

RIFT: A Single-Bitstream, Runtime-Adaptive FPGA-Based Accelerator for Multimodal AI

Hyunwoo Oh*, Hanning Chen*, Sanggeon Yun*, Yang Ni†, Suyeon Jang*,
Behnam Khaleghi‡, Fei Wen§, and Mohsen Imani*

*University of California, Irvine, †Purdue University Northwest, ‡Qualcomm, §Samsung Semiconductor
Email: {hyunwoo, m.imani}@uci.edu

Abstract—Multimodal models spanning ViTs, CNNs, GNNs, and NLP stress embedded systems because their heterogeneous compute and memory behaviors complicate resource allocation, load balancing, and real-time inference. We present RIFT, a *single-bitstream* FPGA accelerator and compiler for end-to-end multimodal inference. RIFT unifies layers as DDMM/SD-DMM/SpMM kernels executed on a runtime *mode-switchable engine* that morphs among weight-/output-stationary systolic, $1 \times C_S$ SIMD, and a routable adder tree (RADT) on a *shared datapath*. A two-stage hardware top- k unit, width-matched to the array, performs in-stream token pruning with minimal buffering, and *dependency-aware scheduling* overlaps independent kernels across multiple RPU—achieving adaptation without bitstream reconfiguration. On Alveo U50 and ZCU104, RIFT reduces latency by up to $22.57 \times$ versus an RTX 4090 and $6.86 \times$ versus a Jetson Orin Nano at ~ 20 – 21 W; pruning alone yields up to $7.8 \times$ on ViT-heavy workloads.

I. INTRODUCTION

Multimodal AIs combining ViTs, CNNs, GNNs, and transformer NLP enable vision–language grounding and graph reasoning [1]–[3], but their heterogeneous compute/memory behaviors hinder utilization and real-time inference.

ViTs are a major bottleneck: token-heavy attention and FFNs inflate cost [1], [4]–[7]. Token pruning reduces compute by dropping low-importance tokens [8]–[12], yet the resulting irregular sparsity often underutilizes GPUs, so realized speedups lag theory [12]–[15]. Existing accelerators typically optimize isolated modalities or kernels, leaving end-to-end multimodal graphs under-served [13], [14], [16]–[18]. FPGAs can match heterogeneity, but bitstream swapping is costly [19], [20] and duplicating dense/sparse engines inflates area and routing; even multimodal designs remain incomplete or omit *run-time* pruning [21], [22]. What we need is a single bitstream end-to-end accelerator capable of adapting to diverse workloads, while embracing dynamic token sparsity.

We observe that these layers can be expressed as three matrix kernels—dense–dense (DDMM), sampled dense–dense (SDDMM), and sparse (SpMM)—if hardware can switch dataflow at runtime. We introduce RIFT, an FPGA accelerator and compiler that map DDMM/SDDMM/SpMM onto a shared datapath with a mode-switchable engine supporting WS/OS systolic, $1 \times C_S$ SIMD, and a routable adder tree (RADT) for sparse reductions. A width-matched two-stage top- k performs in-stream token pruning so attention scores are filtered as produced [23], and dependency-aware scheduling overlaps independent kernels across reconfigurable processing units (RPUs), enabling adaptation within a single bitstream.

- RIFT integrates ViT, CNN, GNN, NLP as DDMM/SDDMM/SpMM and executes them on single bitstream.
- A shared PE array switches among WS/OS systolic, $1 \times C_S$ SIMD, and RADT, avoiding duplicate engines by adapting to given kernels and sparsity.
- In-stream top- k pruning plus dependency-aware scheduling increases utilization by shrinking downstream work and overlapping independent kernels across RPUs.

We evaluate RIFT on Xilinx Alveo U50 and ZCU104 across TinyCLIP [24], MDETR [1], and MissionGNN [3]. RIFT reduces latency by up to $22.57 \times$ versus RTX 4090 and $6.86 \times$ versus Jetson Orin Nano at ~ 20 – 21 W; pruning yields up to $7.8 \times$ speedup on ViT-heavy cases and dependency-aware scheduling improves throughput by up to 79%.

