VIRTUAL: Vector-based Dynamic Power Estimation via Decoupled Multi-Modality Learning

Yuntao Lu[1], Mingjun Wang[1,2], Yihan Wen[1], Boyu Han[3], Jianan Mu[2], Huawei Li[2], Bei Yu[1]

[1]The Chinese University of Hong Kong
[2]Institute of Computing Technology, CAS
[3]Stanford University

October 27, 2025

Outline

At a Glance

Vector-based Dynamic Power Estimation via Decoupled Multi-Modality Learning

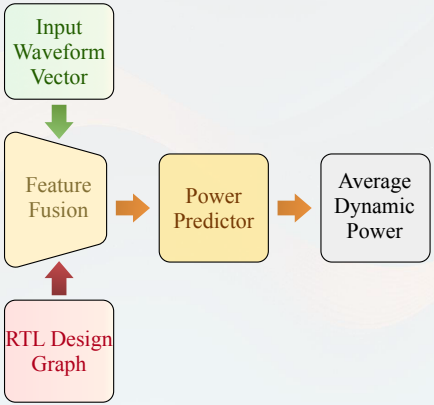

Fig. 1 An overview architecture of VIRTUAL.

Innovations

1. Fast end-to-end power prediction.
   - Eliminate gate-level simulation.
   - Read input vector and RTL graph.
2. Multi-modal decoupling and fusion.
   - Decouple input vectors & RTL graphs.
   - Eliminate toggle-rate computation.
3. Self-Supervised Contrastive Learning
   - Power-aware vector encoding.
   - Functionally-focused RTL encoding.

Performance Highlights

# RTL Designs	Pearson	MAPE	Speedup
2004	0.84	23.43%	14.27×

Motivation: Traditional Power Analysis Workflow

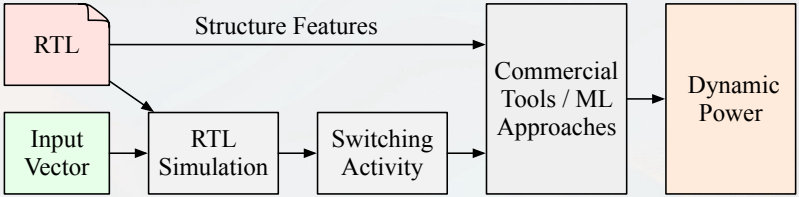

Fig. 2 Traditional power estimation workflow by commercial tools or ML approaches.

Bottlenecks

1. Runtime: Simulation & synthesis hours to days.
2. Iteration: Full flow re-execution of each change.
3. Scalability: Larger designs, longer runtimes.

Impacts

1. Higher feedback latency.
2. Limited optimization iterations.

Motivation: Challenges & Limitations of Power Analysis

Challenges of Traditional Power Analysis Flow

1. RTL/Gate-level(Netlist) simulation.
2. VCD/SAIF files for switching activity.
3. Toggle rate extraction from VCD/SAIF files.

State-of-the-Art (SOTA) AI-driven Power Estimations

Tab. I SOTA AI-based approaches for power estimation.

Method	Netlist Req.	Gate-Level Sim.	RTL Sim.	Toggle Rate Req.	Speed
GRANNITE[1]	Yes	Yes	No	Yes	Slow
GRASPE[2]	Partial	Partial	Partial	Yes	Medium
MasterRTL[3]	No	No	Yes	Yes	Medium

$\Rightarrow$ Approaches rely on simulation results for modeling switching activities for each design.

[1] **Yanqing Zhang, Haoxing Ren, and Brucek Khailany (2020). "GRANNITE: Graph Neural Network Inference for Transferable Power Estimation".** In: *Proc. DAC*. IEEE, pp. 1–6.

[2] **MB Rakesh et al. (2023). "GRASPE: Accurate Post-synthesis Power Estimation from RTL using Graph Representation Learning".** In: *Proc. ISCAS*. IEEE, pp. 1–5.

[3] **Wenji Fang et al. (2023). "MasterRTL: A Pre-synthesis PPA Estimation Framework for Any RTL Design".** In: *Proc. ICCAD*. IEEE, pp. 1–9.

Motivation: Gaps for Fast and Accurate Power Estimation

Limitations of Current AI-driven Methods

- Weak input vector modeling (toggle rates, switching patterns).
- Require simulation for toggle rate extraction.
- Limited generalization across input vectors and RTL designs.

Bridging Gaps

Goal: Enabling fast and accurate power estimation for diverse inputs and RTLs.

- Representations for input vectors and RTL designs.
- Direct RTL-level estimation without simulation.
- Generalize across diverse inputs and RTLs.

Problem Definition

Problem Formula

Given

1. **Input Waveform Matrix**: $W \in [0,1]^{M \times T}$,
   - M: # of input ports.
   - T: # of clock cycles.
2. **RTL Graph**: $G = (V, E)$.
   - V: Set of functional units.
   - E: Set of signal connections between units.

