

Grouped-Query Attention (GQA)

Comprehensive Demo and Analysis

GQA reduces KV cache memory by sharing key/value heads across groups of query heads. This unifies MHA, GQA, and MQA along a single axis: the number of KV heads.

Used by: Llama 2 70B (8 KV heads), Mistral 7B (8 KV heads), Llama 3, and all modern production LLMs.

Random seed: 42
Number of visualizations: 6
Examples: 6

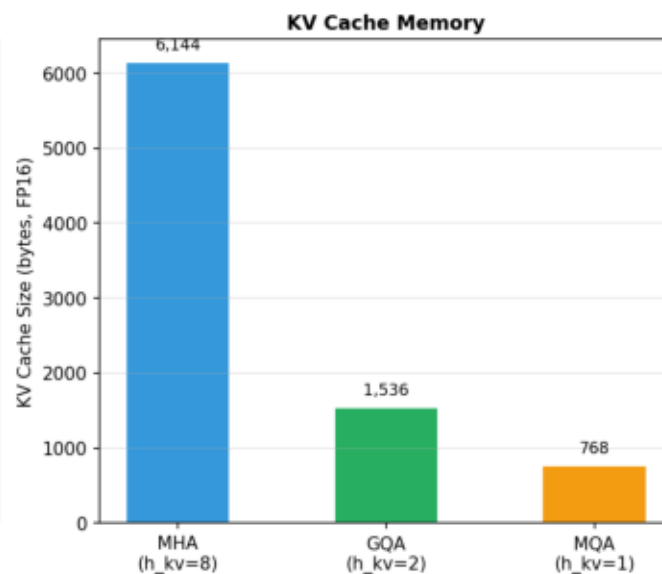
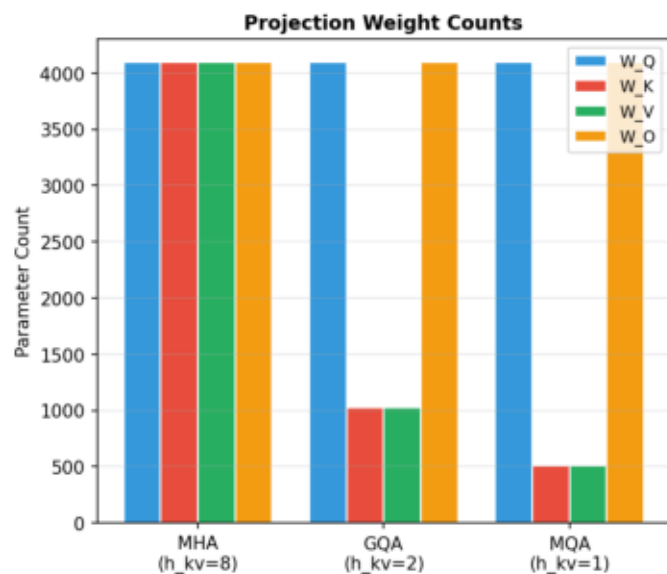
Generated by demo.py

