Flash Attention Concept

The Algorithm That Makes Long-Context Attention Tractable

Flash attention reformulates standard attention to avoid materializing the full N x N attention matrix. By processing Q, K, V in tiles and using online softmax to incrementally compute exact results, it reduces attention memory from O(N^2) to O(N) while producing numerically identical output.

This demo covers:

1. Numerical equivalence: tiled == standard within float64 precision
2. Memory comparison: O(N^2) vs O(N) with actual byte counts
3. Block size analysis: memory vs iteration trade-offs
4. Tiling visualization: processing order and block coverage
5. Causal masking: correct causal attention with block skipping
6. Scaling analysis: projections to 8K, 32K, 128K contexts
7. No-materialization proof: max intermediate stays constant

Random seed: 42
Number of visualizations: 7
Examples: 7

Generated by demo.py

Mathematical Foundation

Standard Attention (the problem)

Given $Q, K, V \in \mathbb{R}^{N \times d}$, compute:

$$S = \frac{QK^\top}{\sqrt{d}} \in \mathbb{R}^{N \times N}$$

$$P = \text{softmax}(S) \in \mathbb{R}^{N \times N} \qquad O = PV \in \mathbb{R}^{N \times d}$$

Peak memory: $O(N^2)$ for S and P matrices

Online Softmax (the key insight)

Softmax can be computed in streaming chunks. For chunk $x^{(j)}$:

$$m_{\text{new}} = \max(m, \max(x^{(j)}))$$

$$\ell = \ell \cdot \exp(m - m_{\text{new}}) + \sum_i \exp(x_i^{(j)} - m_{\text{new}})$$

$$m \leftarrow m_{\text{new}}$$

$$\text{Final: softmax}(x)_i = \exp(x_i - m) \,/\, \ell$$

The correction factor $\exp(m_{\text{old}} - m_{\text{new}})$ *rescales previous sums.*

Memory Analysis

Standard: Memory $= O(N^2)$ (must store full $N \times N$ matrices S and P)

Flash: Memory $= O(N)$ (only per-row statistics m, ℓ and output O)

Block matrices S_{ij}, P_{ij} are $O(B_r \cdot B_c) = O(1)$ relative to N

Tiled Attention Algorithm (Flash Attention)

Algorithm

Initialize: $m_i = -\infty,\ \ell_i = 0,\ O = \mathbf{0}_{N\times d}$ for all i

For each KV block j: $K_j = K[jB_c : (j+1)B_c],\ V_j = V[jB_c : (j+1)B_c]$

For each Q block i: $Q_i = Q[iB_r : (i+1)B_r]$

Compute block scores:

$$S_{ij} = \frac{Q_i K_j^\top}{\sqrt{d}} \in \mathbb{R}^{B_r \times B_c}$$

Online softmax update:

$$\tilde{m}_{ij} = \mathrm{rowmax}(S_{ij}), \quad \tilde{P}_{ij} = \exp(S_{ij} - \tilde{m}_{ij}), \quad \tilde{\ell}_{ij} = \mathrm{rowsum}(\tilde{P}_{ij})$$

Combine statistics:

$$m_{\text{new}} = \max(m_i, \tilde{m}_{ij})$$

$$\alpha = \exp(m_i - m_{\text{new}}), \quad \beta = \exp(\tilde{m}_{ij} - m_{\text{new}})$$

$$\ell_{\text{new}} = \ell_i \cdot \alpha + \tilde{\ell}_{ij} \cdot \beta$$

Update output:

$$O_i = \frac{O_i \cdot \alpha \cdot \ell_i + (\tilde{P}_{ij} \cdot \beta) V_j}{\ell_{\text{new}}}$$

$$m_i \leftarrow m_{\text{new}}, \quad \ell_i \leftarrow \ell_{\text{new}}$$

Key Property

The rescaling $\alpha = \exp(m_{\text{old}} - m_{\text{new}})$ corrects all previous accumulations when a new block reveals a larger maximum. This yields EXACT softmax.

Summary of Findings

1. NUMERICAL EQUIVALENCE: Tiled attention produces bit-identical results to standard attention for all tested configurations (N up to 1024, various block sizes including non-divisible N). Online softmax is numerically stable even for extreme values (+/-1000). Max error < 1e-10.