II. ARCHITECTURE OVERVIEW

RIFT uses an $r \times c$ grid of RPUs with local inter-RPU buffers and a shared-datapath mode-switchable engine (MSE) per RPU (Fig. 1). Each RPU integrates: (i) an MSE PE array, (ii) an in-stream, width-matched top- k for run-time pruning, (iii) nonlinear units (e.g., SoftMax/GELU approximations), and (iv) a lightweight feed scheduler for systolic alignment and indexed sparse reads. Kernels run for thousands of cycles; switching modes drains in-flight data and updates small control registers (mode ID, routing masks, loop bounds), so adaptation incurs negligible overhead and requires no reconfiguration.

A. Mode-Switchable Engine (MSE)

The MSE is a single PE array that time-shares dense and sparse dataflows via small muxes and per-PE op control. Each PE supports west/north inputs, a partial-sum path, a small local register file, and MAC/ADD/PASS operations. A B-stream steering mux selects WS vs. OS wavefronts for DDMM; lightweight local broadcast/tap paths enable SIMD lanes and tree reductions without long global wiring.

Modes. (1) *WS/OS systolic* for dense DDMM (conv-as-GEMM, MLP, full attention); WS favors high weight reuse, OS favors wide feature maps/many tokens. (2) $1 \times C_S$ *SIMD* for moderately sparse SDDMM/SpMM when active operands per row/col $\approx C_S$ and fairly uniform (e.g., bounded-degree GNN). (3) *RADT* forms a programmable reduction tree for highly sparse or skewed reductions. (4) *SIMD element-wise* for bias/activation/edge functions. Sharing one array avoids duplicating dense/sparse engines, reducing routing pressure and helping maintain F_{\max} under pruning.

