UniFormer: Unified and Efficient Transformer for Reasoning Across General and Custom Computing

Zhuoheng Ran*
Department of Electrical Engineering
City University of Hong Kong
zhuoheran2-c@my.cityu.edu.hk

Chong Wu
Department of Electrical Engineering
City University of Hong Kong
chongwu2-c@my.cityu.edu.hk

Renjie Xu
Department of Electrical Engineering
City University of Hong Kong
harryxu950510@gmail.com

Maolin Che
School of Mathematics and Statistics
Guizhou University
mlche@gzu.edu.cn

Hong Yan
Department of Electrical Engineering
City University of Hong Kong
h.yan@cityu.edu.hk

Abstract

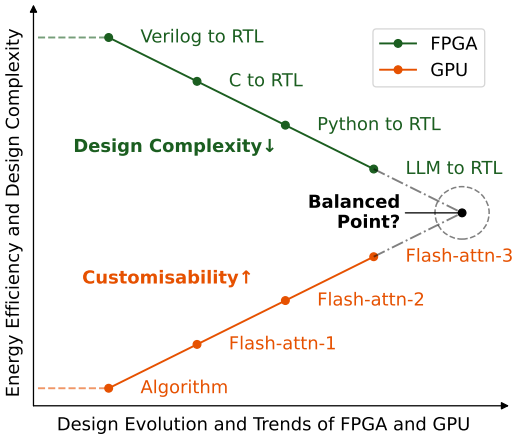
The success of neural networks such as convolutional neural networks (CNNs) has been largely attributed to their effective and widespread deployment on customised computing platforms, including field-programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs). In the current era, Transformer-based architectures underpin the majority of state-of-the-art (SOTA) larger models that are also increasingly deployed on customised computing hardware for low-power and real-time applications. However, the fundamentally different parallel computation paradigms between general-purpose and customised computing often lead to compromises in model transfer and deployability, which typically come at the cost of complexity, efficiency or accuracy. Moreover, many cross-platform optimisation principles have also remained underexplored in existing studies. This paper introduces UniFormer, a unified and efficient Transformer architecture for both general-purpose and customised computing platforms. By enabling higher parallelism and compute–storage fusion, UniFormer achieves state-of-the-art (SOTA) accuracy and latency on GPUs while exhibiting strong adaptability on FPGAs. To the best of our knowledge, this paper is the first efficient Transformer work that jointly considers both general-purpose and customised computing architectures.

1 Introduction

The increasing usage of generative AI (GenAI) has posed significant challenges to both the cost for industry and the 2030 carbon neutrality goals set by governments. Fundamentally, this stems from the attention mechanism that endows Transformers with robust modelling capabilities and underpin state-of-the-art (SOTA) performance in large language models (LLMs) [1–3], Vision Transformers (ViTs) [4–6] and other applications [7–18]. However, the attention mechanism also contributes as a major bottleneck and introduces significant computing complexity and memory overhead [19, 20].

*Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.



Methods	Target Platforms
DuSA [21]	H20 GPU
ELFATT [22]	Jetson and A100 GPU
Keyformer [23]	A100 GPU
Nyströmformer [24]	GTX 1080Ti GPU
CURSA [25]	L20 and A800 GPU
Swin Transformer [26]	A100 GPU
Longformer [27]	RTX 8000 GPU
FlatFormer [28]	Jetson and A6000 GPU
Skyformer [29]	V100 GPU
Polynormer [30]	A6000 GPU and CPU
EfficientViT [31]	Jetson GPU and CPU
UniFormer (Ours)	**GPU / FPGA**

Figure 1: **Problem setting (left):** Recent efficient Transformer works leverage customised GPU kernels to maximise parallelism and memory utilisation, while FPGA designs can already enable fine-grained dataflows and pipelined execution by high-level synthesis under significantly reduced computational complexity. Moreover, both general and customised computing architectures seek more advanced compute–storage fusion mechanisms to alleviate the common bandwidth bottleneck. **Existing works (right):** Representative efficient Transformer implementations with target platforms.

During a long period prior to recent advances, one of the key factors behind the success of many state-of-the-art (SOTA) neural networks, such as convolutional neural network (CNN), was their effective deployment on customised computing hardware that included field-programmable gate arrays (FPGAs) [32–34] and application-specific integrated circuits (ASICs) [35–37] which effectively alleviated the challenges faced by neural networks in energy efficiency and cost effectiveness. In contrast, the vanilla Transformer and existing efficient Transformers are poorly considered and adapted for customised hardware due to structural incompatibilities in distinct computational paradigms [38, 39].

Early work [40–42] on customised hardware typically sought to rearrange or replace certain functions to align with the vanilla Transformer, but this often incurred substantial design complexity. Although recent efficient Transformer variants have introduced various approximations to reduce computational complexity, many of them remain poorly supported on customised hardware [43–46] due to the issues in expensive matrix inversions, limited data reuse and excessive data dependencies. By incorporating kernel-based attention functions (e.g., ReLU [47–49]) and hardware-friendly operations such as convolutions [50–52], recent efficient Transformer works have achieved state-of-the-art (SOTA) performance on GPUs and demonstrated potential for customised hardware deployment [22, 53–56]. However, these variants were primarily designed for GPUs, and their efficiency gains often come at the cost of accuracy. This trade-off becomes particularly pronounced when low-bit quantisation is required for deployment on resource-constrained reconfigurable platforms on customised hardware.

Furthermore, recent advances in efficient Transformers and hardware accelerators also reveal converging optimisation principles across general-purpose and customised computing architectures, as illustrated in Figure 1 (left). Motivated by these observations and based on the architecture of ELFATT [22], this work makes the following contributions: a) We propose **UniFormer**, a dual-branch attention mechanism that adopts matrix multiplication as the fundamental optimisation primitive to minimise computational switching. The fusion of global linear attention and local block-wise attention reduces data dependency and enhances parallelism to enable efficient compute–storage fusion across diverse workloads. b) For GPU deployments, we develop a Triton-based GPU kernel for the global attention branch, which achieves significantly lower latency than the PyTorch implementation. Across various sequence lengths and batch configurations, the proposed kernel delivers a throughput improvement ranging from **1.79**× to **2.36**×, while maintaining comparable or higher Top-1 accuracy on ImageNet. c) For FPGA deployments, UniFormer achieves **470**× acceleration and energy efficiency gains over vanilla Transformers with cross-platform deployment capabilities.

