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

***Abstract*—Dynamic power analysis in digital integrated circuits (ICs) conventionally relies on gate-level synthesis and simulation, creating a critical bottleneck in iterative design flows. We propose VIRTUAL, a multi-modality learning framework for rapid post-synthesis dynamic power estimation directly from Register-Transfer Level (RTL) implementations and input waveform vectors, eliminating the need for gate-level synthesis and extensive simulation. By decoupling features from input port waveforms and RTL implementations, VIRTUAL employs a transformer-based encoder to extract temporal patterns from input port waveforms and a graph neural network (GNN) to capture structural and functional dependencies within RTL implementations. Through self-supervised contrastive learning across sequential and graph modalities, the framework learns robust power-relevant representations with minimal labeled data. Subsequently, VIRTUAL refines the multi-modality embeddings using a lightweight fusion module and a power prediction head, enabling dynamic power estimation for fixed clock periods within seconds or minutes. Experimental evaluations demonstrate approximately a Pearson correlation coefficient (PCC) of 0.842 and a mean absolute percentage error (MAPE) of 23.43%, while achieving $14.27\times$ speedup compared to traditional gate-level power analysis workflows. Experimental results across diverse RTL designs and input port waveforms validate that our proposed learning-based approach maintains the accuracy while significantly reducing design iteration time, transforming hours of synthesis and simulation into minutes of direct prediction.**

I. Introduction

As the complexity of IC designs increases with continued semiconductor technology scaling [1], rapid and accurate analysis of performance, power, and area (PPA) metrics has become essential. Dynamic power consumption critically impacts energy efficiency, thermal management, and system performance [2]. Without timely power feedback during early design stages, architectural optimization becomes constrained, potentially delaying time-to-market [3], [4].

Traditional EDA design flows synthesize RTL implementations into gate-level netlists using commercial tools, i.e, Synopsys Design Compiler [5], followed by extensive gate-level simulations with tools, i.e, Synopsys VCS [6] to extract switching activities [7]. To estimate vector-based dynamic power consumption of RTL designs, engineers use industrial solutions, i.e, Synopsys Prime Power [8], and researchers intend to utilize deep learning and other advanced approaches [9]. Although existing approaches estimate dynamic power accurately, the methodologies require hours and weeks for synthesis and simulation iterations, severely limiting complex design exploration within development constraints [10].

Recently, expertise pursued earlier-stage power prediction to reduce computational overhead [11], [12]. An advanced machine learning (ML) approach, GRANNITE [13], accelerates the performance of power estimation, but it remains unclear what the exact requirements are of gate-level netlists and input port waveform traces. As shown in Fig. 1(a), although the successive RTL-wise ML-based dynamic power estimation methodologies MasterRTL [14] and GRASPE [15] reduce gate-level dependencies [16], these approaches rely on extensive RTL simulations to extract switching activity for toggle rate computation and inadequately model multi-port, multi-cycle patterns of input port waveforms [17]. Combining ML with EDA knowledge has shown promise in reducing simulation overhead, but a truly gate-free solution that maintains robust accuracy across diverse designs remains an open challenge.

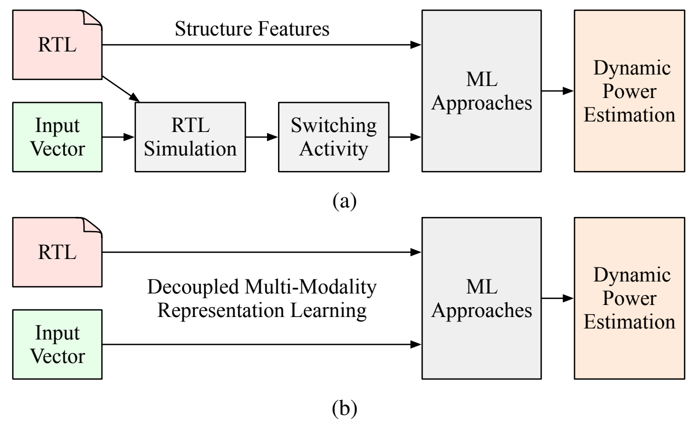


Fig. 1 Comparison of the workflow of vector-based dynamic power estimation between current ML-based approaches in (a) and the proposed VIRTUAL framework in (b).

Three fundamental challenges impact rapid RTL-level dynamic power estimation for multi-port, multi-cycle RTL designs with variant input port waveforms without gate-level synthesis and simulations. At first, bridging temporal waveform sequences across multiple ports with structural RTL connectivity requires specialized multi-modality modeling. Secondly, absence of gate-level simulation precludes traditional toggle-rate metrics, necessitating sophisticated feature extraction to correlate waveform behavior with RTL topology [18]. At last, limited availability of high-quality power labels and significant design variation complicate cross-design generalization [19]. Self-supervised representation learning, particularly contrastive learning techniques, has emerged as a promising solution to identify power-relevant patterns from large waveform vector and RTL graph data with limited or no labels, substantially improving both prediction accuracy and the ability to generalize across different designs and inputs [20], [21], [22].

To address the limitations associated with dynamic power estimation, we present VIRTUAL, a multi-modality learning framework for vector-based dynamic power analysis that operates directly on multi-port waveforms and RTL graphs without gate-level synthesis, as illustrated in Fig. 1(b). Our approach employs a dual-stream architecture composed of a transformer-based encoder (IVE) that processes temporal input port waveform vectors and a GNN-based encoder (RGE) that captures RTL structural dependencies concurrently. The decoupling of distinct representations is integrated using a feature fusion module (FFM) with a multi-head attention mechanism, followed by multiple fully-connected neural networks (NN) as the

*Equal contribution.
†Corresponding author.

dynamic power prediction (DPP) head to predict dynamic power values in fixed clock periods. Through unsupervised pretraining for dual feature encoders of input port vectors and RTL graphs, VIRTUAL conducts supervised fine-tuning encoders and training the FFM and DPP with limited dynamic power labels to deliver predicted dynamic power values within seconds to minutes which is ten times faster than state-of-the-art ML-based dynamic power prediction approach [14], and [15], and orders of magnitude faster than traditional power analysis workflows requiring hours to days by synthesis and simulations. Comprehensive evaluation of VIRTUAL confirms a high accuracy across diverse RTL designs with various input vectors while significantly reducing analysis time. Our proposed dynamic power estimation framework provides designers with an effective tool to expedite power feedback in the early stages of the EDA workflow. Our work makes the following key contributions:

- Decoupling and Fusing Multi-Modality Features. We disengage and jointly extract features of sequential vectors of input port waveforms and RTL graph structures, eliminating time-intensive gate-level synthesis and simulation-based switching activities and toggle rate extraction.
- Self-Supervised Contrastive Learning of Feature Extraction. We develop power-aware contrastive learning strategies for input vectors and functionality-preserving RTL graphs, which learns transferable representations that generalize across designs and input patterns.
- Efficient End-to-End Vector-Based Dynamic Power Prediction. Through attention-based multi-modal fusion, our framework produces accurate predicted dynamic power values within seconds to minutes, drastically reducing conventional simulation bottlenecks over conventional simulation-centric flows.
- High Generalization and Industrial Feasibility. Experimental evaluation demonstrates 0.842 PCC, 0.836 R^2, and 23.43% MAPE for vector-based dynamic power estimation, achieving $14.27\times$ speedup over conventional workflows while maintaining industrial accuracy requirements.