Summary of Findings

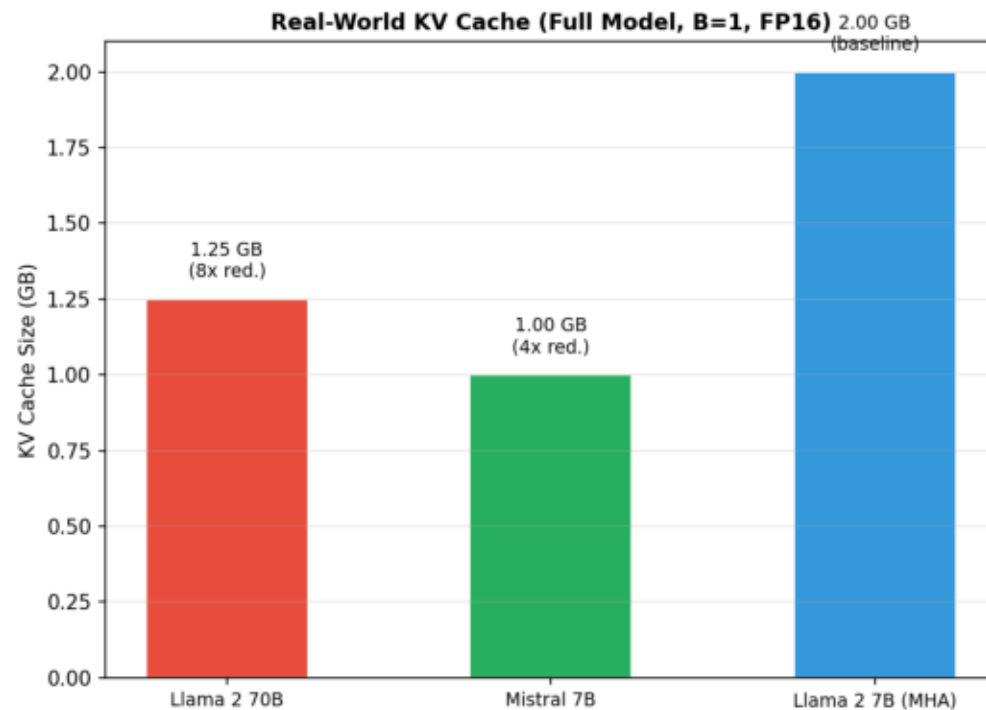
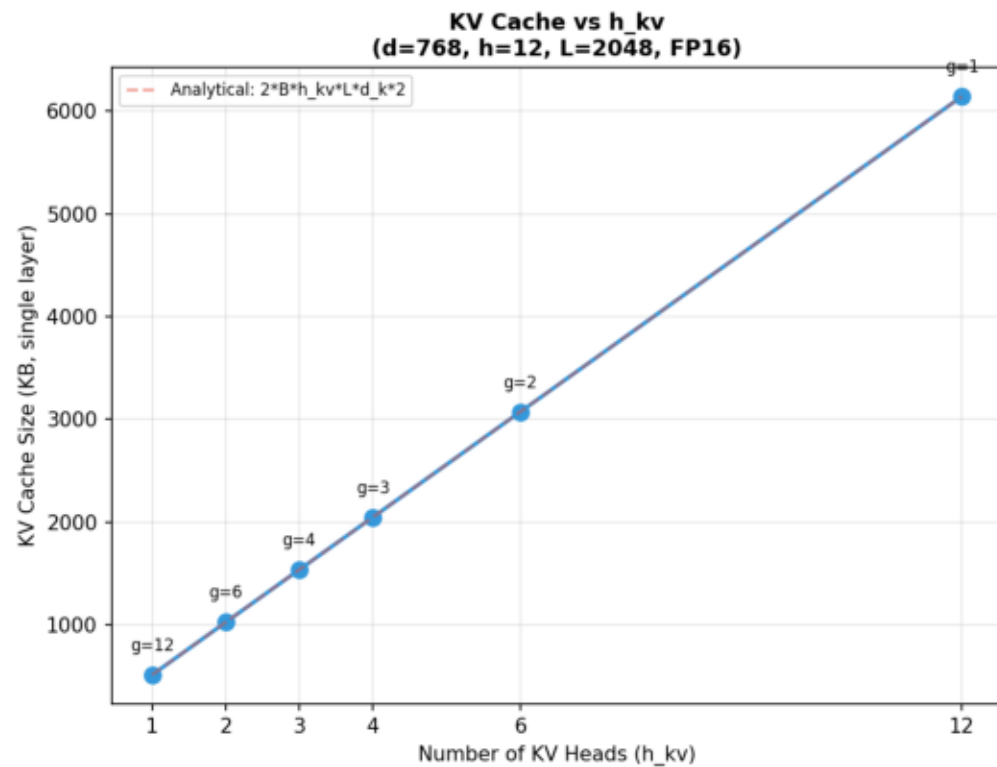
1. MHA vs GQA vs MQA: GQA with $h_{kv}=2$ saves 37.5% of KV projection parameters and reduces KV cache by 4x. MQA achieves maximum 7/8 savings but at the cost of representational diversity.
2. KV Cache Memory: Cache scales linearly with h_{kv} and L. Llama 2 70B uses 1.25 GB for KV cache (8x reduction vs hypothetical MHA at 10 GB). Mistral 7B uses 1.0 GB at 8K context. All values $B=1$, FP16.
3. Attention Patterns: Heads within a GQA group share K/V but produce different attention patterns via independent Q projections. This is the key insight -- diversity comes from Q, not KV.
CAVEAT: Shown patterns are from random init, not trained.
4. FLOPs: Attention core FLOPs (QK^T , softmax, AV) are IDENTICAL across MHA/GQA/MQA since all h query heads still compute attention. Only projection FLOPs differ. At long sequences, variants converge.
5. Gradient Flow: Per-KV-head gradients increase with group size because `reduce_kv_grad` sums g gradient contributions. Scaling is superlinear, exceeding \sqrt{g} due to correlations from shared K/V.
Backward correctness verified via numerical gradient check.
6. Scaling: KV cache is the key bottleneck for long-context inference. With $h_{kv}=8$, a 32-layer model can handle 320K+ tokens within 80GB. MHA ($h_{kv}=32$) would be limited to ~160K tokens in the same budget.