To the best of our knowledge, this is the **first** efficient Transformer work that takes into account both general-purpose and customised computing architectures.

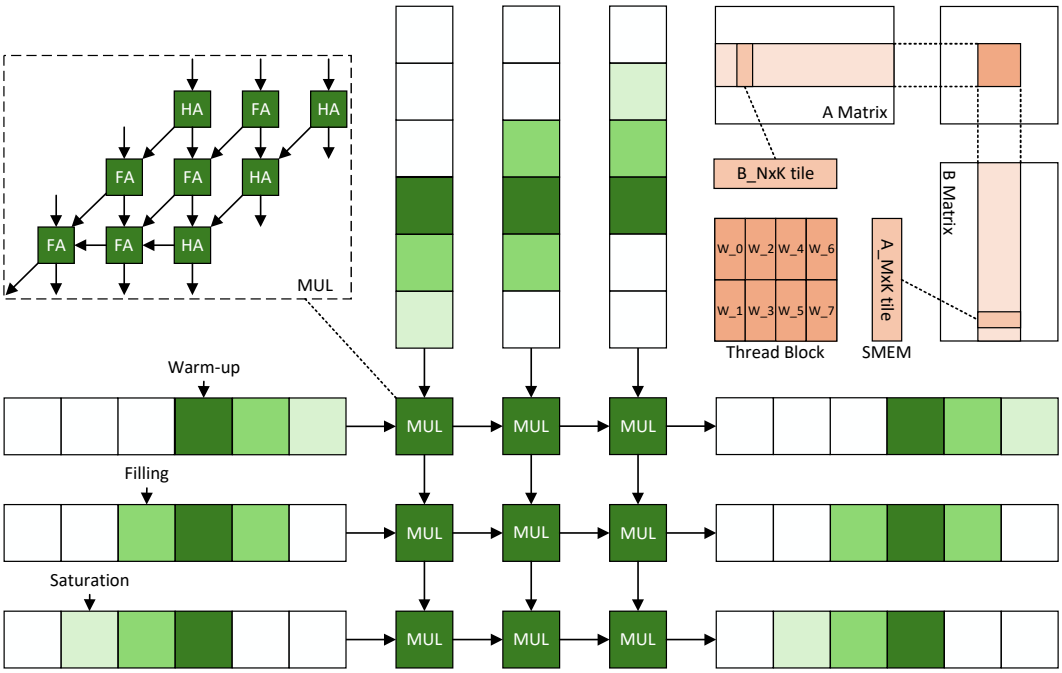


Figure 2: Matrix multiplication on parallel computing architectures, such as FPGAs and GPUs, can be further decomposed into fine-grained parallel blocks and multiply-add operations to enable customised and hardware-aware optimisations. Favourable trade-offs between design complexity and efficiency make matrix multiplication an efficient computational primitive across both architectures.

2 Preliminaries: Matrix Multiplication in Attention and Parallel Computing

Many operations introduced in recent efficient Transformer variants are not directly applicable to customised computing platforms, as they often rely on some improved functions that are poorly supported or inefficient on customised computing platform. Among the solutions and operations that remain feasible, hybrid structures such as ReLU-based attention mechanisms and convolutional operations show promising potential as hardware-friendly alternatives. However, these designs still incur non-trivial overhead due to frequent switching between varying types of operations and may suffer from degradation in both accuracy and generalisation as a result of deviations from the original GEMM-based similarity computation. These limitations are further exacerbated on customised hardware, where fixed-point arithmetic and quantisation are indispensable for efficient deployment.

In contrast, GEMM can serve as the "highest common factor" for optimisation in general-purpose computing architectures and customised computing architectures for the following reasons: First, since hybrid designs with diverse operators often lead to computational imbalance and irregular dataflows, incurring additional scheduling and communication overhead, GEMM preserves the mathematical fidelity of attention and minimises accuracy loss compared to replacing it with alternative operators such as convolutions or other operations. Second, many challenges common to GPUs and FPGAs, such as limited off-chip bandwidth and the necessity of compute–storage fusion, can be further addressed through well-optimised GEMM computing components. Moreover, GEMM has been extensively studied and highly optimised for both GPUs and FPGAs, making it a natural optimisation target. On GPUs, tensor cores enable cycle-level fused multiply–accumulate operations with warp-level collaboration, shared memory tiling and WMMA APIs to support massively parallel GEMM execution. Whereas in FPGAs, GEMM can be mapped to systolic arrays that are deeply pipelined to maximise throughput and energy efficiency, with each multiplication unit further decomposed into combinational adders to exploit fine-grained parallelism. In addition, GEMM has been a central focus in approximate computing and low-precision arithmetic research, where units with reduced bitwidth multiply-accumulate maintain efficiency without compromising accuracy. As illustrated in Figure 2, these facts and paradigms suggest that GEMM can serve as a minimal optimised primitive and is well-suited for efficient Transformers deployed across GPUs and FPGAs.

