

Intelligent Circuit Design and Implementation with Machine Learning in Design Automation



Zhiyao Xie
Advisor: Yiran Chen, Hai (Helen) Li
Duke University



Background and Takeaway

Traditional Chip Design: A flowchart showing RTL Design, Synthesis, Layout, Verification, and Fabrication. ML for EDA in commercial tools (Cadence Innovus™, Synopsys ICC™ II) is shown as a separate module.

Contributions of my Ph.D. works:

- Bridge separated design steps with ML, covering RTL, netlist, layout, etc.
- Facilitate early-stage chip optimizations
- Not just ML for chip, also ML in chip
- Targeting all major chip design objectives

My vision

Chip design & implementation → Customized ML works → Truly Intelligent Solutions

Power Modeling of RTL [MICRO'21] (Best Paper Award)

Highlights (my work APOLLO)

- Fast and accurate design-time power model handling millions-of-cycles benchmarks in minutes
- Low-cost and accurate runtime on-chip power meter (OPM)
- Unprecedented per-cycle temporal resolution
- Fully automated development process for any given design

Overview

APOLLO: Design-time power model and Runtime on-chip power meter. It uses a neural network to predict power based on RTL signals and a hardware OPM to measure actual power.

Key Method

- Automatically select only ~100 RTL signals as input power proxies
- The selection is performed by MCP-based feature pruning algorithm
- Construct OPM hardware without multipliers, with weight quantization

Experiment Setup

Flowchart showing the development process: Design-time power model (RTL to C++) → Pruning (reaching Q non-zero weights) → Re-training (reaching Q retrained weights) → OPM Hardware (using quantized weights). Runtime power model on hardware (HLS to C++) is verified against OPM in RTL.

Experimental Results

- 90-95% accuracy of 17-million-cycles SPEC workload in minutes
- 0.2% area overhead for on-chip meter without accuracy degradation

Power (scaled) vs Timing window index (unit: 1 clock cycle)

Plot showing Power (scaled) over time for various events: dhrystone, mapwrx_cpu, dcache_miss, saxyg_sind, mapwrx_12, lcache_miss, cache_miss, dawp, memcpy_12, throttling_1, throttling_2, throttling_3.

Interconnect of Netlist [ASP-DAC'21] [TCAD (Under-review)]

Highlights (my work Net²)

- First GNN-based method for pre-placement net length estimation
- First ML-based detailed timing estimator before placement

Key Method

- Extract global topology information through partitioning/clustering
- Customized graph attention network (GAT) method

Diagram illustrating the process: Netlist → Clustering → Pre-placement timing (Features, Time^a / Time^f using RF) → Post-placement timing (Net size).

Experimental Results

- High accuracy for individual net length prediction
- Improve slack estimations from commercial tools

WNS on All Designs

Scatter plot of Pre-placement slack (ns) vs Post-placement slack (ns) for WNS on All Designs. Data points are colored by design: ISCAS'89 (black), ITC'99 (grey), Faraday (light blue), OpenCores (dark blue), ANUBIS (brown), Gaisler (green), and Average All (blue). Regression lines show R² values: 0.99 (Our Work), 0.55 (Commercial Tools).

TNS on All Designs

Scatter plot of Pre-placement slack (ns) vs Post-placement slack (ns) for TNS on All Designs. Data points are colored by design: ISCAS'89 (black), ITC'99 (grey), Faraday (light blue), OpenCores (dark blue), ANUBIS (brown), Gaisler (green), and Average All (blue). Regression lines show R² values: 0.97 (Our Work), 0.38 (Commercial Tools).

IR Drop Estimation of Layout [ASP-DAC'21] [ICCAD'20]

Highlights (my work PowerNet)

- First method that claims to perform design-independent fast IR drop estimations, for both vectorless and vector-based estimations
- 30X faster than simulation-based commercial IR drop analysis tools

Key Method

- Time-decomposed power density as input features
- The max estimated IR drop among all time frames as final prediction

Diagram showing the PowerNet architecture: Part of Layout as Input (Cells, Signal propagation, Switch early, Switch late) → Power map at time frame 1, ..., Power map at time frame N → Same CNN (IR₁, ..., IR_N) → Max → IR.