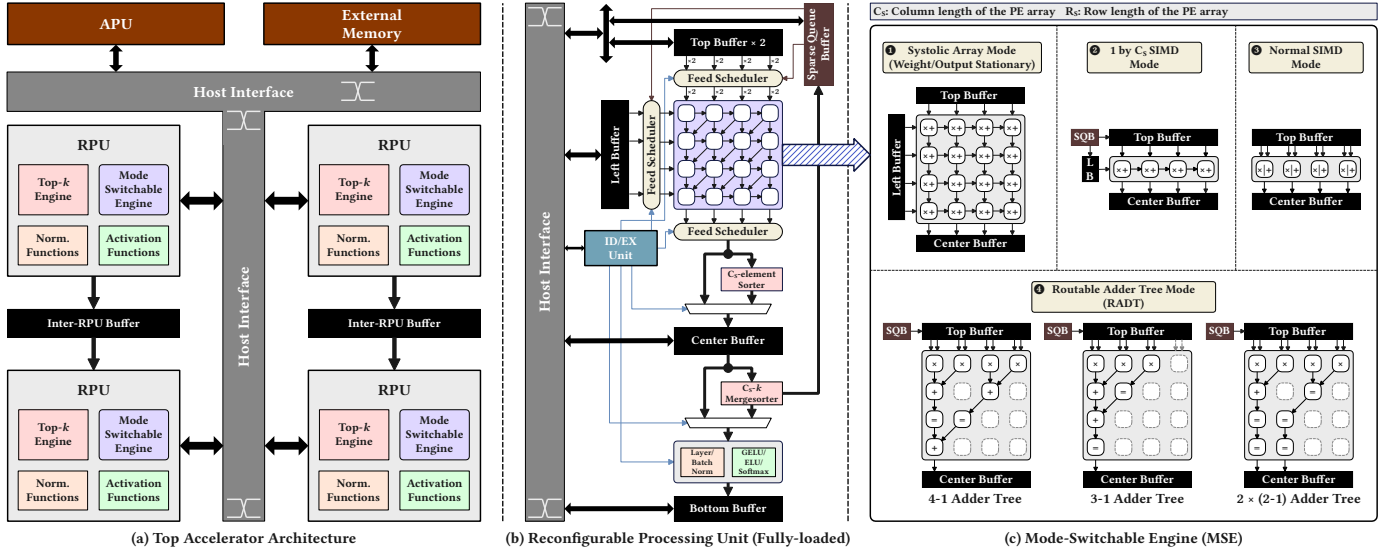


Fig. 1: **RIFT overview.** An RPU grid with local buffers; each RPU integrates a shared-datapath mode-switchable engine (MSE), in-stream width-matched top- k pruning, nonlinear units, and a feed scheduler. The PE array switches among WS/OS systolic (DDMM), $1 \times C_S$ SIMD and RADT (SDDMM/SpMM), and SIMD element-wise modes, enabling runtime adaptation without bitstream reconfiguration.

B. In-stream Top- k Pruning

RIFT prunes tokens immediately after score computation so sparsity propagates early. A two-stage top- k uses a fixed-width (up to C_S) pipelined compare network matched to the array output, followed by a small merge stage to select final k . Width-matching caps sorter cost and buffering; pruned indices stream forward to gate subsequent kernel reads and compute.

III. COMPILER AND RUNTIME

RIFT maps a model to pruning-aware execution blocks for the RPU grid. Layers are *predictable* (fixed-shape, e.g., MLP/conv) or *fuzzy* (runtime-dependent, e.g., variable-length sequences or dynamically pruned attention). The compiler emits binaries for predictable layers and templates for fuzzy layers, completed at runtime with observed fields.

Kernel unification and mode selection. Layers are lowered to DDMM/SDDMM/SpMM and assigned an MSE mode using a shape/sparsity policy: dense layers \rightarrow WS/OS; pruned attention (SDDMM) and message passing (SpMM) \rightarrow $1 \times C_S$ when activity is near C_S and uniform, else \rightarrow RADT under low/skewed activity. RADT routing is restricted to a small homogeneous candidate set for tractable compile time.

Dependency-aware scheduling. A dependency DAG exposes inter-kernel concurrency (e.g., overlapping $Q/K/V$ and independent branches) across RPUs. At runtime, a lightweight controller dispatches ready blocks, maintains backpressure via queues, and can update mode/tiling for the next block under sparsity drift—without reconfiguration.

IV. EVALUATION

We evaluate RIFT on Alveo U50 (300 MHz) and ZCU104 (200 MHz) with int8, comparing to RTX 4090 and Jetson Orin Nano (batch=1). Workloads include TinyCLIP (ViT+NLP), MDETR (CNN+NLP), and MissionGNN (ViT/CNN+GNN).

Scheduling. On multi-RPU U50, our scheduling overlaps independent work and improves throughput by up to 79.2%.

TABLE I: Headline results (batch=1, int8).

Max speedup vs. RTX 4090 (TinyCLIP-A, U50)	22.57 \times
Max speedup vs. Orin Nano (TinyCLIP-A, ZCU104)	6.86 \times
Max pruning speedup (DynamicViT)	7.8 \times
Max scheduling throughput gain (U50)	79.2%
U50 board power	\sim 20–21 W

Pruning. Using DynamicViT with $p \in \{0.1, 0.2, 0.3\}$, RIFT achieves up to 7.3 \times (U50) and 7.8 \times (ZCU104) speedup on ViT-heavy cases by streaming pruned indices into downstream SDDMM/SpMM and selecting sparse-friendly modes ($1 \times C_S$ or RADT) as sparsity/skew increases.

Cross-platform. On TinyCLIP-A, RIFT achieves up to 22.57 \times speedup vs. RTX 4090 (U50) and 6.86 \times vs. Orin Nano (ZCU104) while U50 operates at \sim 20–21 W.