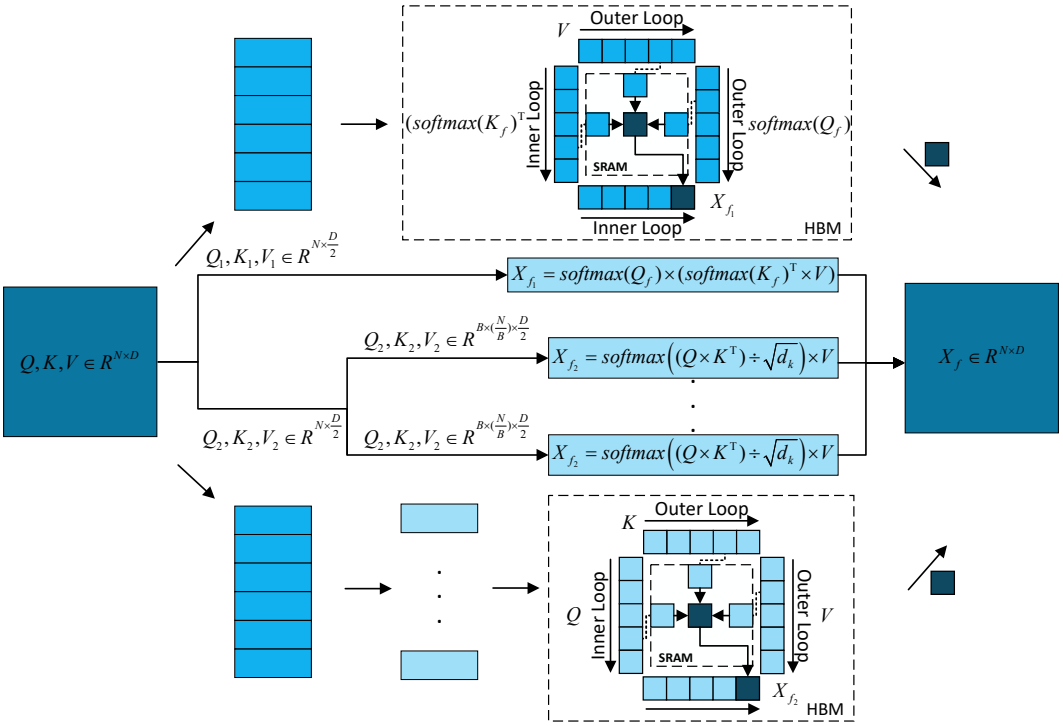


Figure 3: The proposed dual-branch Transformers. The input features are split into more parallel streams and can be fused with Triton kernels with inner-loop and outer-loop scheduling strategies to maximise data and memory usage (e.g., SRAM and HBM in GPUs, BRAM and DRAM in FPGAs).

3 Methodology

We discuss in Preliminaries with Figure 2 the role of general matrix multiplication (GEMM) as the fundamental optimisation primitive across general and customised computing architectures. To achieve higher parallelism and compute-storage fusion, we propose a dual-branch attention mechanism that integrates local structure modelling with global dependency modelling based on matrix multiplication, which can be divided into local and global branches and is shown in Figure 3. This solution can effectively mitigate full-size matrix multiplications and softmax operations in vanilla Transformers and significantly enhance computational fusion and parallelism. Specifically, the input sequence is bifurcated into two branches: the global branch captures long-range token dependencies using a linear complexity attention mechanism, while the local branch focuses on fine-grained context modelling through a block-wise strategy that partitions the sequence into parallel and independent blocks. This hierarchical decomposition facilitates extensive parallel computation and preserves global dependencies without sacrificing local precision. For improved fusion of compute and storage, we implement attention computations within the global branch using a Triton-based fused kernel inspired by FlashAttention, maximising on-chip data reuse and minimising memory transfer overhead on both GPU and FPGA targets. The overall architecture of UniFormer is illustrated in Figure 3.

Given $Q, K, V \in \mathbb{R}^{B\times N\times d_k}$, we partition the sequence into T windows of length $N_w = s^2$:

$$(Q_{b\_all}, K_{b\_all}, V_{b\_all}) = \text{blockify}(Q, K, V) \in \mathbb{R}^{(B\cdot T)\times N_w\times d_k}. \tag{1}$$

Local branch. Within each window i, we apply scaled dot-product attention:

$$X_b^{(i)} = \text{softmax}\left(\frac{Q_b^{(i)}(K_b^{(i)})^\top}{\sqrt{d_k}}\right) V_b^{(i)}. \tag{2}$$

The window outputs are merged to the full sequence by deblocking.

Algorithm 1 Algorithm 1: Block-Local Attention

Input: $Q, K, V \in \mathbb{R}^{B\times N\times d_k}$; block size BLOCK_SIZE (and tiling BLOCK_N)
Output: Local-branch output $X_1 \in \mathbb{R}^{B\times N\times d_k}$

$(Q_{\text{loc}}, K_{\text{loc}}, V_{\text{loc}}) \leftarrow \text{SelectPortionForLocal}(Q, K, V)$
$(Q_b, K_b, V_b) \leftarrow \text{blockify}(Q_{\text{loc}}, K_{\text{loc}}, V_{\text{loc}})$
Initialize $X_{1,b}$ with the shape of Q_b; $s \leftarrow \sqrt{d_k}$
for each block index i **in parallel do**
  $Q_i \leftarrow Q_b[i,:,:],\ K_i \leftarrow K_b[i,:,:],\ V_i \leftarrow V_b[i,:,:]$
  for each row q_r of Q_i **do**
    $m \leftarrow -\infty;\ l \leftarrow 0;\ \mathbf{acc} \leftarrow \mathbf{0} \in \mathbb{R}^{d_k}$
    for each sub-block $(K_{i,j}, V_{i,j})$ tiled by BLOCK_N **do**
      $S \leftarrow q_r K_{i,j}^\top / s;\ m_{\text{old}} \leftarrow m;\ m \leftarrow \max(m, \max(S))$
      $P \leftarrow \exp(S - m)$
      $\mathbf{acc} \leftarrow \mathbf{acc} \cdot \exp(m_{\text{old}} - m) + PV_{i,j}$
      $l \leftarrow l \cdot \exp(m_{\text{old}} - m) + \sum P$
    end for
    $X_{1,b}[i,r,:] \leftarrow \mathbf{acc}/l$
  end for
end for
$X_1 \leftarrow \text{deblockify}(X_{1,b})$

Global branch. In parallel, the global branch as linear attention [57, 25] on $Q_g, K_g, V_g \in \mathbb{R}^{N_g\times d_k}$:

$$X_g = \text{softmax}_{\textsf{feat}}(Q_g)\left(\text{softmax}_{\textsf{seq}}(K_g)^\top V_g\right) = \text{softmax}_{\textsf{feat}}(Q_g)\, C_g, \tag{3}$$

where $\text{softmax}_{\textsf{feat}}$ normalises along the characteristic dimension (d_k), $\text{softmax}_{\textsf{seq}}$ normalises along the sequence dimension (N_g) and $C_g = \text{softmax}_{\textsf{seq}}(K_g)^\top V_g \in \mathbb{R}^{d_k\times d_k}$ is a global content matrix.

On GPUs, each of the $B \cdot T$ blocks can be mapped to independent thread blocks, and fused kernels can be designed to jointly execute local and global branches for higher throughput. In FPGAs, block computations are mapped to dedicated, deeply pipelined processing elements, enabling massive spatial parallelism. The reduced block size N_w allows data to fit into fast on-chip memory (SRAM on GPUs, BRAM on FPGAs), alleviating HBM/DRAM bottlenecks on GPU/FPGA to reduce latency.