Experimental Results

- Integrated to guide mitigation flow to reduce IR violations by 20-30%
- Interpretability: showing violations triggered at different time frames

ROC AUC

Bar chart comparing ROC AUC for ICCAD'18 and Ours across four time frames (D1, D2, D3, D4) and Average (Ave). Ours consistently shows higher ROC AUC than ICCAD'18.

Design	Mitigation	Violated Cell	# Hotspots
MD1	Before Mitigate	22185	5092
	After Mitigate	17052	3778
MD2	Before Mitigate	31097	3627
	After Mitigate	23941	2489
Improvement		23%	26%
		23%	31%

Where Max takes early time-frame

Two heatmaps showing ground-truth IR drop on the same layout for early time-frames.

Ground-truth IR drop on the same layout

Two heatmaps showing ground-truth IR drop on the same layout for late time-frames.

Where Max takes late time-frame

Two heatmaps showing ground-truth IR drop on the same layout for late time-frames.

Routability Estimation of Layout [ICCAD'18]

Highlights (my work RouteNet)

- First deep learning-based DRV estimator, capturing global information
- Orders-of-magnitude speedup compared with accurate simulations

Overview

Diagram illustrating the process: Layout 1 → Layout 2 → Image classification with CNN (cat / dog) → Layout 1 → Image segmentation with FCN (cat) → Layout 1.

Key Method

- Define novel features capturing macro and long interconnect impact
- 3D input tensor constructed by stacking multiple 2D features

Input Tensor

Diagram showing the construction of a 3D input tensor from multiple 2D feature maps (e.g., 4x4x3).

Experimental Results

- Fast and high-fidelity routability prediction at the same time

Bar chart comparing True Positive Rate for ISPD'17 and Ours across designs D1-D5 and Average. Line chart showing Error with 1st (10³ #DRV) vs Inference time (sec / placement) for various methods: SVM, LR, TR, GR, RouteNet, and RouteNet_w_train.

Parameter Tuning for the Design Flow [ASP-DAC'20]

Highlights (my work FIST)

- First flow tuning method leveraging prior data from other designs
- An approximate sampling strategy which leverages the idea of semi-supervised learning

Experimental Results

- 1.8% area improvement on industrial design compared with best solutions hand-tuned by designers

Setup TNS (ns)

Bar chart comparing Setup TNS (ns) for DAC'13 and Ours across Power, SetupTime, HoldTime, and Ave. Scatter plot showing Setup TNS (ns) vs Area (μm²) for various designs.

Selected Related Publications

My Publications Presented in This Poster

- Zhiyao Xie, et al. "APOLLO: An Automated Power Modeling Framework for Runtime Power Introspection in High-volume Commercial Microprocessors." In **MICRO**, 2021. (**Best Paper Award**)
- Zhiyao Xie, et al. "Pre-placement Net Length and Timing Estimation by Customized Graph Neural Network." In **TCAD**, Under-review.
- Zhiyao Xie, et al. "Net²: A Graph Attention Network Method Customized for Pre-placement Net Length Estimation." In **ASP-DAC**, 2021.
- Zhiyao Xie, et al. "PowerNet: Transferable dynamic IR drop estimation via maximum convolutional neural network." In **ASP-DAC**, 2020.
- Zhiyao Xie, et al. "Fast IR Drop Estimation with Machine Learning." In **ICCAD**, 2020.
- Zhiyao Xie, et al. "FIST: A Feature Importance Sampling and Tree-based Method for Automatic Design Flow Parameter Tuning." In **ASP-DAC**, 2020.
- Zhiyao Xie, et al. "RouteNet: Routability Prediction for Mixed-size Designs using Convolutional Neural Network." In **ICCAD**, 2018.

My Other Publications on This Topic

- Chen-Chia Chang, Jingyu Pan, Tunhou Zhang, Zhiyao Xie, et al. "Automatic Routability Predictor Development using Neural Architecture Search." In **ICCAD**, 2021.
- Rongjian Liang, Zhiyao Xie, et al. "Routing-free Crosstalk Prediction." In **ICCAD**, 2020.
- Yu-Hung Huang, Zhiyao Xie, et al. "Routability-driven Macro Placement with Embedded CNN-based Prediction Model." In **DATE**, 2019.

Contact me for any further discussion: zhiyao.xie@duke.edu

Zhiyao Xie – ECE Department – Duke University

Email: zhiyao.xie@duke.edu