Objective:

- Learn **a mapping function** $F_{power}(W, G) \rightarrow \hat{P}$

VIRTUAL: Framework Overview

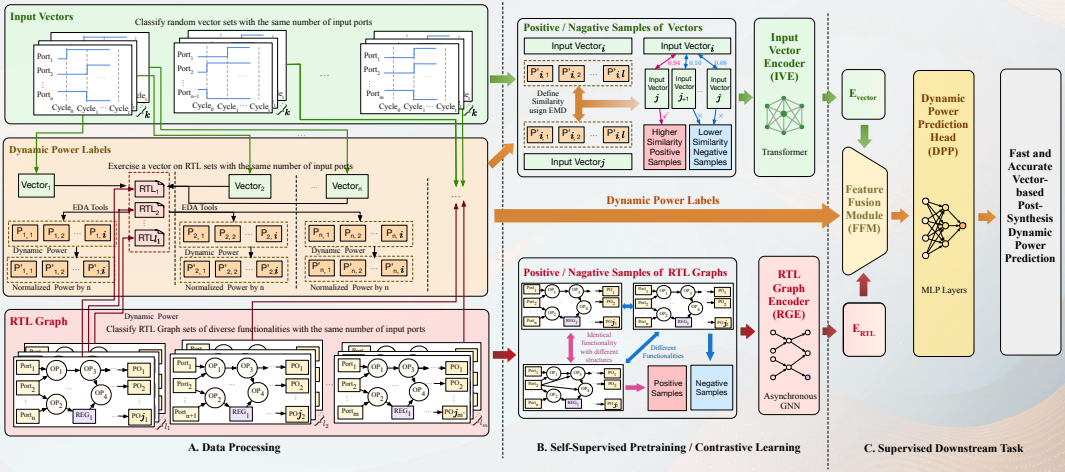

Fig. 3 Overview of the VIRTUAL framework for dynamic power estimation.

VIRTUAL: Three-Stage Learning Strategy

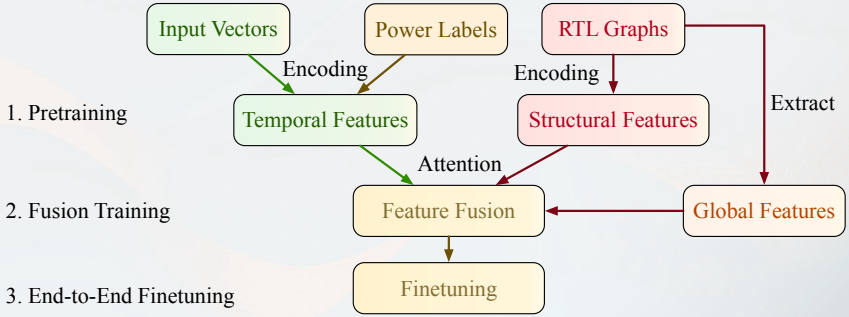

Fig. 4 Three-stage learning strategy of VIRTUAL.

Stage 1: Pretraining

- Power-relevant features.
- Functional-focused features.

Stage 2: Fusion Training

- Temporal features.
- Structural features.

Stage 3: Fine-tuning

- Joint optimized modules.
- Supervised learning.

Input Vector Encoding: Pattern Extraction

Transformer-Based Pattern Extraction

1. **Tokenization**:
   - Split fixed-length segments.
   - Add positional encoding.
2. **Multi-head Self-Attention**:
   - Capture long-range dependencies.
   - Learn ports correlation.
   - Identify switching patterns.
3. **Feed-forward Networks**:
   - Transform attention outputs.
   - Extract power-relevant temporal features.

$\Rightarrow$ Temporal embedding $E_{\text{vector}} \in \mathbb{R}^d$ capturing switching patterns.

Input Vector Encoding: Power-Aware Contrastive Learning

Vectors with similar power distributions should have similar embeddings

Pairing Strategy

- Normalization: normalize power distributions for each vector by port count on multiple designs.
- Positive pairs: vectors with similar power distributions.
- Negative pairs: others.
- Learn to push apart dissimilar vectors in embedding views.

InfoNCE Loss Function

$$\mathcal{L}_{\text{vector}} = -\sum \log \frac{\exp(\cos(E_{\text{anchor}}, E_{\text{pos}})/\tau)}{\sum \exp(\cos(E_{\text{anchor}}, E_{\text{neg}})/\tau)}$$

, where τ is temperature, cos is cosine similarity.

Benefits

- Generalize across input waveforms.
- Capture switching patterns.
- Minimal labels required.

RTL Graph Encoding: Structure Extraction

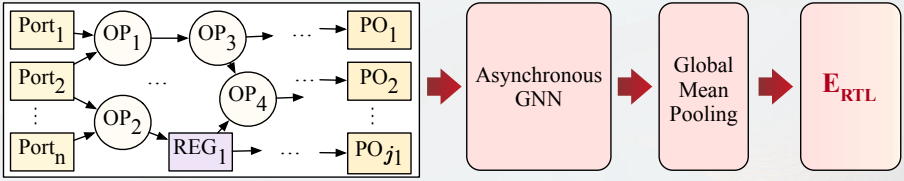

Fig. 5 GNN-based RTL functionality encoding.