Algorithm 2 Algorithm 2: Global Linear Attention

Input: $Q, K, V \in \mathbb{R}^{B\times N\times d_k}$
Output: Global-branch output $X_2 \in \mathbb{R}^{B\times N\times d_k}$

$(Q_g, K_g, V_g) \leftarrow \text{SelectPortionForGlobal}(Q, K, V)$
$Q'_g \leftarrow \text{softmax}_{\text{feat}}(Q_g)$
Initialize $C_g \leftarrow \mathbf{0} \in \mathbb{R}^{B\times d_k\times d_k}$
for each batch b **in parallel do**
  for each feature $f = 1 \ldots d_k$ **in parallel do**
    $m \leftarrow -\infty;\ L' \leftarrow 0;\ \mathbf{cv} \leftarrow \mathbf{0} \in \mathbb{R}^{d_k}$
    for each sequence block j of $(K_g[b], V_g[b])$ **do**
      $K_{B,f} \leftarrow f$-th column of block j in $K_g[b];\ V_B \leftarrow$ block j in $V_g[b]$
      $m_{\text{old}} \leftarrow m;\ m \leftarrow \max(m, \max(K_{B,f}))$
      $P \leftarrow \exp(K_{B,f} - m)$
      $\mathbf{cv} \leftarrow \mathbf{cv} \cdot \exp(m_{\text{old}} - m) + P^\top V_B$
      $L' \leftarrow L' \cdot \exp(m_{\text{old}} - m) + \sum P$
    end for
    $C_g[b,f,:] \leftarrow \mathbf{cv}/L'$
  end for
end for
$X_2 \leftarrow Q'_g\, C_g$

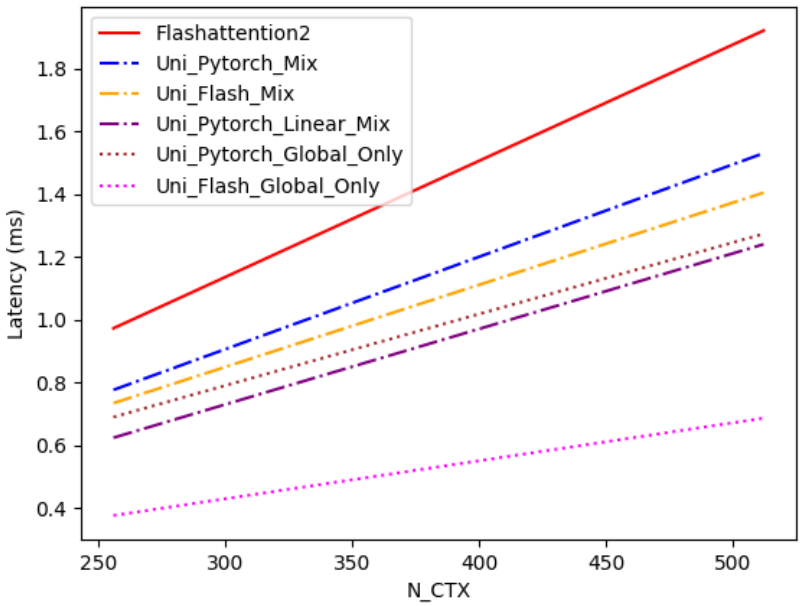


Figure 4: Throughput Analysis. Throughput performance on an H800 GPU (batch size = 64, head = 16, dimension = 64) under varying sequence lengths. The plot illustrates the latency and throughput characteristics across model configurations. The more results are provided in Appendices A.1 & A.2.

4 Results

Based on a dual-branch attention mechanism, this paper proposes a Triton-based accelerated kernel for the global branch and the native compatibility of the local branch with FlashAttention2, which allows for flexible implementation as the accelerated kernel-based, Pytorch-based local or global attention without accuracy compromise. Figure 4 shows the various latency comparisons of these combinations on the H800 GPU (Batch Size = 64, Head = 16, Dim = 64) by evaluating different sequence lengths. Additional results on more GPUs are also available in Appendices A.1 and A.2.

Finding 1: **Uni_Flash_Global_Only** (Using only global attention with the global attention-accelerated kernel) achieves the lowest latency, which is consistent with the expectations and demonstrates the effectiveness of our linear attention Triton kernel design. **Uni_Flash_Mix** (Using global and local attention with the global and local attention-accelerated kernel) combines FlashAttention2 for the local branch with our Triton linear attention kernel design for the global branch, which outperforms official FlashAttention2 and standard Pytorch implementation **Uni_Pytorch_Mix** (Global and local attention by Pytorch) and again validates the effectiveness of our Triton kernel design.

Finding 2: It is worth noting that **Uni_Flash_Mix**, which applies both local and global custom acceleration kernels but exhibits higher latency than **Uni_Pytorch_Linear_Mix** (global and local attention with the global accelerated kernel and local Pytorch implementations). This suggests that further customised kernel designs do not always lead to acceleration gains, which we attribute to potential kernel contention and the highly optimised PyTorch's native attention operators. Another supporting proof is that **Uni_Pytorch_Linear_Mix** achieves even lower latency than **Uni_Pytorch_Global_Only** (which applies linear attention to both branches and serves as a control group for the latency test). This further indicates that combining PyTorch's highly optimised local attention with a customised global kernel yields a more efficient hybrid design that better exploits the structural characteristics of the UniFormer architecture. This observation also further reveals the potential downsides of overusing compute-storage fusion, such as kernel contention, thread-level interference and scheduling overhead.

Finding 3: Against the strong baseline **Local-Pytorch+Global-Pytorch** that uses PyTorch implementations for both branches, **Local-PyTorch+Global-Kernel** achieves a $1.37\times$ acceleration (4280 vs. 3119 img/s) with identical accuracy and model size. These results effectively demonstrate that combining the Triton-accelerated global attention kernel with an optimised local attention path is a superior design strategy to fully exploit GPU bandwidth without sacrificing model performance.

Table 1: Accuracy and throughput comparison. Accuracy and throughput results of representative state-of-the-art (SOTA) Transformer models on ImageNet (224×224 resolution, H20 GPU, batch size = 512). All models have 20M–21M parameters for fair comparison.