Example 1: MHA vs GQA vs MQA -- Parameters and KV Cache

MHA vs GQA vs MQA ($d_{\text{model}}=64$, $h=8$)

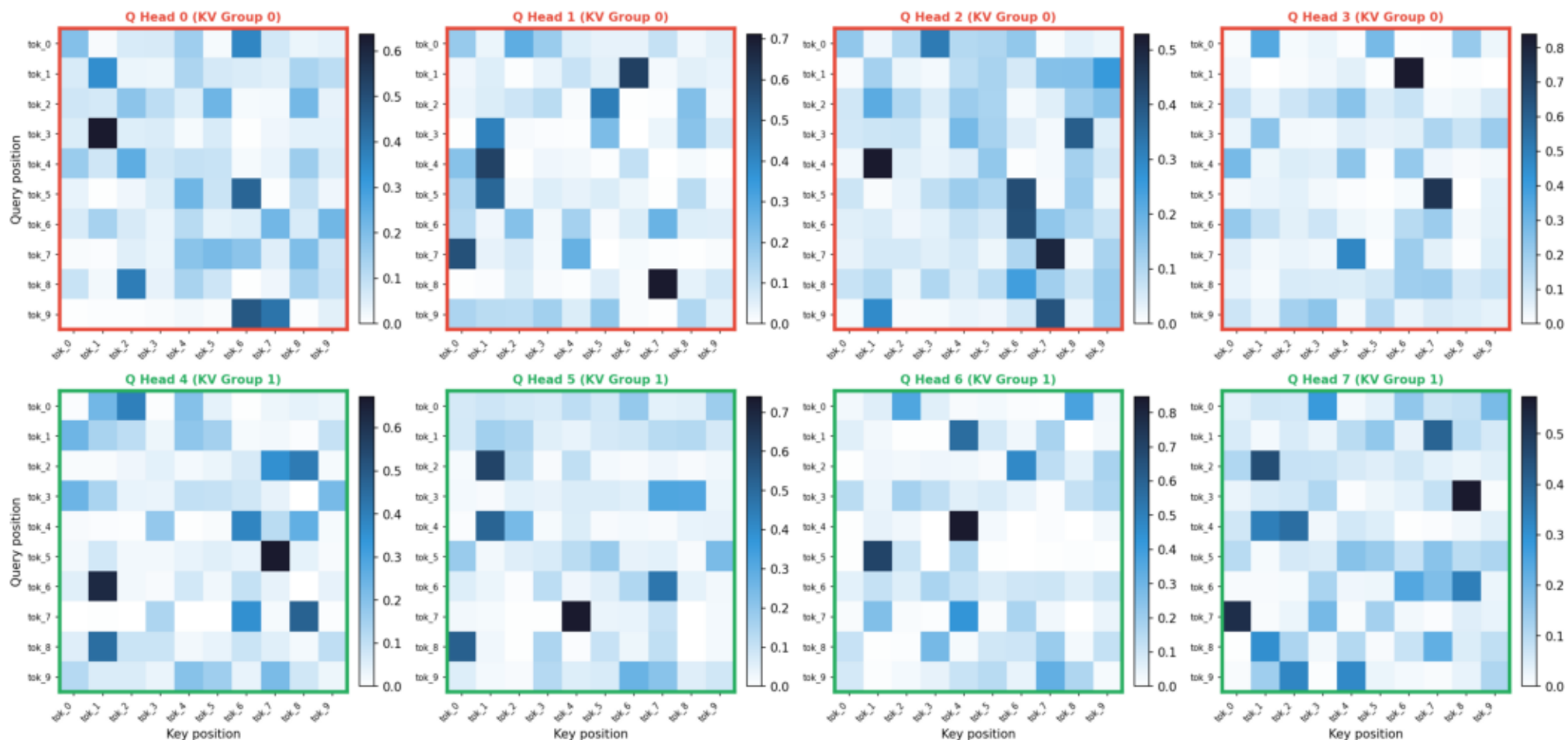


Example 2: KV Cache Memory Analysis with Real-World Configs