RTL Graph Representation

- Nodes: Simple functional units (registers, logic gates, operators).
- Edges: Signal connections (data flow).
- Node features: Unit type, fan-in/out, depth, bit-width, etc.

RTL Graph Encoding: Asynchronous Message Passing

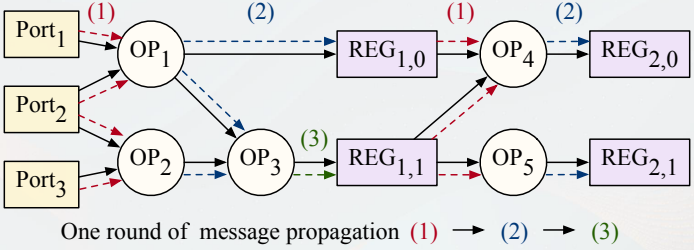

Fig. 6 Asynchronous message passing scheme mimicking signal propagation.

Asynchronous GNN Architecture

- Message Passing: Mimic hardware signal propagation.
- Multi-head Graph Attention: Aggregate neighbor information.
- Temporal Updates: Model sequential signal flow.
- Global Pooling: Aggregate node embeddings to the graph-level.

$\Rightarrow$ Structural embedding $E_{\text{RTL}} \in \mathbb{R}^d$ capturing circuit functionality.

RTL Graph Encoding: Functionality-Focused Contrastive Learning

Functionally equivalent RTL designs should have similar embeddings.

Pairing Strategy

- Positive pairs: Functionally equivalent designs
  - Same logic function, different optimizations.
  - Generated via Design Compiler with different settings.
- Negative pairs: Others.
- Learn to push apart dissimilar graphs in embedding views.

Benefits

- Generalize across RTL designs.
- Capture structural patterns.
- Learn without labels.

VIRTUAL: Feature Fusion & Power Prediction

Feature Fusion Module

1. **Fusion Embeddings**:
   - Temporal embedding: $E_{\text{vector}} \in \mathbb{R}^d$
   - Structural embedding: $E_{\text{graph}} \in \mathbb{R}^d$
   - Global features: $E_{\text{global}} \in \mathbb{R}^k$
2. **Multi-head Cross-Attention**:
   - Learn correlations of E_{vector} and E_{graph}.
   - Weighted combination of both modalities.
3. **Concatenation**:
   - Combine with global features.
   - Unified representation: $E_{\text{fused}} \in \mathbb{R}^{2d+k}$

Power Prediction Head

1. **Architecture**: 2-layer fully-connected neural network.
2. **Supervised Training**:

$$\mathcal{L}_{\text{power}} = \text{MSE}(\hat{P}, P) = \frac{1}{N}\sum_{i=1}^{N}(\hat{P}_i - P_i)^2$$

3. **End-to-End Fine-tuning**:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{power}} + \lambda_1\mathcal{L}_{\text{vector}} + \lambda_2\mathcal{L}_{\text{graph}}$$

Inference

- **End-to-End Prediction**: RTL graph G + Input vectors $W \rightarrow$ average dynamic power.

Evaluation: Experimental Setup

Datasets

- **Circuits**: 2004 RTL designs from OpenCores, HuggingFade, and Github.
  - Scalability: $\sim$ 500 to $\sim$ 30,000 nodes.
  - Functionality: arithmetics, controllers, communication, etc.
- **Input Vectors**: 100 vectors per circuit.
  - $\sim$ 60 to $\sim$ 1,000 input ports.
  - 1,000 clock cycles per vector.
- **Ground Truth**: Average dynamic power values (switching + internal) of Design Compiler.

Metrics

- **PCC**: Correlation measurement.
- **R^2**: Goodness of fitness.
- **MAPE**: Relative error.
- **Speedup**: Runtime over traditional flow.

Evaluation: Performance Summary

Tab. II Performance comparison between VIRTUAL AND SOTA AI-based methods

Method	Circuit-based			Vector-based		
	PCC	R^2	MAPE	PCC	R^2	MAPE
MasterRTL	0.786	0.752	30.85%	0.757	0.692	30.79%
GRASPE	0.815	0.775	29.72%	0.780	0.647	30.41%
VIRTUAL w/o Contrastive	0.757	0.729	29.09%	0.745	0.765	26.48%
VIRTUAL (Full)	**0.842**	**0.798**	**23.43%**	**0.834**	**0.776**	**24.56%**

- Strong correlation with ground truth (PCC: 0.842).
- Relative low error across designs (MAPE: 23.43%).
- Significant speedup over the traditional workflow (14×).

Observations

- VIRTUAL outperforms SOTA AI-based methods in PCC and MAPE.
- Contrastive learning improves performance.
- Strong generalization across designs and inputs.

Conclusions & Future Work

Conclusions

- **Runtime**: Eliminates synthesis/simulation for power analysis.
- **Effectiveness**: Demonstrates the effectiveness of multi-modal contrastive learning.
- **Rapid Feedback**: Enables design optimizations with rapid feedback.

Future Work

- **Scalability**: Handle larger designs (thousands of nodes).
- **Generalization**: Few-shot learning for unseen design families.
- **Transferability**: Transfer learning across different technology nodes.

Thanks!

Q&A