Work	Acc. (%)	FPS
EfficientViT [31]	82.0	1812
Agent Attention [58]	82.5	2395
Vanilla Transformer [59]	82.9	2102
Flatten Transformer [60]	82.8	1988
Local-Pytorch+Global-Pytorch	82.9	3119 (All: 14.85s)
Local-PyTorch+Global-Kernel	**82.9**	**4280 (All: 10.66s)**

Finding 4: Table 1 summarises the accuracy, throughput and model size of several state-of-the-art (SOTA) Transformer models on the ImageNet classification task, evaluated at input resolution 224 and batch size 512 on the H20 GPU. Compared with representative efficient Transformer baselines, including EfficientViT [31], Agent Attention [58], Vanilla Transformer [59] and Flatten Transformer [60], our proposed **Local-Pytorch+Linear-Kernel** configuration achieves a throughput improvement ranging from $1.79\times$ to $2.36\times$, while maintaining comparable or higher Top-1 accuracy.

Table 2: Latency comparison of vanilla attention [59] with ours.

Input Size	Vanilla		UniFormer	
	Cycles	Latency (ns)	Cycles	Latency (ns)
8	6,063	3.032×10^4	3,608	1.804×10^4
16	21,032	1.052×10^5	4,918	2.459×10^4
64	299,000	1.495×10^6	12,123	6.062×10^4
128	1,171,384	5.857×10^6	23,947	1.197×10^5
256	4,636,472	2.318×10^7	47,595	2.380×10^5
512	23,002,669	1.150×10^8	94,891	4.745×10^5
1024	89,259,053	4.463×10^8	189,483	9.474×10^5

Finding 5: As shown in Table 2 and Figure 7 in Appendix A.3, the FPGA implementation reveals that the latency of vanilla attention grows quadratically with input size due to its complexity $O(N^2)$. In contrast, UniFormer maintains a nearly linear growth pattern with input size and achieves a speedup ranging from $180\times$ to $470\times$ and improved energy efficiency compared to vanilla attention.

5 Discussion

This work presented UniFormer, a unified and efficient Transformer architecture that bridges the gap between general-purpose and customised computing platforms. By leveraging a dual-branch attention mechanism that integrates global linear attention with block-wise local feature extraction, UniFormer achieves high parallelism, reduced data dependency and effective compute-storage fusion without compromising accuracy. On GPUs, the proposed Triton-based fused kernels deliver substantial latency reduction and throughput improvement, while on FPGAs, UniFormer demonstrates notable acceleration and gains in energy efficiency under the Vitis HLS implementation. These results collectively validate the scalability, portability and cross-platform adaptability of the proposed architecture across distinct computing architecture. To the best of our knowledge, this is the first efficient Transformer work that takes into account both general-purpose and customised architectures.

Acknowledgments and Disclosure of Funding

This work is supported by Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA), Hong Kong Research Grants Council (Project 11204821), and City University of Hong Kong (Projects 9610034 and 9610460).

References

[1] Zixian Huang, Wenhao Zhu, Gong Cheng, Lei Li, and Fei Yuan. Mindmerger: Efficiently boosting LLM reasoning in non-english languages. *Advances in Neural Information Processing Systems*, 37:34161–34187, 2024.

[2] Wenyu Du, Tongxu Luo, Zihan Qiu, Zeyu Huang, Yikang Shen, Reynold Cheng, Yike Guo, and Jie Fu. Stacking your transformers: A closer look at model growth for efficient LLM pre-training. *Advances in Neural Information Processing Systems*, 37:10491–10540, 2024.

[3] Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. Do LLMs dream of elephants (when told not to)? latent concept association and associative memory in transformers. *Advances in Neural Information Processing Systems*, 37:67712–67757, 2024.

[4] Anxhelo Diko, Danilo Avola, Marco Cascio, and Luigi Cinque. ReViT: Enhancing vision transformers feature diversity with attention residual connections. *Pattern Recognition*, 156: 110853, 2024.

[5] Ali Hatamizadeh and Jan Kautz. Mambavision: A hybrid mamba-transformer vision backbone. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25261–25270, 2025.

[6] Xuesong Nie, Haoyuan Jin, Yunfeng Yan, Xi Chen, Zhihang Zhu, and Donglian Qi. ScopeViT: Scale-aware vision transformer. *Pattern Recognition*, 153:110470, 2024.

[7] Roy Ganz, Yair Kittenplon, Aviad Aberdam, Elad Ben Avraham, Oren Nuriel, Shai Mazor, and Ron Litman. Question aware vision transformer for multimodal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13861–13871, 2024.

[8] Renjie Xu, Tong Wei, Yimin Wei, and Hong Yan. UTV decomposition of dual matrices and its applications. *Computational and Applied Mathematics*, 43(1):41, 2024.

[9] Chong Wu, Jiangbin Zheng, Zhenan Feng, Houwang Zhang, Le Zhang, Jiawang Cao, and Hong Yan. Fuzzy SLIC: Fuzzy simple linear iterative clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(6):2114–2124, 2020.

[10] Siddharth Srivastava and Gaurav Sharma. Omnivec2-a novel transformer based network for large scale multimodal and multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27412–27424, 2024.

[11] Zhuoheng Ran and Haosen Yu. Performance and security enhancement solutions for positron emission tomography medical hardware. In *2022 IEEE 16th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pages 1–5. IEEE, 2022.

[12] Paul Waligora, Muhammad Haseeb Aslam, Muhammad Osama Zeeshan, Soufiane Belharbi, Alessandro Lameiras Koerich, Marco Pedersoli, Simon Bacon, and Eric Granger. Joint multimodal transformer for emotion recognition in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4625–4635, 2024.

[13] Anonymous. GP-STPCA: Generalized power method for sparse tensor principal component analysis. In *Submitted to The Fourteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=mVFFqmvkDi`. under review.

[14] Renjie Xu, Yimin Wei, and Hong Yan. LU decomposition and generalized Autoone-Takagi decomposition of dual matrices and their applications. *Linear and Multilinear Algebra*, 73(14): 3179–3201, 2025.

[15] Danting Cai, Yufei Liu, Sen Li, Zhihong Xie, Jiancheng Li, Zhuoheng Ran, and Xueli Liu. A two-way wireless charging system for unmanned aerial vehicles and its path planning design, July 2022. URL https://patents.google.com/patent/CN114683882A. CN Patent 114683882A.