II. Background and Related Works

A. Limitations of Traditional Gate-Level Power Analysis

In conventional digital circuit development, designers typically synthesize RTL code into gate-level netlists utilizing commercial tools [23], such as DC [24]. Gate-level simulations on ICs are conducted using EDA tools, notably Synopsys VCS [6] and PrimePower [8], to collect toggle rates and gate-level circuit parameters for dynamic power estimation [25]. Although this flow is physically valid, it presents several practical limitations. Firstly, both synthesis and gate-level simulations require significant computational resources [26], resulting in extended turnaround times. This overhead increases considerably with larger circuit sizes. Secondly, any modification to the design or variations in the fabrication process necessitate a repetition of the entire flow, which further extends the design cycle [27]. Consequently, the increased time and cost associated with this process hinder the provision of precise and timely power feedback during architectural exploration or early optimizations, thereby limiting the efficacy of multi-version and multi-scenario design iterations.

B. ML-based Power Estimation Approaches at RTL

To reduce reliance on gate-level simulation and synthesis, recent research conducted in both academic and industrial settings has introduced a variety of ML methodologies. Especially, these approaches leverage deep learning techniques for power modeling and estimation. A summary of select representative works is presented in TABLE I, which emphasizes their dependencies on netlist generation, gate-level simulation, RTL simulation, and the modeling capabilities concerning input vector sequences. As illustrated in TABLE I, recent ML-based strategies seek to bypass or expedite specific components of the gate-level simulation process. For example, GRANNITE [13] utilizes both netlist information and gate-level simulation data to train its predictive model, thereby mitigating some iteration overhead. Nevertheless, it continues to depend on comprehensive netlists and parasitic extraction. Similarly, GRASPE [15] employs a GNN-based methodology on unoptimized and unmapped post-translated (UUPT) circuits to infer average toggle rates, partially alleviating synthesis requirements while maintaining specific dependencies on netlist transformation and RTL simulations. And MasterRTL [14] extends the analysis to the RTL domain, thereby reducing the reliance on gate-level netlists; however, it still requires RTL-level simulations to capture switching activities. Collectively, these solutions underscore the potential of integrating ML within electronic design automation (EDA) workflows. Despite significant advancements in accuracy, speed, and generalizability, the pursuit of a robust estimation that is independent of netlist requirements and does not necessitate extensive simulation remains a pressing challenge that calls for more comprehensive solutions.

C. Challenges in Multi-Modality Spatiotemporal Modeling

Estimating dynamic power directly from RTL representations involves intricate spatiotemporal considerations. Contemporary circuits frequently encompass multiple input ports and vectors that extend over numerous clock cycles [28], which renders the modeling of input vectors inherently high-dimensional. Traditional netlist-based methodologies utilize explicit interconnections and parasitic parameters to assess toggle activity [29]. Conversely, a netlist-free approach must derive these behaviors from the graph structure of the RTL, which includes elements such as registers, arithmetic logic units, and controllers, in addition to the temporal patterns associated with input vectors. Basic metrics, such as the average switching rate, prove inadequate for effectively capturing the complex inter-dependencies that exist across various ports and time intervals. Moreover, real-world scenarios display significant variability in design styles and input stimuli; however, high-fidelity labels, which can be obtained through commercial tools, i.e., PrimePower [8], are often limited and expensive. This limitation necessitates the development of robust cross-design generalization techniques and the ability to learn from a paucity of labeled data [30], [31]. Conventional supervised methods frequently struggle to adapt to novel designs or unfamiliar vector distributions, highlighting the necessity for advanced representation learning techniques that can extract discriminative features under conditions of weak supervision [32].

D. Decoupling Spatiotemporal Features with Self-Supervised Learning

To address the complexities associated with multiple ports, varying dimensions, and diverse distributions inherent to this problem, it may be advantageous to integrate spatiotemporal decoupling with self-supervised [20] and contrastive learning methodologies [33]. An aspect focuses on the time-series modeling of input vectors, employing transformer-based architectures [34] to capture flipping patterns and temporal dependencies effectively. Conversely, another aspect utilizes graph encoding, employing GNNs for RTL functional units and their interconnections [35]. By leveraging extensive unlabeled or partially labeled datasets, it is possible to learn generalizable representations through the pretraining of these two encoders in a self-supervised or weakly supervised manner. Following this, a fine-tuning phase can be implemented using a limited set of precise power labels to

TABLE I Comparison among various dynamic power estimation approaches.

Approach	Netlist Required	Gate-Level Simulation	RTL Simulation	Input Vector Modeling	End-to-End Performance
GRANNITE [13]	Yes	Yes	No	Weak	Slow
GRASPE [15]	Partial	Partial	Partial	Weak	Medium
MasterRTL [14]	No	No	Yes	Weak	Medium
VIRTUAL (Ours)	**No**	**No**	**No**	**Strong**	**Fast**

align the pre-trained embeddings with actual dynamic power values, thereby facilitating accurate predictions on previously unseen circuits or vectors. Importantly, this pipeline can significantly reduce the power estimation cycle from several days to just seconds or minutes, while maintaining near gate-level accuracy. This advancement not only accelerates design iterations but also enhances the potential for deep front-end optimizations.

III. Problem Definition

Given an RTL implementation of a digital IC abstracted as a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where nodes $v \in \mathbf{V}$ represent functional units, including registers, arithmetic units, and control blocks, and edges $e \in \mathbf{E}$ denote signal connections between functional units. For circuits with M input ports operating over T clock cycles, we can represent the input sequences as a matrix $\mathbf{W} \in \mathbb{R}^{M\times T}$, we represent input vectors for waveforms as $\mathbf{W} \in \mathbb{R}^{M\times T}$, where each column captures logic values across all ports at a specific cycle.

The primary objective is to learn a mapping function $F(\mathbf{W}, \mathbf{G}) \mapsto \hat{P}$ that accurately predicts dynamic power consumption P with input port vectors $\mathbf{W}$ and circuit RTL graphs $\mathbf{G}$ without gate-level synthesis or simulation of commercial EDA tools. Physically, dynamic power comprises

$$P_{\text{dynamic}} = P_{\text{internal}} + P_{\text{switching}}, \tag{1}$$

where P_{dynamic} denotes the total dynamic power of circuits with an input vector, P_{internal} denotes the internal power breakdown which is incurred by short-circuit current and internal capacitance charging, and $P_{\text{switching}}$ is the switching power when charging and discharging load capacitance during logic transitions of signal switching. The switching power follows

$$P_{\text{switching}} \approx \alpha \cdot C_{\text{load}} \cdot V_{\text{dd}}^2 \cdot f_{\text{clk}}, \tag{2}$$

with switching activity α, load capacitance C_{load}, supply voltage V_{dd}, and clock frequency f_{clk}.

Given a dataset $\mathbf{D} = (\mathbf{W}_i, \mathbf{G}_i, P_i)$, where $i \in \{1, 2, \cdots, N\}$ containing N samples of input vectors $\mathbf{W}_i$, RTL graphs $\mathbf{G}_i$, and ground-truth dynamic power labels P_i derive from commercial EDA tools. To accurately estimate dynamic power, we model the relationship between dynamic power results, input vectors, and RTL graphs. The objective is to minimize the prediction error between the predicted dynamic power value $\hat{P}_i$ and the actual power value P_i, as shown in Equation (3), therefore we ensure that our predicted dynamic power values closely match the exact measurements, enabling accurate power estimation during inference on unseen circuits and diverse input patterns of ports.

$$\min \quad \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(F(\mathbf{W}_i, \mathbf{G}_i; \Theta), P_i), \tag{3}$$

where Θ represents model parameters and $\mathcal{L}$ is the loss function.

The key challenge lies in directly extracting power-relevant structural features from structural RTL graphs and temporal waveform patterns from input vectors without traditional toggle-rate from simulation switching activities, requiring effective multi-modality representation learning to capture the complex relationship between circuit topology, input switching behavior, and resulting power consumption.

IV. VIRTUAL Framework

A. Overview of VIRTUAL Framework

VIRTUAL employs a dual-stream architecture to directly estimate dynamic power from RTL implementations and input waveforms without gate-level synthesis, as illustrated in Fig. 2. The framework decouples the extraction of temporal features from input vectors of waveforms and spatial features from graphs of RTL implementations through specialized encoders, subsequently merging separate representations for the downstream power prediction task.