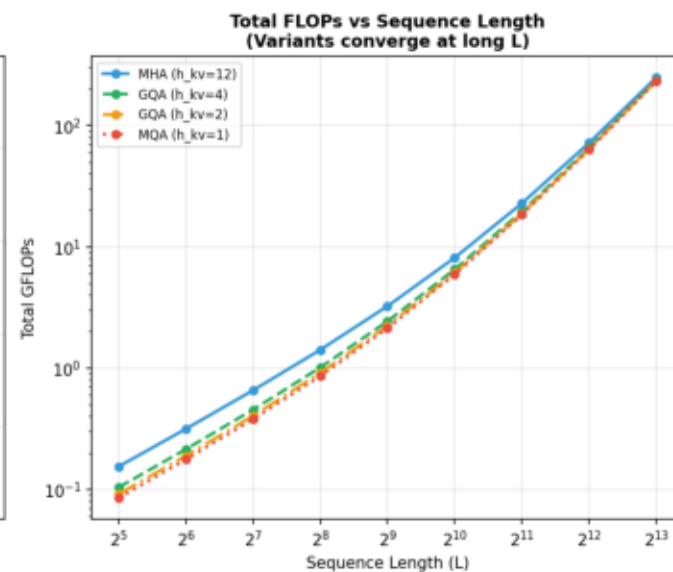
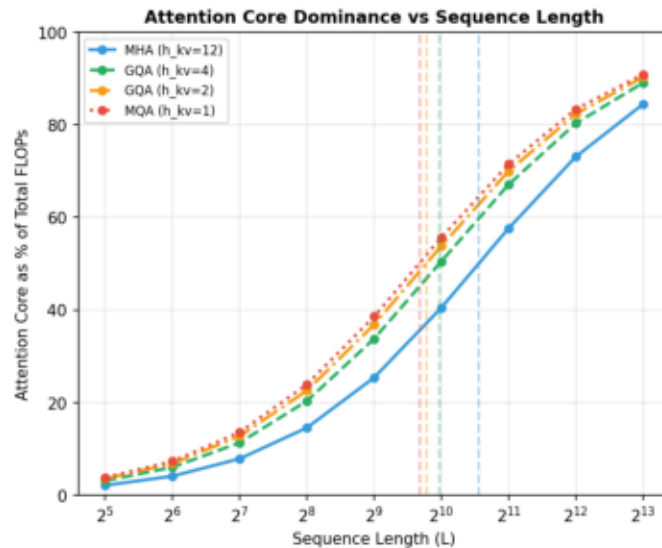
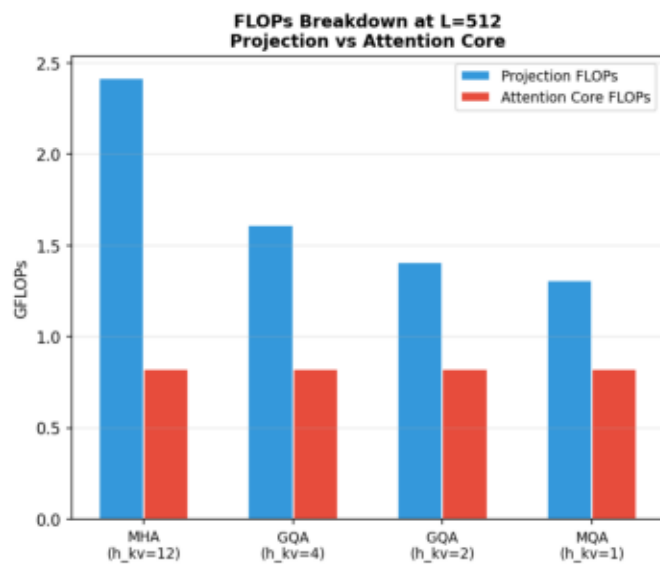
Example 3: GQA Attention Patterns (Shared KV Groups)