[16] Renjie Xu, Chong Wu, Maolin Che, Zhuoheng Ran, Yimin Wei, and Hong Yan. sparseGeoHOPCA: A geometric solution to sparse higher-order PCA without covariance estimation. *arXiv preprint arXiv:2506.08670*, 2025.

[17] Yi Lu, Jiawang Cao, Yongliang Wu, Bozheng Li, Licheng Tang, Yangguang Ji, Chong Wu, Jay Wu, and Wenbo Zhu. RSVP: Reasoning segmentation via visual prompting and multi-modal chain-of-thought. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14699–14716, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.715. URL https://aclanthology.org/2025.acl-long.715/.

[18] Zhuoheng Ran and Haosen Yu. Medical hardware for stimulation and metrology of the heart: Electrocardiogram and pacemaker. In *2023 15th International Conference on Computer Research and Development (ICCRD)*, pages 305–310. IEEE, 2023.

[19] Lorenzo Papa, Paolo Russo, Irene Amerini, and Luping Zhou. A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):7682–7700, 2024.

[20] Yubin Qin, Yang Wang, Dazheng Deng, Xiaolong Yang, Zhiren Zhao, Yang Zhou, Yuanqi Fan, Jingchuan Wei, Tianbao Chen, Leibo Liu, et al. Ayaka: A versatile transformer accelerator with low-rank estimation and heterogeneous dataflow. *IEEE Journal of Solid-State Circuits*, 59(10): 3342–3356, 2024.

[21] Chong Wu, Jiawang Cao, Renjie Xu, Zhuoheng Ran, Maolin Che, Wenbo Zhu, and Hong Yan. DuSA: Fast and accurate dual-stage sparse attention mechanism accelerating both training and inference. In *Advances in Neural Information Processing Systems*, 2025.

[22] Chong Wu, Maolin Che, Renjie Xu, Zhuoheng Ran, and Hong Yan. ELFATT: Efficient linear fast attention for vision transformers. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pages 9140–9149, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3754825. URL https://doi.org/10.1145/3746027.3754825.

[23] Muhammad Adnan, Akhil Arunkumar, Gaurav Jain, Prashant J Nair, Ilya Soloveychik, and Purushotham Kamath. Keyformer: KV cache reduction through key tokens selection for efficient generative inference. *Proceedings of Machine Learning and Systems*, 6:114–127, 2024.

[24] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A Nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14138–14148, 2021.

[25] Chong Wu, Maolin Che, and Hong Yan. The CUR decomposition of self-attention matrices in vision transformers. *TechRxiv*, 2024. doi: 10.36227/techrxiv.171392846.60982484/v4. URL http://dx.doi.org/10.36227/techrxiv.171392846.60982484/v4.

[26] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022.

[27] Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, and Bryan Catanzaro. Long-short transformer: Efficient transformers for language and vision. *Advances in Neural Information Processing Systems*, 34:17723–17736, 2021.

[28] Zhijian Liu, Xinyu Yang, Haotian Tang, Shang Yang, and Song Han. Flatformer: Flattened window attention for efficient point cloud transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1200–1211, 2023.

[29] Yifan Chen, Qi Zeng, Heng Ji, and Yun Yang. Skyformer: Remodel self-attention with gaussian kernel and nystr\" om method. *Advances in Neural Information Processing Systems*, 34: 2122–2135, 2021.

[30] Chenhui Deng, Zichao Yue, and Zhiru Zhang. Polynormer: Polynomial-expressive graph transformer in linear time. In *International Conference on Learning Representations (ICLR)*. ICLR, 2024.

[31] Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17302–17313, 2023.

[32] Roberto Bosio, Filippo Minnella, Teodoro Urso, Mario R Casu, Luciano Lavagno, Mihai T Lazarescu, and Paolo Pasini. NN2FPGA: Optimizing CNN inference on FPGAs with binary integer programming. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024.

[33] Ruiqi Chen, Haoyang Zhang, Yuhanxiao Ma, Enhao Tang, Shun Li, Yanxiang Zhu, Jun Yu, and Kun Wang. Graph-OPU: An FPGA-based overlay processor for graph neural networks. In *Proceedings of the 2023 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pages 49–49, 2023.

[34] Alexander Montgomerie-Corcoran, Petros Toupas, Zhewen Yu, and Christos-Savvas Bouganis. SATAY: A streaming architecture toolflow for accelerating YOLO models on FPGA devices. In *2023 International Conference on Field Programmable Technology (ICFPT)*, pages 179–187. IEEE, 2023.

[35] Hadi Esmaeilzadeh, Soroush Ghodrati, Andrew B Kahng, Sean Kinzer, Susmita Dey Manasi, Sachin S Sapatnekar, and Zhiang Wang. Performance analysis of CNN inference/training with convolution and non-convolution operations on ASIC accelerators. *ACM Transactions on Design Automation of Electronic Systems*, 30(1):1–34, 2024.

[36] Farzad Niknia, Ziheng Wang, Shanshan Liu, Pedro Reviriego, Ahmed Louri, and Fabrizio Lombardi. ASIC design of nanoscale artificial neural networks for inference/training by floating-point arithmetic. *IEEE Transactions on Nanotechnology*, 23:208–216, 2024.

[37] Syed Muhammad Abubakar, Yue Yin, Songyao Tan, Hanjun Jiang, and Zhihua Wang. A 746 nW ECG processor ASIC based on ternary neural network. *IEEE Transactions on Biomedical Circuits and Systems*, 16(4):703–713, 2022.

[38] Fan Yang, Xinhao Yang, Hongyi Wang, Zehao Wang, Zhenhua Zhu, Shulin Zeng, and Yu Wang. GLITCHES: GPU-FPGA LLM inference through a collaborative heterogeneous system. In *2024 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–7. IEEE, 2024.

[39] Soujanya Bhowmick. Optimizing transformer inference on FPGA: A study on hardware acceleration, 2023.

[40] Zhengang Li, Mengshu Sun, Alec Lu, Haoyu Ma, Geng Yuan, Yanyue Xie, Hao Tang, Yanyu Li, Miriam Leeser, Zhangyang Wang, et al. Auto-vit-acc: An fpga-aware automatic acceleration framework for vision transformer with mixed-scheme quantization. In *2022 32nd International Conference on Field-Programmable Logic and Applications (FPL)*, pages 109–116. IEEE, 2022.