The IVE processes multi-port waveforms $\mathbf{W} \in \mathbb{R}^{M\times T}$ through transformer-based sequential modeling, extracting temporal embeddings $\mathbf{E}_{\text{vector}}$ that capture critical switching patterns for dynamic power consumption. Concurrently, the RGE analyzes circuit structure $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ by an asynchronous GNN model, producing structural embeddings $\mathbf{E}_{\text{RTL}}$ that encode functional dependencies and connectivity patterns for power-relevant features.

Both encoders undergo a self-supervised learning approach by conducting contrastive learning in input vectors and RTL graph samples. For IVE, we leverage power distribution similarity across diverse circuits to identify functionally equivalent waveform patterns. For RGE, we exploit functional equivalence among RTL graphs with different structural implementations. The pretraining strategy enables robust feature extraction from weakly labeled input vector samples and unlabeled RTL graph samples, addressing the scarcity of ground-truth power labels.

The FFM integrates temporal embeddings from input vectors and structural embeddings from RTL graphs using multi-head attention mechanisms, followed by a DPP head that contains a fully-connected multi-layer perceptron (MLP) to generate predicted power estimates $\hat{P}$. During supervised fine-tuning, limited labeled data aligns learned representations with actual power measurements from commercial EDA tools.

Given an unseen pair of an input vector and an RTL graph, VIRTUAL produces power estimates within seconds to minutes, which achieves orders-of-magnitude speedup over traditional workflows with synthesis and simulation while maintaining accuracy over actual power labels.

B. Data Processing

To prepare essential data for the VIRTUAL framework, we engage the data processing phase in three primary tasks of generating input vectors, constructing RTL simple operator graphs (SOGs), and acquiring ground-truth dynamic power labels from EDA tools. Algo. 1 outlines the systematic data processing pipeline. We partition RTL implementations according to the number of input ports M to ensure consistent dimensionality within groups. For each group, we randomly generate input vectors $\mathbf{W} \in \mathbb{R}^{M\times T}$ where $T = 1000$ clock cycles, and i-th row of $\mathbf{W}$ corresponds to an input sequence with the length of T containing only 0 or 1 logic values for the i-th input port of a circuit, retaining high-toggle-rate vectors to maximize power variation

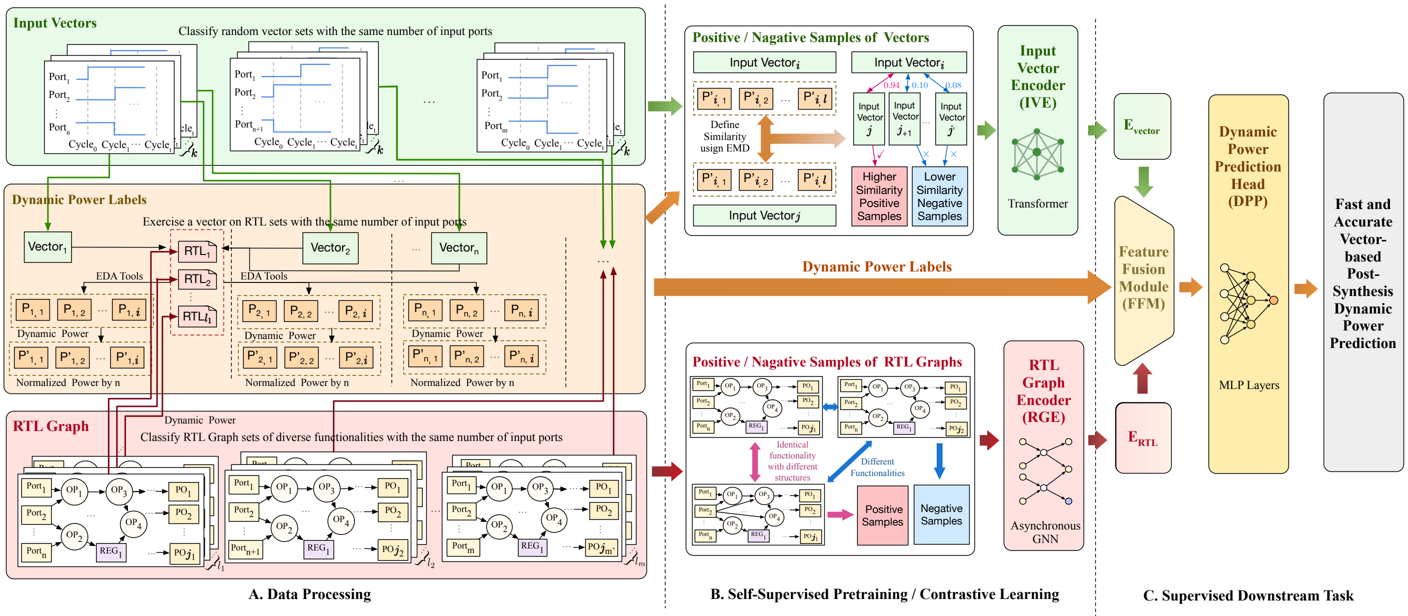


Fig. 2 An overview of the VIRTUAL framework that comprises three processing phases. (A) A data processing phase to prepare input vectors, RTL graphs, and ground-truth power labels in (A). A self-supervised learning phase utilizes contrastive learning for decoupling feature extraction from input vectors and RTL graphs in (B). Finally, a supervised fine-tuning phase for the downstream task to accurately predict dynamic power values in (C).

coverage. In a group, we generate $K = 100$ input vectors of each size of input port count M, resulting in a set of input vectors $\mathcal{W} = \{\mathbf{W}_i\}$, where $i \in \{1, 2, \cdots, K\}$ for each group of circuits.

To represent the RTL implementations, we parse the source RTL code into a simplified SOG representation, where nodes correspond to functional units, and edges denote signal propagations. Each node is characterized by features of cell type (CellType), the number of fan-in/fan-out of the unit (FanIn/FanOut), and topological depth in the graph (DPI/DPO), as summarized in TABLE II. The feature abstraction captures essential structural properties for GNNs to conduct power modeling without gate-level details.

To obtain ground-truth dynamic power labels, we utilize a streamlined flow that includes RTL synthesis by Synopsys Design Compiler, gate-level simulation using Synopsys VCS, and power analysis using Synopsys Prime Power. In addition, we reserve switching activities through VCD/SAIF files from VCS for other existing power prediction frameworks. For each design and corresponding input vector pair, to scale the variance, we normalize power values by the number of nodes in an RTL SOG as $P' = \frac{P}{|\mathbf{V}|}$, enabling consistent learning across circuits of varying complexity while maintaining the relative distribution of power values.

C. Self-Supervised Feature Extraction

To extract power-relevant features from both multi-port input vectors and RTL graphs, we adopt a contrastive learning strategy under a self-supervised or weakly supervised setting. Specifically, we decompose the challenge into two distinct learning streams. The first is IVE contrastive learning, which focuses on time-series input vectors, while the second is RGE contrastive learning, which emphasizes circuit-structure embeddings. By jointly pretraining these two modalities, the framework can learn power-sensitive representations efficiently.

