-Q. Fang, H. Ren, S.-Y. Fang, Y. Chen and J. Hu, “RouteNet: Routability Prediction for Mixed-Size Designs Using Convolutional Neural Network,” *IEEE/ACM International Conference on Computer-Aided Design*, 2018.
- E. C. Barboza, N. Shukla, Y. Chen and J. Hu, “Machine Learning-Based Pre-routing Timing Prediction with Reduced Pessimism,” *ACM/IEEE Design Automation Conference*, 2019.
- R. Liang, H. Xiang, D. Pandey, L. Reddy, S. Ramji, G.-J. Nam and J. Hu, “DRC Hotspot Prediction at Sub-10nm Process Nodes Using Customized Convolutional Network,” *ACM International Symposium on Physical Design, 2020*.
- Z. Xie, R. Liang, X. Xu, J. Hu, Y. Duan and Y. Chen, “Net^2: A Graph Neural Network Method Customized for Pre-Layout Wirelength Estimation,” *ACM/IEEE Asia and South-Pacific Design Automation Conference*, 2021.

Students on Task 2810.021, 2810.022

- Zhiyao Xie, Duke University, expected graduation: 2022
- Rongjian Liang, TAMU, expected graduation: 2021

- Internship: Zhiyao Xie, ARM, summer 2020
- Internship: Rongjian Liang, IBM, summer 2019, 2020
- Planned internship: Rongjian Liang, NXP, summer 2021

Interactions with SRC Companies

- Weekly meetings with IBM
- Meetings with ARM
- Biweekly meetings with NXP

Thank You!