2. MEMORY COMPARISON: Standard attention requires O(N^2) memory for the S and P matrices. Tiled attention requires O(N) -- only block-sized intermediates plus per-row statistics. At N=8192 (d=64, B=32, FP32) the ratio exceeds 240x, growing linearly with N.

3. BLOCK SIZE TRADE-OFFS: Smaller blocks use less peak memory but require more tile iterations. Larger blocks reduce iterations but need more memory per tile. Total computation is always N^2 * d regardless of block size. In practice, block size is chosen to fit GPU SRAM (~48 KB).

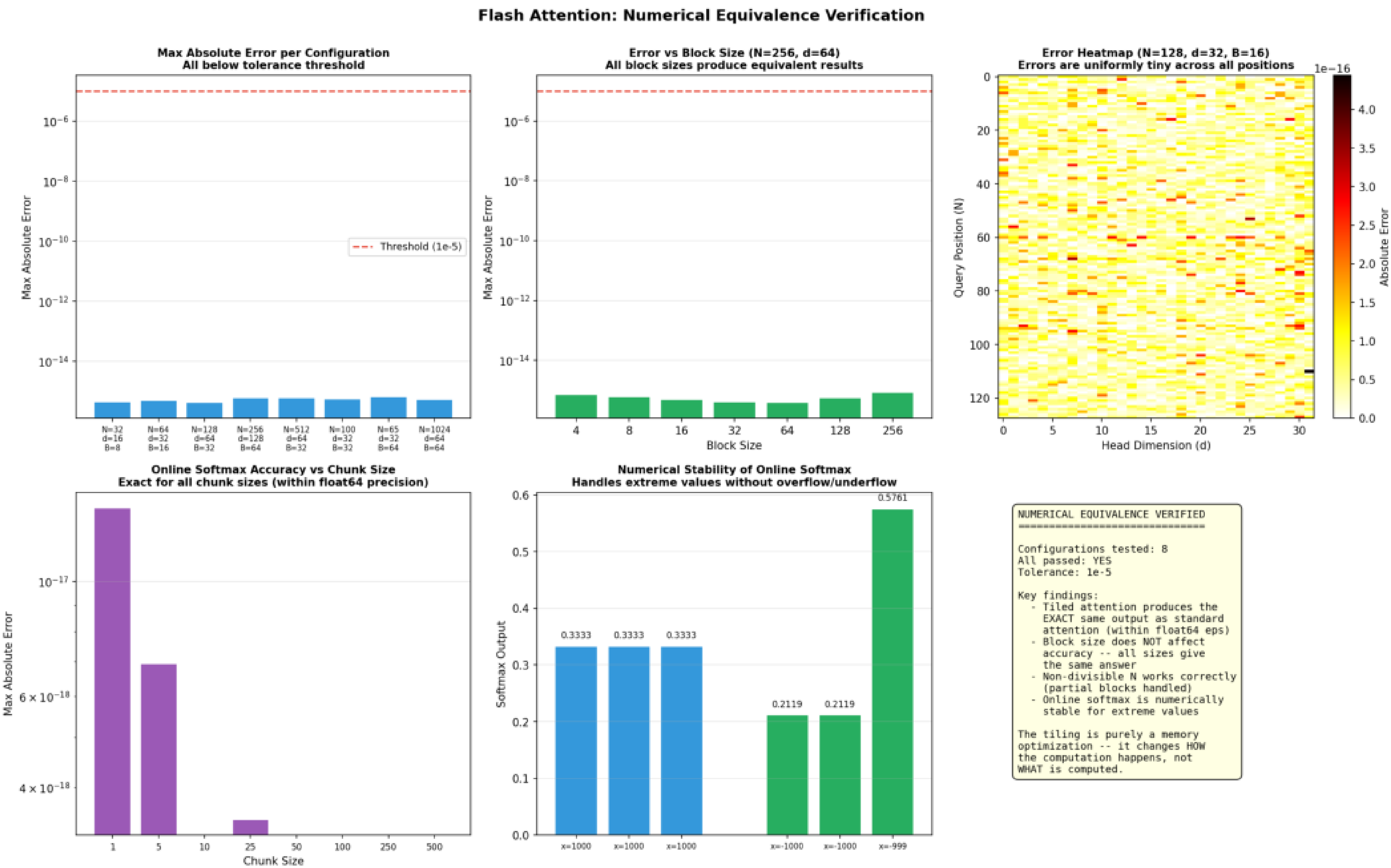
4. TILING VISUALIZATION: The algorithm processes tiles in outer-KV, inner-Q order. K/V blocks are loaded once and reused across all Q blocks. For causal masking, ~50% of tiles are entirely above the diagonal and skipped.