For the IVE pretraining by contrastive learning approach, we employ a strategy to discover toggling patterns from unlabeled and weakly labeled input vectors. Algo. 2 formally presents the procedure,

Algorithm 1: Data Processing and Power Label Generation

Input: RTL designs $\mathbf{G}$
Output: A set of input vector sequence $\mathbf{W}$, SOG graphs $\mathbf{G}'$, original power labels $\mathbf{P}$, normalized power labels $\mathbf{P}'$

```
Partition G into G_i, where i ∈ {1, 2, ..., M} based on M input ports;
W, G', P, P' ← ∅;
for each G_i do
    Generate {W_j}, where j ∈ {1, 2, ··· , K}, and
      W_j ∈ ℝ^{i×1000} ; // Fixed sequence length as 1000
    W_i ← vectors with high toggle rates for G_i;
    W ← W ∪ W_i;
    for each G do
        G'_G ← (V_G, E_G) with node features
          [CellType, FanIn, FanOut, DPI, DPO] ;
          // Construct SOG graph
        G' ← G' ∪ {G'_G};
        Netlist_G ← DesignCompiler(G);
        for each W do
            /* VCS→SAIF→PowerValues */
            VCD ← VCS(Netlist_G, W);
            P_{G,W} ← PrimePower(Netlist_G, VCD);
            P ← P ∪ {(G, W, P_(G,W))} ; // Original power value
            P' ← P' ∪ {(G, W, P_(G,W)/|V'_G'|)} ;
              // Normalized power value per node
return (W, G', P, P');
```

which involves acquiring power profiles for each input vector, computing cross-port similarities in terms of Earth Mover's Distance (EMD), and performing contrastive training on all input vectors. With an

TABLE II Node features of RTL SOGs.

Features	Type	Width
SOG Cell Type (CellType)	One-hot	12
Fan-out Number (FanIn)	Integer	1
Fan-in Number (FanOut)	Integer	1
Depth Per Input (DPI)	Integer	1
Depth Per Output (DPO)	Integer	1
Total		16

input vector $\mathbf{W}_i \in \mathbb{R}^{M\times T}$ with the number of input port counts M. The available and functionality diverse RTL designs are partitioned into groups $\{\mathbf{G}_k\}$, each having a distinct port count k. For $\mathbf{W}_i$, the matching group $\mathbf{G}_k$ satisfies $k = c$. Averaging power across each design $\mathbf{G}_{k,c} \in \mathbf{G}_k$ yields a normalized vector of the power distribution,

$$\mathbf{P}'(\mathbf{W}_i) \; = \; \Big[\frac{\text{Power}(\mathbf{G}_{K,C}, \mathbf{W}_i)}{\text{Nodes}(\mathbf{G}_{k,c})}\Big]_{j=1}^{|\mathbf{G}_k|}. \tag{4}$$

Once each vector $\mathbf{W}_i$ has its corresponding normalized power profile $\mathbf{P}'(\mathbf{W}_i)$, the EMD between $\mathbf{P}'(\mathbf{W}_i)$ and $\mathbf{P}'(\mathbf{W}_j)$ is measured for all pairs of vectors, regardless of their port counts. The distance is then normalized to define

$$\text{S}(\mathbf{W}_i, \mathbf{W}_j) \; = \; 1 \; - \; \frac{\text{EMD}\big(\mathbf{P}'(\mathbf{W}_i), \mathbf{P}'(\mathbf{W}_j)\big)}{\max_{(i',\, j')} \text{EMD}\big(\mathbf{P}'(\mathbf{W}_{i'}), \mathbf{P}'(\mathbf{W}_{j'})\big)}, \tag{5}$$

leading to a uniform cross-port similarity metric. Two vectors $\mathbf{W}_i$ and $\mathbf{W}_j$ form a positive pair if $\text{S}(\mathbf{W}_i, \mathbf{W}_j) > \delta$, and a negative pair if $\text{S}(\mathbf{W}_i, \mathbf{W}_j) \le \delta'$.

To capture deep toggling behaviors, each $\mathbf{W}_i$ is treated as an $M \times T$ time-series with the port count M and sequence length T. A transformer-based model with multi-head self-attention extracts a high-dimensional embedding $\text{E}_{\text{vector}} \in \mathbb{R}^{d_{\text{out}}}$, focusing on features closely tied to dynamic power. The training process involves selecting each vector in a mini-batch as an anchor, with its positive/negative samples (as determined above) passed into the InfoNCE loss.

$$\mathcal{L}_{\text{NCE}} = - \sum_{(\text{anchor,pos})} \log \frac{e^{\frac{\cos(\text{E}_{\text{anchor}}, \text{E}_{\text{pos}})}{\tau}}}{e^{\frac{\cos(\text{E}_{\text{anchor}}, \text{E}_{\text{neg}})}{\tau}}}, \tag{6}$$

where E_{anchor}, E_{pos}, and E_{neg} are the embeddings of the anchor, positive, and negative samples, respectively. $\cos(\cdot,\cdot)$ is the cosine similarity metric, and τ is a temperature hyperparameter.

The self-supervised contrastive pretraining produces robust power-oriented representations in the encoder. We can utilize the pretrained weights and transfer to downstream supervised tasks, accelerating convergence and preserving accuracy even in scenarios with limited labeled data.

In parallel with the pretraining of IVE, we conduct another pre-training task on RTL graphs by a contrastive learning approach to capture discriminative structural features tied to dynamic powers. As depicted in Fig. 3(b), each RTL is parsed into a directed SOG graph $\mathbf{G}' = (\mathbf{V}', \mathbf{E}')$. In the RGE, we adopt an asynchronous propagation GNN to approximate the forward signal propagation in the circuit. Each node contains features including logic type, the number of fan-in and fan-out, and depth in graphs. By multi-head graph attention updates, the model merges information from neighbors of each node and eventually pools node embeddings into a single circuit-level embedding $\text{E}_{\text{RTL}} \in \mathbb{R}^{d_{\text{out}}}$.

We form positive pairs for RTL designs through the same and equivalent functional behavior, and negative pairs for those with significantly divergent functionalities. Conceptually, E_{anchor} and E_{pos} should be near each other if their corresponding RTL circuits are

Algorithm 2: Contrastive learning approach of IVE with the similarity of power distribution.

Input: A set of vector stimuli $\mathbf{W} = \{\mathbf{W}_i \in \mathbb{R}^{C\times T}\}_{i=1}^N$ with varying port counts C of size of the set N, a set of RTL groups $\{\mathbf{G}_k\}_{k=1}^K$ with the identical number of input ports k for each, and the initial IVE $\text{En}_{\text{vector}}(\cdot; \theta_{\text{vector}})$

Output: Pretrained power-aware encoder parameters θ^*_{vector} of IVE

`//Step 1: Power profiles and similarity of input vectors`

for *each* $k \in \{1, 2, ..., K\}$ **do**
  $\mathbf{W}_{\mathbf{G}_k} \leftarrow \{\mathbf{W}_i \mid \text{PortCount}(\mathbf{W}_i) = \text{PortCount}(\mathbf{G}_k)\}$;
  for *each* $\mathbf{W}_j \in \mathbf{W}_{\mathbf{G}_k}$ **do**
    $\mathbf{P}'(\mathbf{W}_j) \leftarrow [\frac{\text{Power}(\mathbf{G}_{k,l}, \mathbf{W}_j)}{\text{Nodes}(\mathbf{G}_{k,l})}]_{l=1}^{|\mathbf{G}_k|}$;

`//Step 2: Compute pairwise similarities across all vectors crossing various numbers of ports`

for *each* $i \in \{1, 2, ..., N\}$ **do**
  for *each* $j \in \{1, 2, ..., N\}$ *and* $j \neq i$ **do**
    $\mathbf{D}_{\text{EMD}} \leftarrow \text{EMD}\big(\mathbf{P}'(\mathbf{W}_i), \mathbf{P}'(\mathbf{W}_j)\big)$;
    $\mathbf{S}_{i,j} \leftarrow 1 - \frac{\mathbf{D}_{\text{EMD}}}{\max_{(i',\, j')} \text{EMD}\big(\mathbf{P}'(\mathbf{v}_{i'}), \mathbf{P}'(\mathbf{v}_{j'})\big)}$;
    $\mathbf{S}_{j,i} \leftarrow \mathbf{S}_{i,j}$;