[41] Teng Wang, Lei Gong, Chao Wang, Yang Yang, Yingxue Gao, Xuehai Zhou, and Huaping Chen. Via: A novel vision-transformer accelerator based on fpga. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(11):4088–4099, 2022.

[42] Kyle Marino, Pengmiao Zhang, and Viktor K Prasanna. Me-vit: A single-load memory-efficient FPGA accelerator for vision transformers. In *2023 IEEE 30th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, pages 213–223. IEEE, 2023.

[43] Zhuoyang Zhang, Han Cai, and Song Han. Efficientvit-sam: Accelerated segment anything model without performance loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7859–7863, 2024.

[44] Zhuoheng Ran, Muhammad AA Abdelgawad, Zekai Zhang, Ray CC Cheung, and Hong Yan. RO-SVD: A reconfigurable hardware copyright protection framework for aigc applications. In *2024 IEEE 35th International Conference on Application-Specific Systems, Architectures and Processors (ASAP)*, pages 135–142. IEEE, 2024.

[45] Renjie Xu, Shenghao Feng, Yimin Wei, and Hong Yan. CUR and generalized CUR decompositions of quaternion matrices and their applications. *Numerical Functional Analysis and Optimization*, 45(3):234–258, 2024.

[46] Zhuoheng Ran and Haizhou Sun. Hardware strategies in medical systems for visually impaired treatment. In *2022 8th International Conference on Systems and Informatics (ICSAI)*, pages 1–5. IEEE, 2022.

[47] Kai Shen, Junliang Guo, Xu Tan, Siliang Tang, Rui Wang, and Jiang Bian. A study on ReLU and softmax in transformer. *arXiv preprint arXiv:2302.06461*, 2023.

[48] Tony Zhang and Rickard Brännvall. InhibiDistilbert: Knowledge distillation for a relu and addition-based transformer. *arXiv preprint arXiv:2503.15983*, 2025.

[49] Mitchell Wortsman, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Replacing softmax with ReLU in vision transformers. *arXiv preprint arXiv:2309.08586*, 2023.

[50] Asifullah Khan, Zunaira Rauf, Anabia Sohail, Abdul Rehman Khan, Hifsa Asif, Aqsa Asif, and Umair Farooq. A survey of the vision transformers and their CNN-transformer based variants. *Artificial Intelligence Review*, 56(Suppl 3):2917–2970, 2023.

[51] Jaewon Son, Jaehun Park, and Kwangsu Kim. CSTA: CNN-based spatiotemporal attention for video summarization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18847–18856, 2024.

[52] Feiniu Yuan, Zhengxiao Zhang, and Zhijun Fang. An effective CNN and transformer complementary network for medical image segmentation. *Pattern Recognition*, 136:109228, 2023.

[53] Chong Wu, Maolin Che, Renjie Xu, Zhuoheng Ran, and Hong Yan. System and method for efficient linear fast attention for vision transformer (ELFATT), January 2025. US Patent App. 19/011,779.

[54] Chong Wu, Maolin Che, Renjie Xu, Zhuoheng Ran, and Hong Yan. System and method for efficient linear fast attention for vision transformer (ELFATT), May 2025. URL `https://patents.google.com/patent/HK30118021A2`. HK Patent 30118021A2.

[55] Haikuo Shao, Huihong Shi, Wendong Mao, and Zhongfeng Wang. An fpga-based reconfigurable accelerator for convolution-transformer hybrid efficientvit. In *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2024.

[56] Zhuoheng Ran, Zewen Ye, Chong Wu, Ray CC Cheung, and Hong Yan. FastViT: Real-time linear attention accelerator for dense predictions of vision transformer (vit). In *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2025.

[57] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3531–3539, 2021.

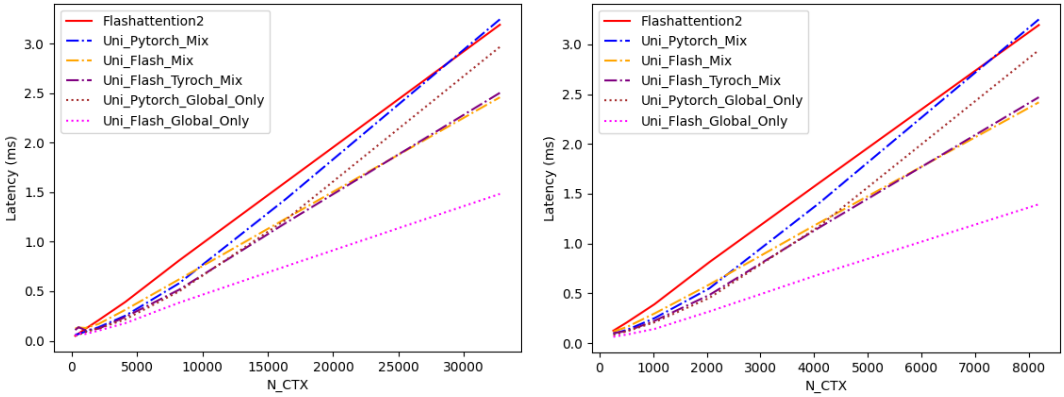
[58] Dongchen Han, Tianzhu Ye, Yizeng Han, Zhuofan Xia, Siyuan Pan, Pengfei Wan, Shiji Song, and Gao Huang. Agent attention: On the integration of softmax and linear attention. In *European Conference on Computer Vision*, pages 124–140. Springer, 2024.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[60] Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5961–5971, 2023.