5. CAUSAL MASKING: Tiled causal attention matches reference causal attention within 1e-5 tolerance, including non-divisible sequence lengths and asymmetric block sizes. Block-level skipping provides ~50% compute savings.

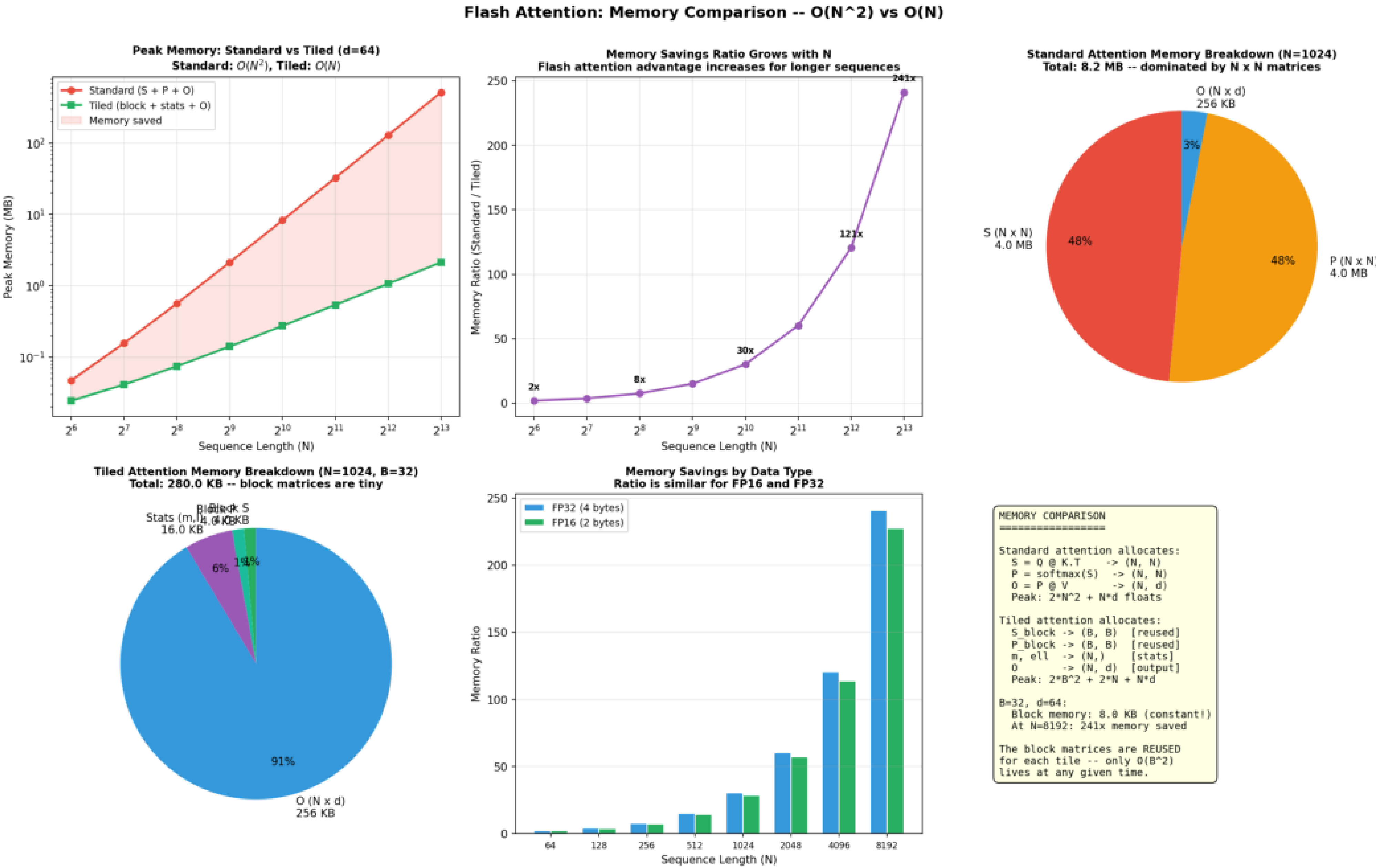
6. SCALING ANALYSIS: At 128K context with FP16, the standard attention matrix alone requires 32 GB per head per layer. Flash attention keeps intermediates at a few KB regardless of context length, enabling 128K+ context models that would be impossible with standard attention.