V. CONCLUSION

RIFT is a single-bitstream FPGA accelerator/compiler that unifies ViT/CNN/GNN/NLP by lowering layers to DDMM/SDDMM/SpMM on a shared mode-switchable PE array. Sparse-friendly modes ($1 \times C_S$, RADT), in-stream width-matched top- k , and scheduling sustain utilization under dynamic sparsity without reconfiguration. Across TinyCLIP, MDETR, and MissionGNN, RIFT delivers up to 22.57 \times (vs. RTX 4090) and 6.86 \times (vs. Orin Nano); pruning yields up to 7.8 \times and scheduling adds up to 79.2% throughput.

ACKNOWLEDGEMENTS

This work was supported in part by the DARPA Young Faculty Award, the National Science Foundation (NSF) under Grants #2127780, #2319198, #2321840, #2312517, and #2235472, #2431561, the Semiconductor Research Corporation (SRC), the Office of Naval Research through the Young Investigator Program Award, and Grants #N00014-21-1-2225 and #N00014-22-1-2067, Army Research Office Grant #W911NF2410360. Additionally, support was provided by the Air Force Office of Scientific Research under Award #FA9550-22-1-0253, along with generous gifts from Xilinx and Cisco.

REFERENCES

- [1] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, "MDETR - Modulated Detection for End-to-End Multi-Modal Understanding," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 1760–1770.
- [2] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, "Reproducible Scaling Laws for Contrastive Language-Image Learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada, Jun. 2023, pp. 2818–2829.
- [3] S. Yun, R. Masukawa, M. Na, and M. Imani, "MissionGNN: Hierarchical Multimodal GNN-based Weakly Supervised Video Anomaly Recognition with Mission-Specific Knowledge Graph Generation," Oct. 2024, arXiv:2406.18815 [cs]. [Online]. Available: <http://arxiv.org/abs/2406.18815>
- [4] S. Chang, P. Wang, M. Lin, F. Wang, D. J. Zhang, R. Jin, and M. Z. Shou, "Making Vision Transformers Efficient from A Token Sparsification View," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, Jun. 2023, pp. 6195–6205.
- [5] Z. Chen, Y. Zhu, Z. Li, F. Yang, C. Zhao, J. Wang, and M. Tang, "The Devil is in Details: Delving Into Lite FFN Design for Vision Transformers," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seoul, Korea, Apr. 2024, pp. 4130–4134.
- [6] S. Jeong, H. E. Barkam, H. Oh, H. Chen, T. Das, Z. Ye, and M. Imani, "iTaskSense: Task-Oriented Object Detection in Resource-Constrained Environments," in *62nd ACM/IEEE Design Automation Conference (DAC)*, 2025, pp. 1–7.
- [7] H. Chen, Y. Ni, W. Huang, H. Oh, T. Das, F. Wen, and M. Imani, "Revisiting Reconfigurable Acceleration of Vision Transformer with Patch Pruning," in *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2025, pp. 1–7.
- [8] S. Tang, J. Zhang, S. Zhu, and P. Tan, "Quadtree Attention for Vision Transformers," in *International Conference on Learning Representations (ICLR)*, Virtual, Apr. 2022, pp. 1–16.
- [9] M. Fayyaz, S. A. Koohpayegani, F. R. Jafari, S. Sengupta, H. R. V. Joze, E. Sommerlade, H. Pirsiavash, and J. Gall, "Adaptive Token Sampling for Efficient Vision Transformers," in *European Conference on Computer Vision (ECCV)*, Tel Aviv, Israel, Oct. 2022, pp. 396–414.
- [10] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh, "DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification," in *35th International Conference on Neural Information Processing Systems (NeurIPS)*, Dec. 2021, pp. 13 937–13 949.
