A Multimodal AI Acceleration with Dynamic Pruning and Run-time Configuration

Hyun Woo Oh*, Hanning Chen*, Sanggeon Yun*, Yang Ni*, Behnam Khaleghi†, Fei Wen‡, and Mohsen Imani*
*University of California, Irvine, †Qualcomm, ‡Samsung Semiconductor
Email: {hyunwooo, m.imani}@uci.edu

***Abstract*—The computational diversity of multimodal AI workloads—spanning vision transformers (ViTs), graph neural networks (GNNs), CNNs, and transformer-based NLP—poses a fundamental challenge to embedded acceleration platforms. We propose a fully integrated FPGA-based acceleration framework that addresses this heterogeneity via compile-time and run-time configurability. Our system introduces a reconfigurable processing unit (RPU) capable of executing dense and sparse matrix operations (DDMM, SpMM, SDDMM), a scalable top-k pruning engine for ViTs, and a domain-specific compiler for hardware-software co-design. The architecture supports real-time configuration without reloading bitstreams, enabling unified deployment across tasks. Implementations on Xilinx U50 and ZCU104 demonstrate up to $22.57\times$ and $6.86\times$ latency reductions versus RTX 4090 and Jetson Orin Nano, respectively, validating the design's efficiency for real-time, resource-limited environments.**

***Index Terms*—Graph Neural Network, Vision Transformer, Hardware Pruning, Computer Architecture, AI Accelerator.**

I. Introduction and Motivation

Multimodal AI tasks, such as vision-language reasoning and graph-informed perception, increasingly combine disparate model types. ViTs require high-throughput matrix operations over many tokens; NLP tasks demand dynamic-length sequence processing; GNNs rely on sparse matrix computations. While GPUs dominate AI inference, their inflexible compute pipelines underperform for sparse and variable workloads. FPGAs offer a reconfigurable alternative but suffer from limitations due to the long reconfiguration time.

II. Proposed Architecture

We introduce an FPGA framework designed for multimodal AI acceleration. Key features include:

- **Reconfigurable Processing Unit (RPU):** A grid of task-adaptive compute arrays capable of executing DDMM, SDDMM, and SpMM kernels with runtime-switchable modes (e.g., WS, OS, SIMD, RADT).
- **Top-k Engine:** A dual-stage sorter combining bitonic and merge sort units enables efficient token pruning in ViTs, with configurable k and scalable logic footprint.
- **Nonlinear Units:** Polynomial and piecewise approximations support GELU, ELU, and SoftMax.
- **Compiler Stack:** Performs layer classification, kernel selection, RPU scheduling, and hardware configuration search, supporting both static and fuzzy (runtime-determined) layers.

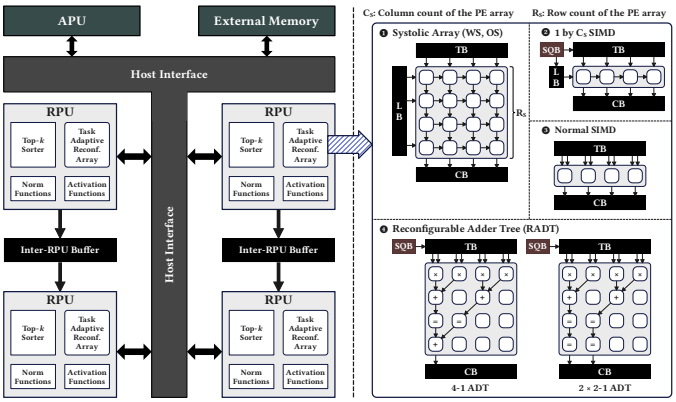

Fig. 1: The architecture of the accelerator and hardware kernels. (**a**) The top architecture. (**b**) The run-time configurable hardware kernels of the task adaptive reconfigurable array.

III. Evaluation

We evaluate the design on three representative workloads:

- **TinyCLIP (ViT+NLP):** Achieved $22.57\times$ latency speedup over RTX 4090, with $3.1\times$ gain from dynamic token pruning (DynamicViT).
- **MissionGNN (ViT+GNN):** Latency does not exceed 25ms even in bigger model configurations.
- **MDETR (CNN+NLP):** Demonstrated compatibility; pruning support for NLP remains future work.

On the ZCU104 (low-cost FPGA), latency dropped from 30ms (Jetson Orin Nano) to 4.37ms, showing $6.86\times$ improvement.

IV. Conclusion and Future Work

This work introduces the first unified FPGA-based accelerator for multimodal AI with runtime pruning and configurable compute. Our results suggest strong potential for real-time AI systems under tight resource budgets. Future work will include weight pruning in CNNs, token pruning in NLP, and fine-grained RPU scheduling.

References

[1] A. Kamath *et al.*, "MDETR: Modulated Detection for End-to-End Multi-Modal Understanding," *ICCV*, 2021.
[2] K. Wu, *et al.*, "TinyCLIP: CLIP Distillation via Affinity Mimicking and Weight Inheritance," *ICCV*, 2023.
[3] Y. Rao *et al.*, "DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification," *NeurIPS*, 2021.
[4] S. Yun *et al.*, "MissionGNN," arXiv:2406.18815, 2024.
[5] H. W. Oh, J. Park, and S. E. Lee, "DL-Sort: A Hybrid Approach to Scalable Hardware-Accelerated Fully-Streaming Sorting," *TCAS-II*, 2024.