7. NO-MATERIALIZATION PROOF: Instrumented tracking confirms the largest intermediate tensor stays constant (B*d elements) as N grows from 64 to 2048. At N=2048 this is 4,096x smaller than the N^2 standard matrix.

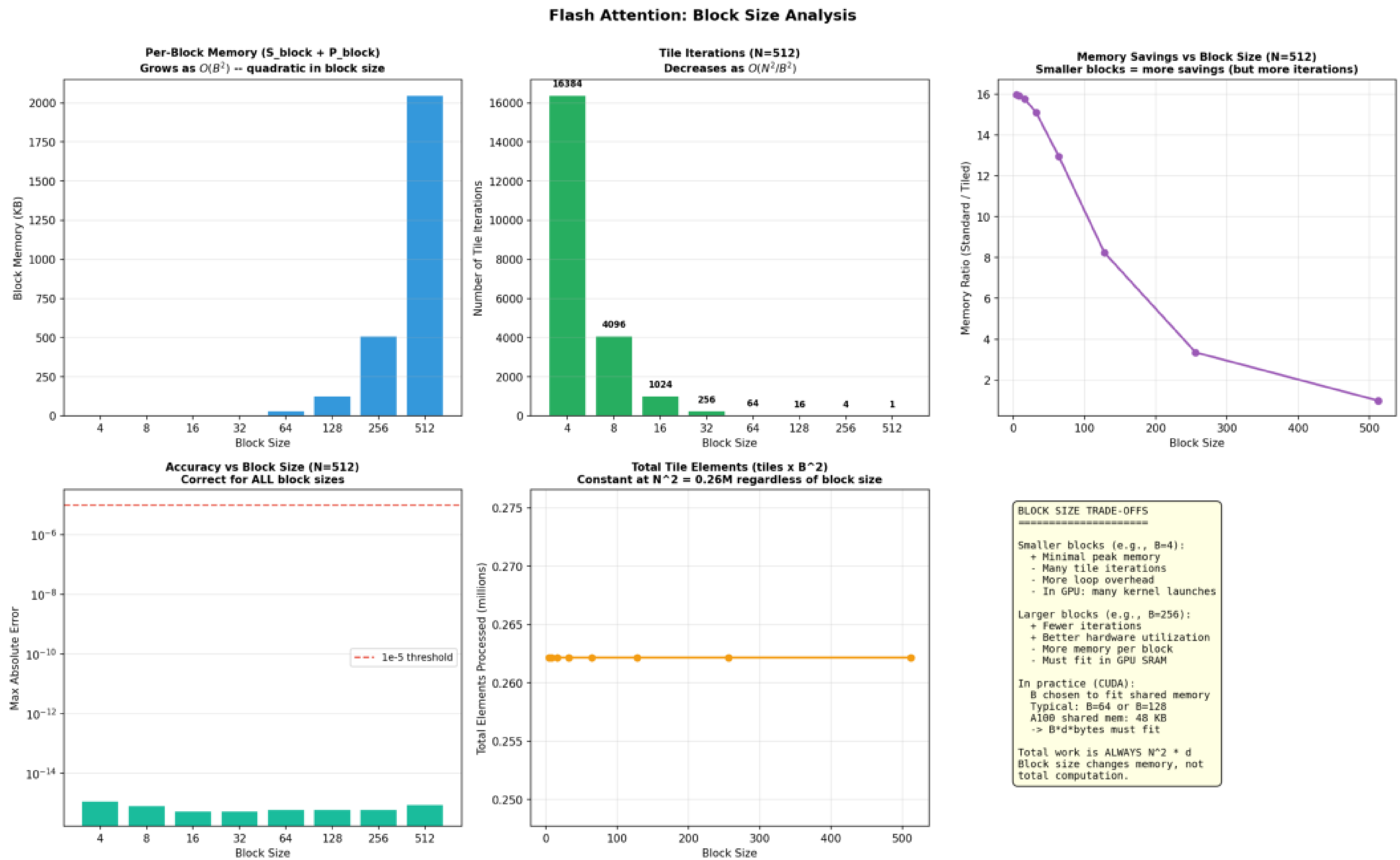
Example 1: Numerical Equivalence Verification