`//Step 3: Perform pretraining via contrastive learning on all input vectors`

for *each* iter $\in \{1, 2, \dots, T\}$ **do**
  Sample a batch $B \subset \{1, 2, \dots, N\}$;
  for *each anchor sample* $a \in B$ **do**
    $\mathbf{Pos}_a \leftarrow \{\, b \in B \mid \mathbf{S}_{a,b} > \delta,\; b \neq a \,\}$;
    $\mathbf{Neg}_a \leftarrow \{\, b \in B \mid \mathbf{S}_{a,b} \le \delta',\; b \neq a \,\}$;
    if $|\mathbf{Pos}_a| > 0$ **then**
      $\mathcal{L}_a \leftarrow - \sum_{\text{pos}\in|\mathbf{Pos}_a|} \log \frac{e^{\frac{\cos(\text{En}_{\text{vector}}(\mathbf{W}_a), \text{En}_{\text{vector}}(\mathbf{W}_{pos}))}{\tau}}}{\sum_{\text{neg}\in|\mathbf{Neg}_a|} e^{\frac{\cos(\text{En}_{\text{vector}}(\mathbf{W}_a), \text{En}_{\text{vector}}(\mathbf{W}_{neg}))}{\tau}}}$;
  $\theta_{\text{vector}} \leftarrow \theta_{\text{vector}} \; - \; \eta \, \nabla_{\theta_{\text{vector}}} \Big(\frac{1}{|B|} \sum_{a\in B} \mathcal{L}_a\Big)$;

`/* Step 4: Return the pretrained IVE */`

return θ_{vector};

functionally similar. In contrast, functionally disparate RTL circuits should map to distant embeddings. Identical to the IVE, we also employ an InfoNCE-based objective function Equation (6) to train the GNN encoder, which clusters functionally similar circuits and separates functionally distinct ones in the embedding space. After this self-supervised training, the RGE can rapidly encode a continuous representation E_{RTL} for any design, facilitating multi-modal fusion with the IVE embeddings in the successive downstream tasks.

D. Supervised Dynamic Power Prediction

After pretraining both IVE and RGE via self-supervised contrastive learning, the model is equipped to extract generalizable representations of time-series input vectors and RTL structures. To achieve power predictions comparable to industrial EDA tools, it is necessary to conduct further end-to-end supervised fine-tuning on samples that possess ground-truth dynamic power labels. Algo. 3 summarizes the fine-tuning procedure.

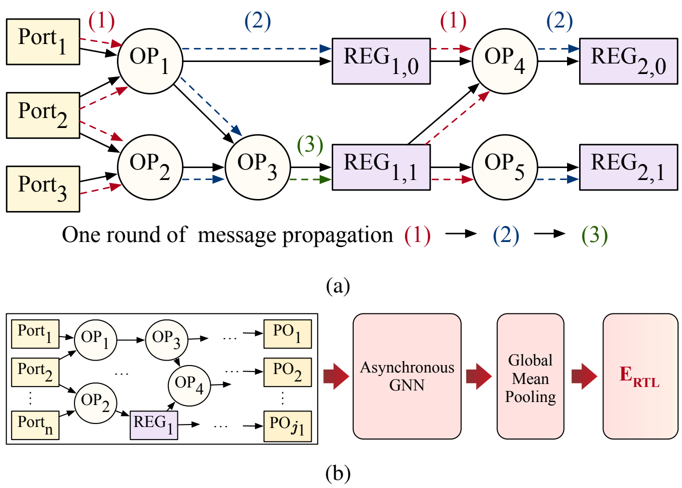


Fig. 3 Processing of GNN-based RTL graph encodings with an asynchronous message-passing scheme that models signal updates of dynamic power simulation flow in (a), and the node embeddings that aggregate through a global mean pooling layer to produce RTL SOG graph embeddings.

Algorithm 3: Multi-modality fine-tuning for RTL dynamic power prediction

Input: Pre-trained IVE and RGE $\mathrm{En}_{\text{vector}}(\cdot;\theta_{\text{vector}})$, $\mathrm{En}_{\text{RTL}}(\cdot;\theta_{\text{RTL}})$, and dynamic power labels $\mathbf{P}$, the sample set $\mathbf{D} = \{(\mathbf{W}_i, \mathbf{G}_i, P_i)\}_{i=1}^{N}$

Output: Optimized parameter set $\Theta^\star = \{\theta^\star_{\text{vector}}, \theta^\star_{\text{RTL}}, \theta^\star_{\text{fuse}}, \theta^\star_{\text{power}}\}$

Initialize the fusion module $\mathrm{Fusion}(\cdot,\cdot;\theta_{\text{fuse}})$ and predictor $\mathrm{Predictor}(\cdot;\theta_{power})$;
$\Theta \leftarrow \{\theta_{\text{vector}}, \theta_{\text{RTL}}, \theta_{\text{fuse}}, \theta_{\text{power}}\}$;
for *each iteration* $iter \in \{1, 2, \ldots, T\}$ **do**
  for *each batch data* $\mathcal{B} \subset \mathcal{D}$ **do**
    $\mathcal{L}_{\text{power}} \leftarrow 0$;
    for *each sample set* $(\mathbf{W}_i, \mathbf{G}_i, P_i), i \in \{1, 2, \ldots, |\mathcal{B}|\}$ **do**
      $\mathrm{E}_{\text{vector}} \leftarrow \mathrm{En}_{\text{vector}}(\mathbf{W}_i;\theta_{\text{vector}})$; `// Encode temporal features by the IVE`
      $\mathrm{E}_{\text{RTL}} \leftarrow \mathrm{En}_{\text{RTL}}(\mathbf{G}_i;\theta_{\text{RTL}})$; `// Encode structural features by the RGE`
      $\mathrm{E}_{\text{fuse}} \leftarrow \mathrm{Fusion}(\mathrm{E}_{\text{vector}}, \mathrm{E}_{\text{RTL}};\theta_{\text{fuse}})$; `// Cross-modality feature fusion`
      $\hat{p} \leftarrow \mathrm{Predictor}(\mathrm{E}_{\text{fuse}};\theta_{\text{power}})$; `// Predict dynamic power values`
      $\mathcal{L}_{\text{power}} \leftarrow \mathcal{L}_{\text{power}} + (\hat{P} - P)^2$; `// Accumulate MSE`
    $\mathcal{L}_{\text{power}} \leftarrow \mathcal{L}_{\text{power}}/|\mathcal{B}|$;
    $\Theta \leftarrow \Theta - \eta\, \nabla_\Theta \mathcal{L}_{\text{power}}$; `// Parameter optimization`
return Θ;

The pretrained IVE generates temporal feature embeddings $\mathrm{E}_{\text{vector}} \in \mathbb{R}^{d_{\text{vec}}}$, while the pretrained RGE produces structural feature embeddings $\mathrm{E}_{\text{RTL}} \in \mathbb{R}^{d_{\text{RTL}}}$. These two modalities jointly characterize the time-domain switching patterns and the circuit functional topology. For feature fusion, VIRTUAL employs a multi-head attention-based mechanism that maps both embeddings to a common feature space through linear projection layers and then applies self-attention to capture fine-grained interactions between temporal and structural features. Additionally, the model incorporates a circuit characteristics encoder that processes global attributes, such as node count, port count, and vector 1-ratio, encoding these features through dedicated feed-forward layers before concatenating them with the attention outputs. Finally, through layer normalization and non-linear transformations, the model generates a highly informative fused representation that effectively preserves the complementary information from both modalities and global circuit characteristics, providing a comprehensive yet compact feature representation for subsequent power prediction tasks.

Given the fused feature embeddings E_{fuse}, a multiple fully-connected neural network performs the final dynamic power regression task. The training objective typically adopts minimizing the MSE loss in Equation (7), driving the model to align with real power measurements via back-propagation.

$$\mathcal{L}_{\text{pred}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{P}_i - P_i)^2, \tag{7}$$

where $\hat{P}_i = F_{power}(\mathbf{W}_i, \mathbf{G}_i)$ is the predicted dynamic power for the i-th sample.