GQA Attention Patterns: 8 Q heads, 2 KV heads (group_size=4)
Heads 0-3 share KV Group 0 (red border), Heads 4-7 share KV Group 1 (green border)
CAVEAT: Patterns from random initialization, not trained weights



Example 4: FLOPs Breakdown -- Projection vs Attention Core

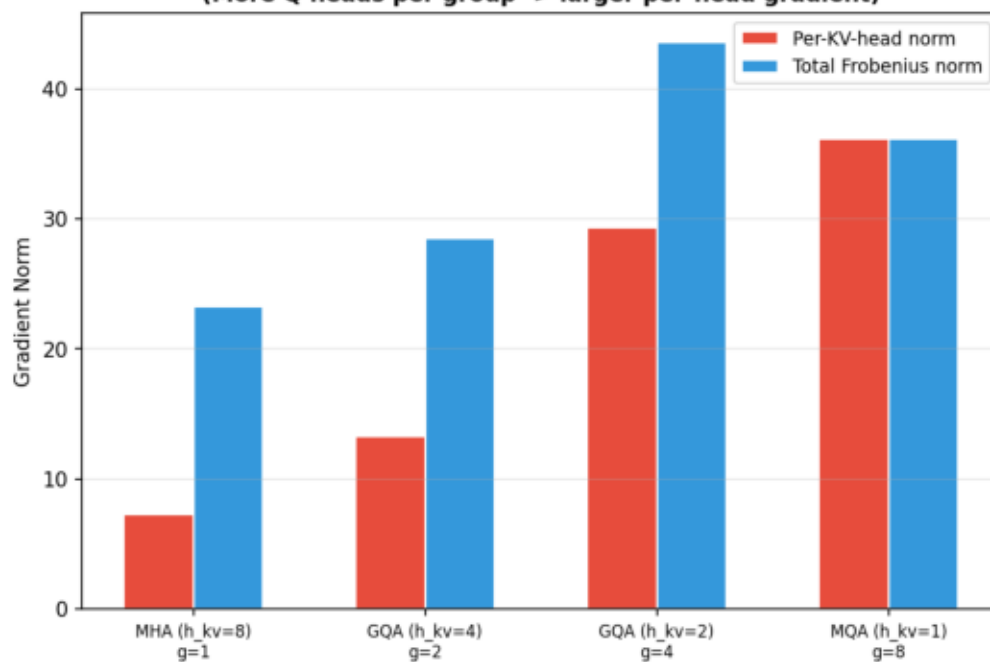
FLOPs Analysis: MHA vs GQA vs MQA (d=768, h=12)