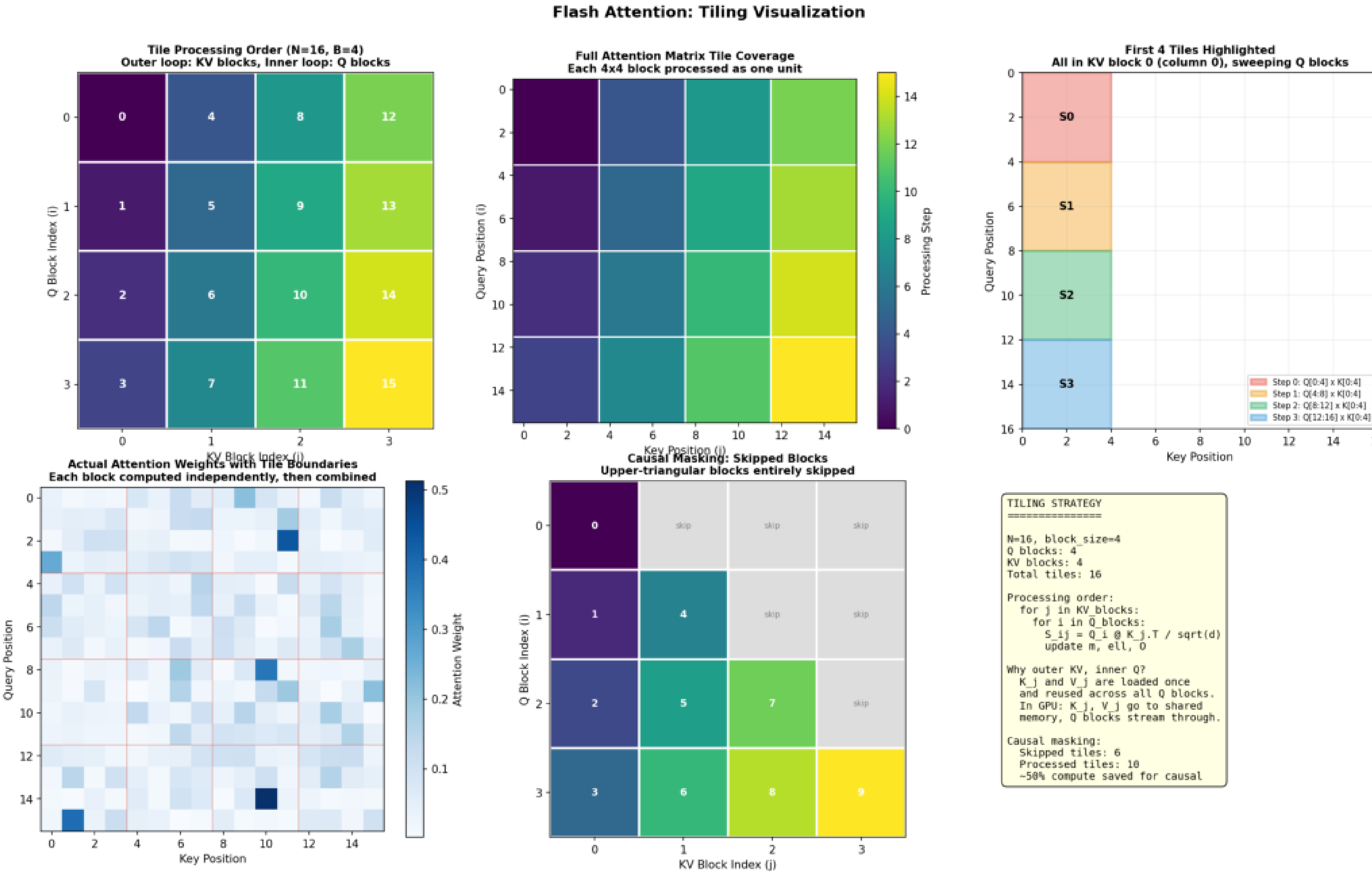
Example 2: Memory Comparison -- O(N^2) vs O(N)