The pretrained parameters from IVE and RGE are loaded and jointly optimized with the newly initialized fusion module and dynamic power predictor in an end-to-end manner. The overall fine-tuning procedure consists of (1) Sampling and Encoding. For each mini-batch with an input vector, an RTL graph, and a total dynamic power label $\{(\mathbf{W}_i, \mathbf{G}_i, P_i)\}$, the IVE and RGE produce the respective temporal and structural embeddings; (2) Feature Fusion. The fusion module combines these embeddings (plus optional circuit parameters) into $\mathrm{E}_{(fuse,i)}$; (3) Power Regression. Through the predictor, the model computes the predicted power $\hat{P}_i$ and then evaluates the regression loss $\mathcal{L}_{\text{pred}}$; (4) Weight Updates. The gradient of $\mathcal{L}_{\text{pred}}$ updates the entire pipeline, including pretrained modules, ensuring consistent adaptation to labeled data; (5) Inference. Once training concludes, the refined model can infer dynamic power for any new input pair $(\mathbf{W}, \mathbf{G})$ in seconds or minutes, substantially reducing reliance on time-consuming gate-level flows.

By aligning the self-supervised representations with a small number of ground-truth power labels, VIRTUAL achieves accurate cross-design, cross-scenario dynamic power estimation, offering a fast and flexible alternative to gate-level simulations.

V. Experiments

A. Experimental Setup

Dataset and Benchmarks. We collected 2004 RTL designs with diverse functionalities and complexities, ranging from arithmetic units to medium-scale processor cores, and with sizes from small to large-scale designs. These designs were collected from the open-source GitHub, Hugging Face [36], OpenCores [37], and RISC-V projects. Each module received hundreds of random input vectors of length T, where $T = 1000$ in our work, denoting 1000 clock cycles and covering a broad range of toggle rates. To acquire the ground-truth dynamic power, we synthesized RTL designs using DC [24] and simulated them at the gate level via VCS [6] to generate VCD or SAIF files. Furthermore, we utilize the commercial power analysis tool PrimePower [8] to measure the detailed dynamic powers of our designs. All samples were split into training, validation, and testing sets in a 75%/15%/10% ratio, ensuring that the test set contained unseen RTLs and new input vectors.

Implementation and Environment. All experiments are executed on a server equipped with 8 NVIDIA A100 GPUs and 80 GB of memory. The framework utilizes PyTorch and PyTorch Geometric (PyG) to conduct distributed data parallel training. The optimizers and schedulers for the learning rate are AdmW and OneCycleLR, with an initial learning rate of 5×10^{-5} and a batch size of 128. The pretraining process of each encoder, which involves vectors and RTL graphs, executes 35 epochs through contrastive learning to ensure convergence, followed by an additional 35 epochs to train the downstream power prediction head using fine-tuned vector and graph encoders. We utilized a transformer-based encoder to embed IVE and a GAT-based Asynchronous Propagation GNN for RGE.

B. Evaluation Metrics and Baselines

Evaluation Metrics. We adopt three complementary criteria for predicted dynamic power over the ground truth: the description of Pearson correlation coefficient (PCC), coefficient of determination (R^2), and mean absolute percentage error (MAPE). PCC measures linear correlation between predictions and truth labels, with higher values implying stronger agreement. R^2 evaluates how much of the variance in the data is explained by the model. MAPE quantifies the prediction error relative to ground truth, which is valuable for indicating practical significance in industrial contexts. Additionally, we measure End-to-End latency for each tested approach across the entire pipeline, encompassing all steps from reading RTL and input vectors to final power estimation.

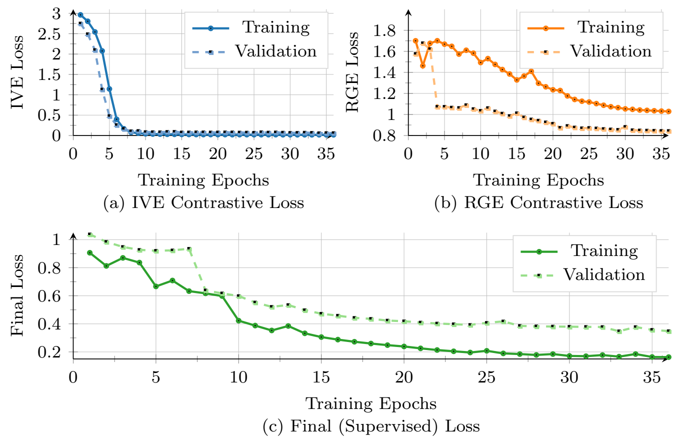
Evaluation Methods. In addition to the proposed VIRTUAL framework, we compare our approach, VIRTUAL, with and without contrastive learning to MasterRTL [14] and GRASPE [15]. We employ the open-source MasterRTL approach, focusing on its GNN-based power module, and reproduce GRASPE to handle RTL sequential circuits. Each register is assigned features as specified in its original paper. The VIRTUAL w/o contrastive pretraining provides an ablation version of VIRTUAL that removes all self-supervised contrastive pretraining, directly training IVE, RGE, FFM, and DPP on labeled data alone, which highlights the contribution of self-supervised contrastive learning.

C. Results and Discussion

In this section, we present quantitative results based on the tables and loss curves provided above. We analyze the proposed VIRTUAL framework alongside the baselines and ablation models in terms of prediction accuracy, speed, and training convergence.

Predictive Performance.. TABLE III reports power estimation outcomes from two complementary viewpoints, reflecting the diverse ways in which matching circuits and input vectors are used. Specifically, TABLE III (a) explores a circuit-based scenario where testing each design against multiple input vectors (more than 100 input vectors), while TABLE III (b) adopts a vector-based perspective by applying a single input stimulus to multiple circuits. This dual analysis not only ensures sufficient statistical samples for correlation measures (e.g., PCC, R^2) but also highlights how the proposed framework handles variation along structural dimensions of RTL graphs and temporal dimensions of input vectors.

In the circuit-based assessment TABLE III(a), VIRTUAL achieves consistently high PCC from 0.806 to 0.863 and R^2 from 0.782 to 0.829, while reducing MAPE down from 23.65% to 28.46%. Advantages are demonstrated on large-scale circuits such as `pfpu_fmul`. By comparison, MasterRTL and GRASPE exhibit weaker correlation and higher error percentages. In contrast, the ablated VIRTUAL w/o



(a) IVE Contrastive Loss (b) RGE Contrastive Loss (c) Final (Supervised) Loss

Fig. 4 Loss curves during self-supervised pertaining. (a) plots the Input Vector Encoder contrastive loss, (b) shows the RTL Graph Encoder contrastive loss, and (c) reports the Supervised Loss.

Contrastive model shows only minor improvements over classical baselines, yet still lags behind the full pipeline. These findings emphasize the benefit of self-supervised contrastive pretraining in extracting robust representations of toggling behavior and structural dependencies.

A similar trend emerges in the vector-based experiments TABLE III(b), where VIRTUAL demonstrates a marked reduction in percentage error, achieving an average MAPE of 23.43%. Notably, even when input vectors vary in port count, toggling frequency, or temporal length, the proposed approach preserves higher PCC and R^2 compared to alternative methods. By effectively blending the Transformer-based Input Vector Encoder with the GNN-driven RTL Graph Encoder, VIRTUAL learns power-critical features across both time-series and structural domains without reliance on time-consuming gate-level simulations.

Collectively, these two complementary evaluations underscore the generalization power of VIRTUAL. The combination of decoupled multi-modal encoders, contrastive self-supervision, and supervised training captures key spatiotemporal cues, thereby yielding robust estimations of dynamic power over a wide variety of circuits and input vectors.