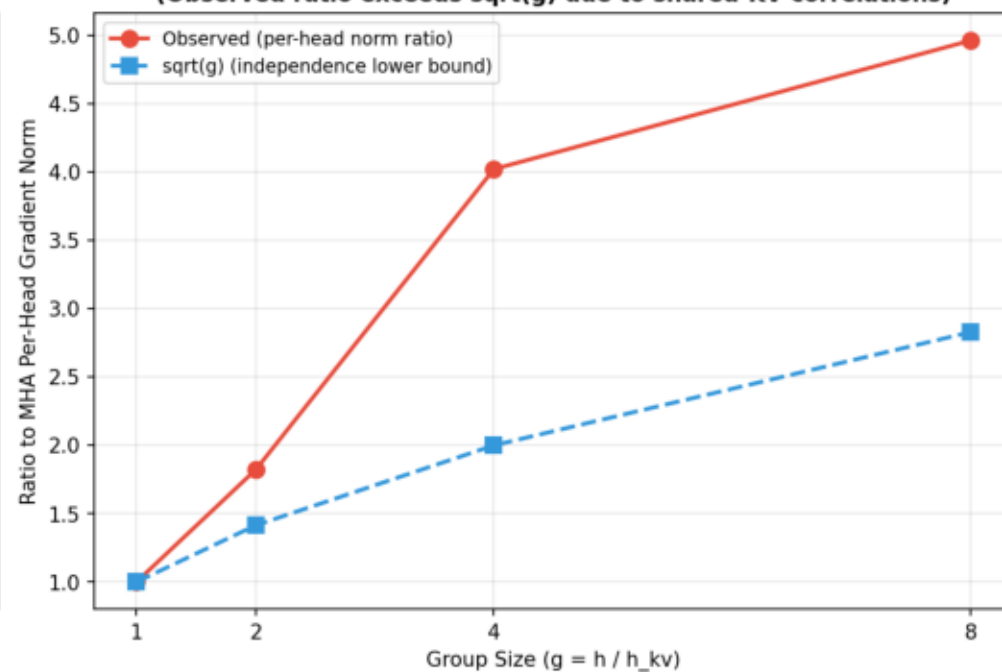
Example 5: Gradient Flow and Accumulation Verification

Gradient Flow in GQA: Multiple Q Heads Accumulate into Shared KV Heads

grad W_K: Per-Head vs Total Norm
(More Q heads per group -> larger per-head gradient)

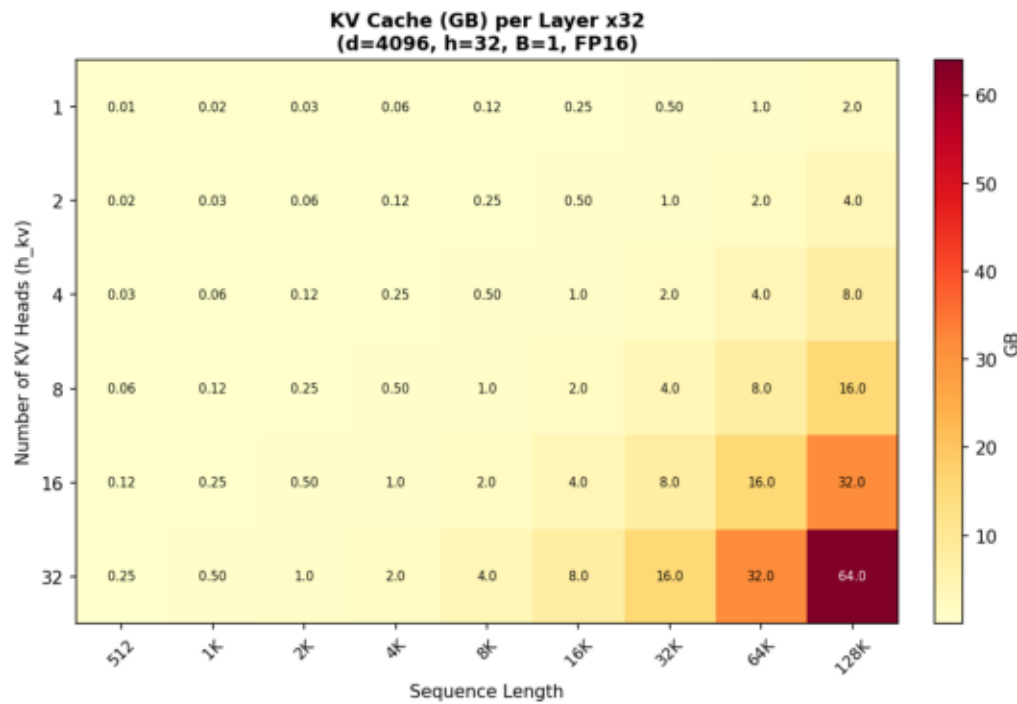


Gradient Accumulation Scaling
(Observed ratio exceeds sqrt(g) due to shared-KV correlations)



Example 6: Scaling Analysis -- Memory Heatmap and Recommendations

KV Cache Memory Scaling: The num_kv_heads vs Sequence Length Tradeoff



KV Cache Growth vs Sequence Length
(with GPU memory budget lines)