A Experiments Supplementary

A.1 Results on 4070Ti Super GPU



(a) Batch Size-1, Head-16, Dimension-64

(b) Batch Size-4, Head-16, Dimension-64

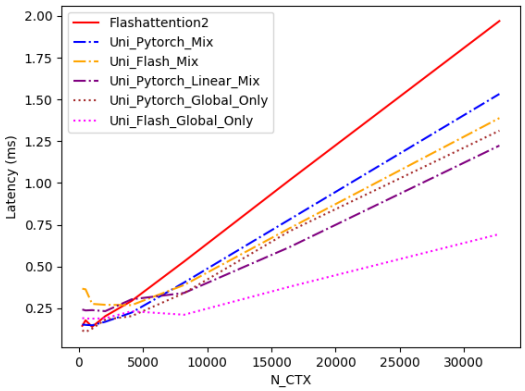
(c) Batch Size-8, Head-16, Dimension-64

(d) Batch Size-16, Head-16, Dimension-64

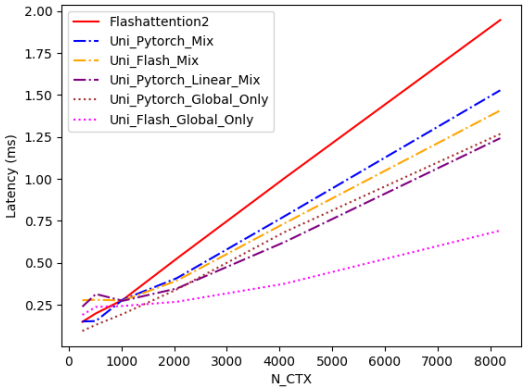
(e) Batch Size-64, Head-16, Dimension-64

Figure 5: Latency vs. input context length (N_{CTX}) under five different attention on 4070 Ti Super.

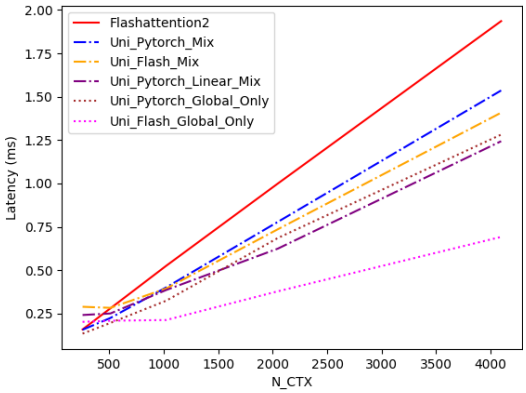
A.2 Results on H800 GPU



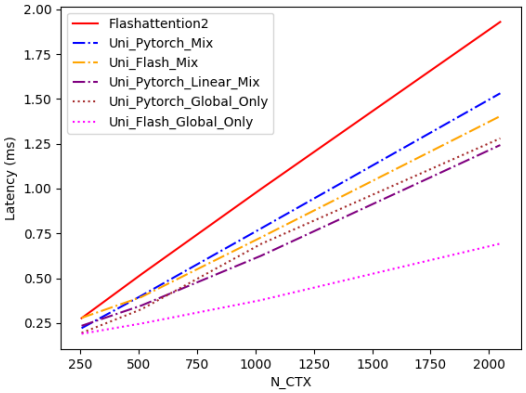
(a) Batch Size-1, Head-16, Dimension-64



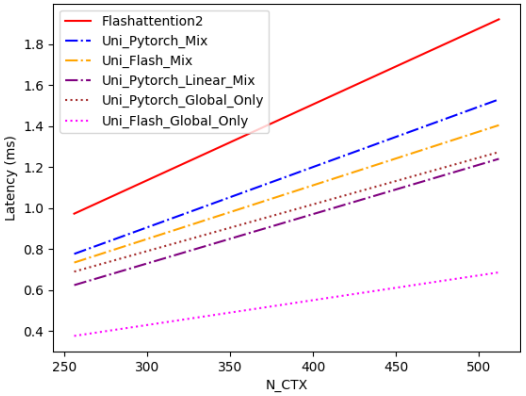
(b) Batch Size-4, Head-16, Dimension-64



(c) Batch Size-8, Head-16, Dimension-64



(d) Batch Size-16, Head-16, Dimension-64



(e) Batch Size-64, Head-16, Dimension-64

Figure 6: Latency vs. input context length (N_{CTX}) under five different attention on H800 GPU.

A.3 Additional Experimental Results in FPGA implementation

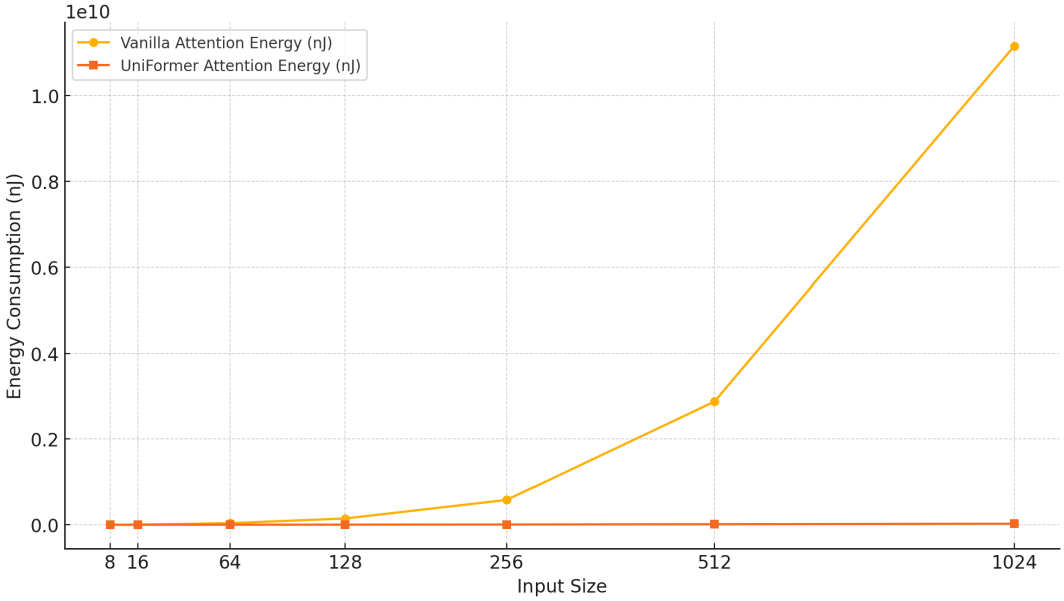


Figure 7: Energy efficiency comparison between Vanilla and UniFormer attention mechanisms.

Experiment Set Up: We implement our FPGA design using the AXU15EG development board by Xilinx Vitis HLS 2024.1, which integrates a Xilinx Zynq UltraScale+ MPSoC XCZU15EG-2FFVB1156 chip. A typical operating frequency of 250 MHz is assumed for all experiments. The estimated power consumption is based on a typical AXU15EG power profile of 25W.