End-to-End Runtime. TABLE IV compares the total runtimes of dynamic power analysis pipelines among Conventional Flow, MasterRTL, GRASPE, and VIRTUAL, each evaluates on large-scale benchmark circuits with 100 input vectors. Conventional Flow (i.e., logic synthesis, simulation, and power analysis) exhibits the highest computational cost, requiring tens of minutes per circuit. By contrast, VIRTUAL delivers a substantial acceleration, averaging a $14.07\times$ speedup. This performance gain stems primarily from avoiding gate-level netlist generation and extensive simulation. Instead, VIRTUAL relies on multi-modal representation learning directly at the RTL and input vector levels. Although MasterRTL and GRASPE also outperform the conventional flow, their partial reliance on simulation or toggling statistics incurs extra overhead compared to VIRTUAL. Notably, in larger circuits (e.g., `diffeq_f_systemC` with over $20,000$ nodes), VIRTUAL achieves greater speed advantages while preserving estimation quality, thereby supporting faster design iterations and power optimization in realistic industrial settings.

Training Convergence and Loss Curves. Fig. 4 displays the loss trajectories for each stage of model training, highlighting the bene-

TABLE III Comparison of Power Estimation Performance from Two Perspectives. (a) Circuit-Based Variation (Multiple Input Vectors per Circuit) and (b) Vector-Based Variation (Multiple Circuits per Input Vector).

a Circuit-Based Performance Across Multiple Input Vectors

Circuit	#Nodes	MasterRTL			GRASPE			VIRTUAL w/o Contrastive			VIRTUAL		
		PCC	R^2	MAPE	PCC	R^2	MAPE	PCC	R^2	MAPE	PCC	R^2	MAPE
`counter`	558	0.756	0.734	32.63%	0.797	0.781	30.78%	0.738	0.701	30.58%	**0.806**	**0.782**	**27.27%**
`tv80_reg`	1,211	0.803	0.767	29.79%	0.818	0.778	29.06%	0.762	0.736	28.38%	**0.836**	**0.805**	**25.67%**
`booth_multiple`	1,600	0.815	0.780	28.63%	0.832	0.791	27.73%	0.776	0.742	27.27%	**0.848**	**0.811**	**24.73%**
`long_to_double`	2,080	0.829	0.790	27.67%	0.839	0.806	26.02%	0.786	0.756	26.67%	**0.863**	**0.829**	**23.65%**
`video_filter`	3,741	0.762	0.723	32.02%	0.805	0.754	30.86%	0.743	0.724	29.90%	**0.818**	**0.785**	**27.89%**
`pfpu_fmul`	4,281	0.753	0.718	34.38%	0.798	0.738	33.86%	0.734	0.714	31.74%	**0.808**	**0.772**	**28.46%**
Average	–	0.786	0.752	30.85%	0.815	0.775	29.72%	0.757	0.729	29.09%	**0.831**	**0.798**	**26.27%**

b Vector-Based Performance Across Multiple Circuits

Input Vector	#Ports	MasterRTL			GRASPE			VIRTUAL w/o Contrastive			VIRTUAL		
		PCC	R^2	MAPE	PCC	R^2	MAPE	PCC	R^2	MAPE	PCC	R^2	MAPE
A	68	0.801	0.688	37.94%	0.761	0.504	30.45%	0.675	0.631	22.39%	**0.806**	**0.701**	**22.19%**
B	74	0.726	0.620	**24.70%**	0.609	0.728	37.47%	0.658	0.799	31.40%	**0.813**	**0.898**	26.49%
C	196	0.769	0.730	27.37%	**0.861**	0.715	**21.22%**	0.725	0.746	28.20%	0.812	**0.831**	21.34%
D	1,059	0.820	0.703	32.48%	0.892	0.634	32.39%	**0.896**	0.797	24.68%	0.885	**0.920**	**23.84%**
E	1,317	0.671	0.721	31.44%	0.876	0.652	30.53%	0.772	**0.853**	25.75%	**0.893**	0.830	**23.31%**
Average	–	0.757	0.692	30.79%	0.780	0.647	30.41%	0.745	0.765	26.48%	**0.842**	**0.836**	**23.43%**

Subtable (a) presents the *circuit-based* perspective, where each circuit is evaluated over multiple ($\geq$ 100) input vectors. **Subtable (b)** focuses on the *vector-based* perspective, where exercising each input vector across multiple circuits. **Baselines:** MasterRTL, GRASPE, and VIRTUAL w/o Contrastive serve as baselines or ablation methods. VIRTUAL (Ours) includes the full self-supervised contrastive pretraining.

TABLE IV End-to-End Runtime Comparison among Conventional Flow, VIRTUAL, and Other Methods on 100 Vectors

Circuit	#Nodes	Conventional Flow	MasterRTL		GRASPE		VIRTUAL	
		Total (s)	Total (s)	Speedup (×)	Total (s)	Speedup (×)	Total (s)	Speedup (×)
`fir_filter`	3,940	568.73	61.23	9.29×	67.62	8.41×	**36.95**	**15.39**×
`mips_greg`	7,085	782.47	86.98	8.99×	98.76	7.92×	**65.29**	**11.98**×
`dual_port_ram`	10,735	992.38	113.84	8.72×	128.68	7.71×	**71.90**	**13.80**×
`flt_recip_iter`	15,863	1231.69	147.42	8.35×	168.24	7.32×	**84.53**	**14.57**×
`riscv_multiplier`	17,873	1526.42	182.96	8.34×	207.34	7.36×	**109.16**	**13.98**×
`diffeq_f_systemC`	20,121	1907.35	216.28	8.81×	243.17	7.84×	**119.81**	**15.92**×
Average	–	–	–	8.75×	–	7.76×	–	**14.27**×

Total (s) measured end-to-end time in seconds for each pipeline on 100 vectors for each large circuit. **Speedup** is relative to Conventional Flow. The **Conventional Flow** includes synthesis + gate-level simulation + power analysis, showing circuit-specific scaling factors. **MasterRTL** and **GRASPE** exhibit reduced efficiency for control-intensive designs due to simulation dependencies and complex state transitions. **VIRTUAL** benefits from complete simulation avoidance, though graph representation overhead increases with circuit complexity, resulting in lower but still significant speedups for larger designs.

ficial effect of self-supervised contrastive pretraining. In Fig. 4 (a), the IVE contrastive learning loss rapidly decreases within the first few epochs, reflecting how the Transformer-based encoder efficiently captures toggling patterns from large input-vector sets. In Fig. 4 (b), the RGE Contrastive Loss converges more gradually, consistent with the complex topological signals embedded in RTL graphs. However, it also reaches a stable minimum, suggesting that the asynchronous GNN effectively extracts the functional and structural insights critical to power prediction. Finally, the DPP supervised learning loss in Fig. 4 (c) exhibits a smooth and pronounced descent. The curve demonstrates how the pre-aligned embeddings for both input vectors and RTL graphs streamline subsequent fine-tuning. Taken together, these learning curves confirm that the collaboration between a specialized Input Vector Encoder (IVE) and an RTL Graph Encoder (RGE), reinforced by contrastive pretraining, contributes to robust accuracy and rapid convergence.

VI. Conclusions

VIRTUAL implements an efficient framework for vector-based dynamic power estimation directly from RTL, eliminating the need for time-consuming gate-level synthesis and simulation. By decoupling input vector processing via transformer-based encoders and RTL structural analysis through graph attention networks, our approach enables accurate power prediction with minimal labeled data. The self-supervised contrastive learning strategy significantly enhances feature representations in both sequential and graph domains, delivering robust power estimations across diverse circuit designs. Experimental results validate VIRTUAL's effectiveness, achieving 0.842 PCC, 0.836 R^2, and 23.43% MAPE while providing more than 14× speedup over traditional power analysis workflows. The acceleration enables rapid design iteration and comprehensive design-space exploration previously constrained by computational bottlenecks. Moving forward, we will explore more sophisticated input vector characterization methods and RTL functional patterns to enhance early-stage power estimation in industrial design flows.

