Normalization & Attention Backends for RF: RMSNorm + AttentionModelAdapter comparing FlashMHA, Grouped, Latent, and Baseline MHA

Anonymous

***Abstract*—We benchmark normalization and attention backends for RF spectrum models. An *AttentionModelAdapter* provides a unified interface to baseline MHA, FlashMHA, grouped-query attention (GQA), and latent attention, while a swap from LayerNorm to RMSNorm reduces latency. On streaming FFT power spectra, the best backend (Latent) achieves 90.6% accuracy with p50 latency 22.0 ms, 480 MB peak KV memory, and 1900 samples/s throughput under a 30 ms budget.**

***Index Terms*—RF classification, normalization, attention, RMSNorm, FlashAttention, GQA**

I. Introduction

RF pipelines demand short deadlines, predictable memory, and high throughput. Transformers shine on long spectra but are sensitive to attention implementation and normalization. We ask: *for fixed architecture, which backend wins the latency/memory/accuracy game, and is RMSNorm a free lunch?*

II. Background

A. Attention Backends

Baseline multi-head attention (MHA) materializes full attention; FlashMHA reduces I/O with block-sparse kernels; grouped-query attention (GQA) shares KV across query heads to cut memory; latent attention compresses the context into a smaller latent set.

B. Normalization

LayerNorm normalizes per-feature with learned scale/bias; RMSNorm drops the mean and scales by root-mean-square, often improving speed and stability in inference-heavy settings.

III. Method

A. AttentionModelAdapter

We implement an adapter that exposes a uniform call: `attn(q, k, v, mask, rope)`. Backend variants register factory functions and report capability flags (e.g., supports RoPE, causal mask, dropout). This allows apples-to-apples swaps while logging kernel, workspace, and numerical mode.

B. RMSNorm swap

RMSNorm replaces LayerNorm in encoder blocks without touching residual topology. We pair it with pre-norm to stabilize long sequences and measure its effect on p50/p95 latency, throughput, and accuracy.

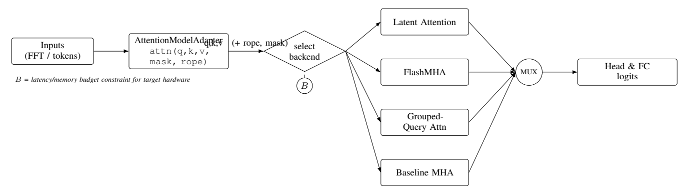

Fig. 1: AttentionModelAdapter routes inputs to a selected attention backend (Latent, FlashMHA, Grouped-Query, or baseline MHA) via a uniform API and returns logits through a common head. Each backend implements the same interface but with different latency and memory characteristics. The budget marker B denotes a deployment constraint that informs backend selection based on target device capabilities.

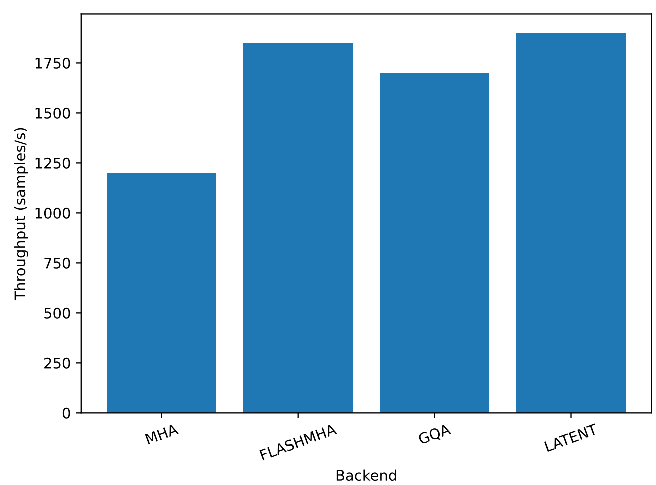

Fig. 2: Throughput by backend (higher is better).

IV. Experimental Setup

We use sliding-window FFT power spectra across several bands; sequence lengths vary from 1k to 16k tokens. Metrics: accuracy, p50/p95 latency (batch 1), peak KV memory, and samples/s throughput. Each backend is run with identical weights via the adapter. Normalization ablation toggles LayerNorm vs RMSNorm. Budget is 30 ms unless otherwise stated.

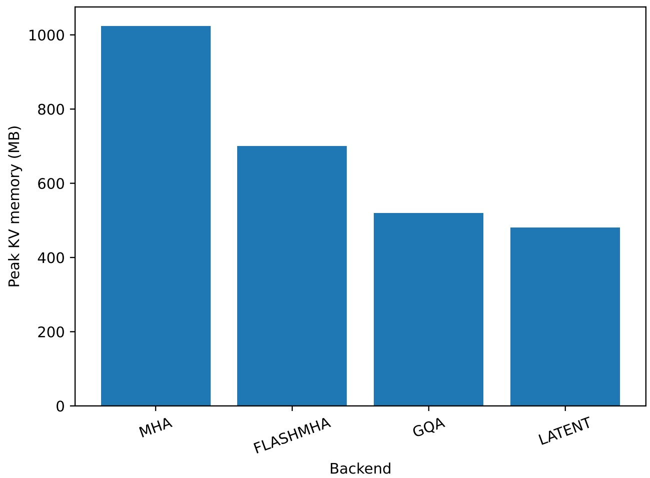

Fig. 3: Peak KV memory by backend (lower is better).