- [11] Y. Xu, Z. Zhang, M. Zhang, K. Sheng, K. Li, W. Dong, L. Zhang, C. Xu, and X. Sun, "Evo-ViT: Slow-Fast Token Evolution for Dynamic Vision Transformer," in *36th AAAI Conference on Artificial Intelligence (AAAI)*, Feb. 2022, pp. 2964–2972.
- [12] Z. Kong, P. Dong, X. Ma, X. Meng, W. Niu, M. Sun, X. Shen, G. Yuan, B. Ren, H. Tang, M. Qin, and Y. Wang, "SPViT: Enabling Faster Vision Transformers via Latency-Aware Soft Token Pruning," in *17th European Conference on Computer Vision (ECCV)*, Oct. 2022, p. 620–640.
- [13] P. Dong, M. Sun, A. Lu, Y. Xie, K. Liu, Z. Kong, X. Meng, Z. Li, X. Lin, Z. Fang, and Y. Wang, "HeatViT: Hardware-Efficient Adaptive Token Pruning for Vision Transformers," in *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, Montreal, QC, Canada, Feb. 2023, pp. 442–455.
- [14] H. You, Z. Sun, H. Shi, Z. Yu, Y. Zhao, Y. Zhang, C. Li, B. Li, and Y. Lin, "ViTCoD: Vision Transformer Acceleration via Dedicated Algorithm and Accelerator Co-Design," in *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, Montreal, QC, Canada, Feb. 2023, pp. 273–286.
- [15] H. Chen, Y. Ni, W. Huang, H. Oh, Y. Liu, T. Das, and M. Imani, "LVLM_CSP: Accelerating Large Vision Language Models via Clustering, Scattering, and Pruning for Reasoning Segmentation," in *ACM International Conference on Multimedia (MM)*, Dublin, Ireland, 2025, p. 3932–3941.
- [16] D. Parikh, S. Li, B. Zhang, R. Kannan, C. Busart, and V. Prasanna, "Accelerating ViT Inference on FPGA through Static and Dynamic Pruning," in *IEEE 32nd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Orlando, FL, USA, May 2024, pp. 78–89.
- [17] L. Lu, Y. Jin, H. Bi, Z. Luo, P. Li, T. Wang, and Y. Liang, "Sanger: A Co-Design Framework for Enabling Sparse Attention using Reconfigurable Architecture," in *54th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-54)*, Virtual, Oct. 2021, pp. 977–991.
- [18] H. Wang, Z. Zhang, and S. Han, "SpAtten: Efficient Sparse Attention Architecture with Cascade Token and Head Pruning," in *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, Seoul, Korea, Feb. 2021, pp. 97–110.
- [19] AMD, *Configuration Time*, December 2024, vivado Design Suite User Guide: Dynamic Function eXchange (UG909), Version 2024.2. [Online]. Available: <https://docs.amd.com/r/en-US/ug909-vivado-partial-reconfiguration/Configuration-Time>
- [20] H. W. Oh, S. An, W. S. Jeong, and S. E. Lee, "RF2P: A Lightweight RISC Processor Optimized for Rapid Migration from IEEE-754 to Posit," in *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2023, pp. 1–6.
- [21] B. Zhang, R. Kannan, C. Busart, and V. Prasanna, "GCV-Turbo: End-to-end Acceleration of GNN-based Computer Vision Tasks on FPGA," in *IEEE 32nd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Orlando, FL, USA, May 2024, pp. 66–77.
- [22] B. Zhang, R. Kannan, C. Busart, and V. K. Prasanna, "VisionAGILE: A Versatile Domain-Specific Accelerator for Computer Vision Tasks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 35, no. 12, pp. 2405–2422, Dec. 2024.
- [23] H. W. Oh, J. Park, and S. E. Lee, "DL-Sort: A Hybrid Approach to Scalable Hardware-Accelerated Fully-Streaming Sorting," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 71, no. 5, pp. 2549–2553, May 2024.
- [24] K. Wu, H. Peng, Z. Zhou, B. Xiao, M. Liu, L. Yuan, H. Xuan, M. Valenzuela, X. S. Chen, X. Wang, H. Chao, and H. Hu, "TinyCLIP: CLIP Distillation via Affinity Mimicking and Weight Inheritance," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 21 913–21 923.