Acknowledgements

The project is supported in part by Research Grants Council of Hong Kong SAR (No. RFS2425-4S02 and No. CUHK14210723), and AI Chip Center for Emerging Smart Systems (ACCESS), Hong Kong.

References

[1] H. Shin, Y. Ban, J. Kil, H. Hwang, K. Cho, S. Sim, B. Lee, I. Shin, H. Kim, and C. Jang, "Machine Learning Applications on 3nm Node Technology and Designs for Improving Block-Level PPA," in *DTCO and Computational Patterning II*, vol. 12495. SPIE, 2023, pp. 276–284.

[2] M. Huang, J. Renau, S.-M. Yoo, and J. Torrellas, "A Framework for Dynamic Energy Efficiency and Temperature Management," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2000, pp. 202–213.

[3] N. P. Dharuman and S. MAHIMKAR, "SystemC in Semiconductor Modeling: Advancing SoC Designs," *SSRN*, 2024.

[4] Y. A. Durrani and T. Riesgo, "Efficient Power Analysis Approach and Its Application to System-on-Chip Design," *Microprocessors and Microsystems*, vol. 46, pp. 11–20, 2016.

[5] P. Kurup and T. Abbasi, *Logic Synthesis using Synopsys®*. Springer Science & Business Media, 1997.

[6] Synopsys, Inc., "VCS: Functional Verification Solution," https://www.synopsys.com/verification/simulation/vcs.html, 2025.

[7] V. B. Rao, D. V. Overhauser, T. N. Trick, and I. N. Hajj, *Switch-Level Timing Simulation of MOS VLSI Circuits*. Springer Science & Business Media, 2012, vol. 66.

[8] Synopsys , Inc., "PrimePower: RTL to Signoff Power Analysis," https://www.synopsys.com/implementation-and-signoff/signoff/primepower.html, 2025.

[9] Y. Zhang, "Deep Learning for Power and Switching Activity Estimation," in *Machine Learning Applications in Electronic Design Automation*. Springer, 2022, pp. 85–114.

[10] A. Raghunathan, N. K. Jha, and S. Dey, *High-level power analysis and optimization*. Springer Science & Business Media, 2012.

[11] B. Fischer, C. Cech, and H. Muhr, "Power Modeling and Analysis in Early Design Phases," in *IEEE/ACM Proceedings Design, Automation and Test in Eurpoe (DATE)*, 2014, pp. 1–6.

[12] S. R. Nesset, "RTL Power Estimation Flow and Its Use in Power Optimization," Master's thesis, NTNU, 2018.

[13] Y. Zhang, H. Ren, and B. Khailany, "GRANNITE: Graph Neural Network Inference for Transferable Power Estimation," in *ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[14] W. Fang, Y. Lu, S. Liu, Q. Zhang, C. Xu, L. W. Wills, H. Zhang, and Z. Xie, "MasterRTL: A Pre-Synthesis PPA Estimation Framework for Any RTL Design," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2023, pp. 1–9.

[15] M. Rakesh, P. Das, A. Terkar, and A. Acharyya, "GRASPE: Graph Representation Learning-based Accurate Post-Synthesis Power Estimation from RTL," in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2023, pp. 1–5.

[16] A. S. Priya, S. Kamatchi, and E. L. Prasad, "Early Register Transfer Level (RTL) Power Estimation in Real-Time System-on-Chips (SoCs)," *Journal of Integrated Science and Technology*, vol. 11, no. 1, pp. 454–454, 2023.

[17] M. Rakesh, P. Das, S. P. KR, and A. Acharyya, "GRILAPE: Graph Representation Inductive Learning-based Average Power Estimation for Frontend ASIC RTL Designs," in *International Conference on VLSI Design*. IEEE, 2023, pp. 1–6.

[18] R. Verma, *Switching Rate Activity Monitoring at RTL and Synthesis Level for Circuit Integrity*. The Ohio State University, 2023.

[19] W. Chen, S. Ray, J. Bhadra, M. Abadir, and L.-C. Wang, "Challenges and trends in modern soc design verification," *IEEE Design & Test*, vol. 34, no. 5, pp. 7–22, 2017.

[20] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, "Self-Supervised Representation Learning: Introduction, Advances, and Challenges," *IEEE Signal Processing Magazine*, vol. 39, no. 3, pp. 42–62, 2022.

[21] P. Kumar, P. Rawat, and S. Chauhan, "Contrastive Self-Supervised Learning: Review, Progress, Challenges and Future Research Directions," *International Journal of Multimedia Information Retrieval*, vol. 11, no. 4, pp. 461–488, 2022.

[22] S. Deldari, H. Xue, A. Saeed, J. He, D. V. Smith, and F. D. Salim, "Beyond Just Vision: A Review on Self-Supervised Representation Learning on Multimodal and Temporal Data," *arXiv preprint*, p. arXiv:2206.02353, 2022.

[23] S. Gayathri and T. Taranath, "RTL Synthesis of Case Study Using Design Compiler," in *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 2017, pp. 1–7.

[24] Synopsys, Inc., "Design Compiler," https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/dc-ultra.html, 2025.

[25] E. Macii, M. Pedram, and F. Somenzi, "High-Level Power Modeling, Estimation, and Optimization," in *ACM/IEEE Design Automation Conference (DAC)*, 1997, pp. 504–511.

[26] Y. Zhang, H. Ren, and B. Khailany, "Opportunities for RTL and Gate Level Simulation Using GPUs," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2020, pp. 1–5.

[27] W. Wolf, *Modern VLSI Design: System-on-Chip Design*. Pearson Education, 2002.

[28] B. H. Calhoun, Y. Cao, X. Li, K. Mai, L. T. Pileggi, R. A. Rutenbar, and K. L. Shepard, "Digital Circuit Design Challenges and Opportunities in the Era of Nanoscale CMOS," *Proceedings of the IEEE*, vol. 96, no. 2, pp. 343–365, 2008.

[29] N. Balachandran, "Low Power Memory Controller Subsystem IP Exploration Using RTL Power Flow: An End-to-End Power Analysis and Reduction Methodology," https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1463783&dswid=-8875, 2020.

[30] N. Wu, Y. Li, H. Yang, H. Chen, S. Dai, C. Hao, C. Yu, and Y. Xie, "Survey of Machine Learning for Software-Assisted Hardware Design Verification: Past, Present, and Prospect," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 29, no. 4, pp. 1–42, 2024.

[31] W. Fang, J. Wang, Y. Lu, S. Liu, Y. Wu, Y. Ma, and Z. Xie, "A Survey of Circuit Foundation Model: Foundation AI Models for VLSI Circuit Design and EDA," *arXiv preprint*, p. arXiv:2504.03711, 2025.

[32] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 8, pp. 1798–1828, 2013.

[33] Y. Wang, C. M. Albrecht, N. A. A. Braham, C. Liu, Z. Xiong, and X. X. Zhu, "Decoupling Common and Unique Representations for Multimodal Self-Supervised Learning," in *European Conference on Computer Vision (ECCV)*, 2024, pp. 286–303.

[34] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in Time Series: A survey," *arXiv preprint*, 2022.

[35] D. Sánchez, L. Servadei, G. N. Kiprit, R. Wille, and W. Ecker, "A Comprehensive Survey on Electronic Design Automation and Graph Neural Networks: Theory and Applications," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 28, no. 2, pp. 1–27, 2023.

[36] H. F. Community, "Hugging Face," https://huggingface.co.

[37] O. Cores, "Open Cores," https://opencores.org/.