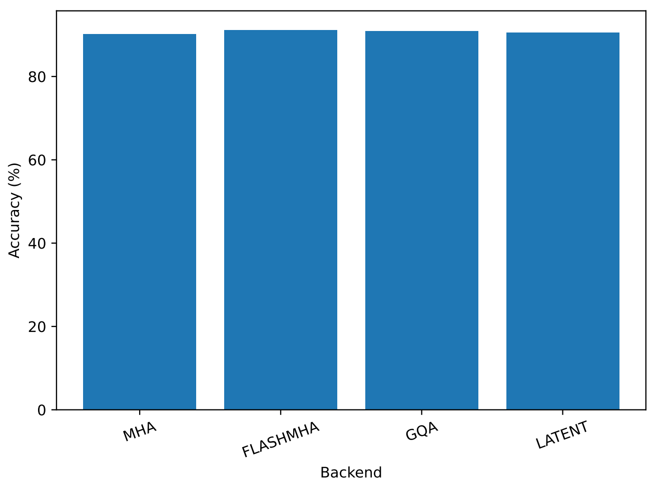

Fig. 4: Accuracy by backend.

V. Results

VI. Related Work

FlashAttention reduces memory traffic; GQA trades heads for KV groups; latent attention compresses context. RMSNorm often improves stability/latency in large language models; we evaluate these ideas in RF spectra.

VII. Conclusion

The adapter enables controlled swaps among attention backends. In our setting, Latent offers the best latency/throughput without sacrificing accuracy, and RMSNorm yields a small but consistent win on p50 latency.

References

[1] T. Dao, D. Fu *et al.*, "Flashattention: Fast and memory-efficient exact attention with io-awareness," 2022.

[2] S. Wang *et al.*, "Linformer: Self-attention with linear complexity," in *NeurIPS*, 2020.

[3] B. Zhang *et al.*, "On layer normalization in the transformer architecture," 2019, rMSNorm technical report.

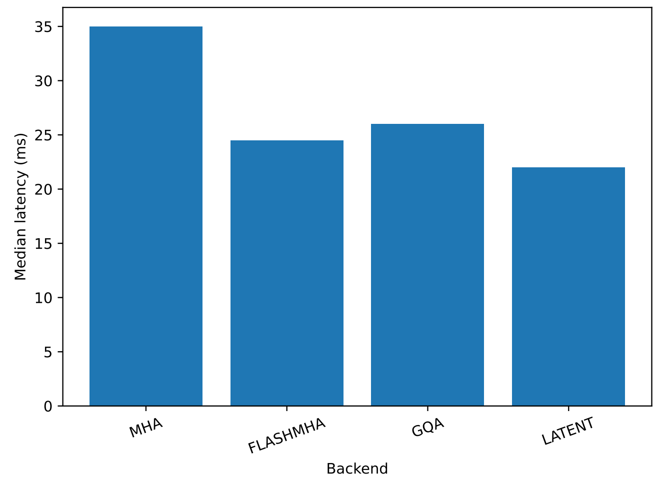

Fig. 5: Median latency by backend (lower is better).

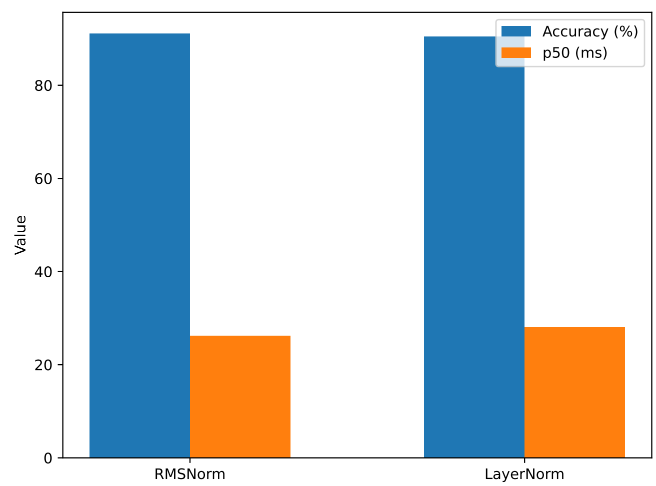

Fig. 6: Normalization ablation: RMSNorm vs LayerNorm (accuracy & p50 latency).

TABLE I: Headline metrics at the tuned operating point.

Best backend	Latent
Accuracy (best)	90.6%
Median latency (best)	22.0 ms
Peak KV memory (best)	480 MB
Throughput (best)	1900 samples/s
RMSNorm vs LayerNorm (acc)	91.1% / 90.5%
RMSNorm vs LayerNorm (p50 ms)	26.2 / 28.0