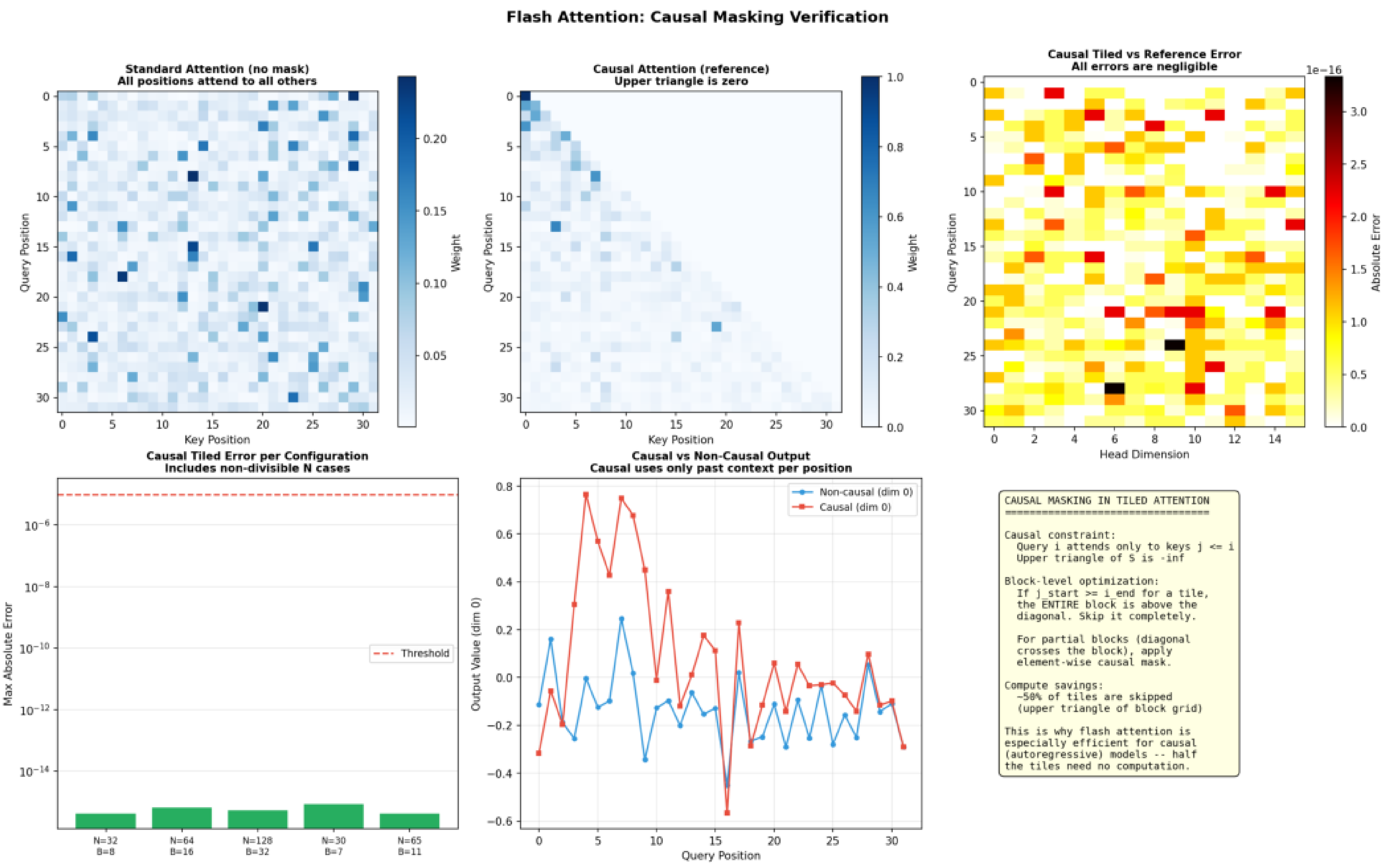
Example 3: Block Size Analysis



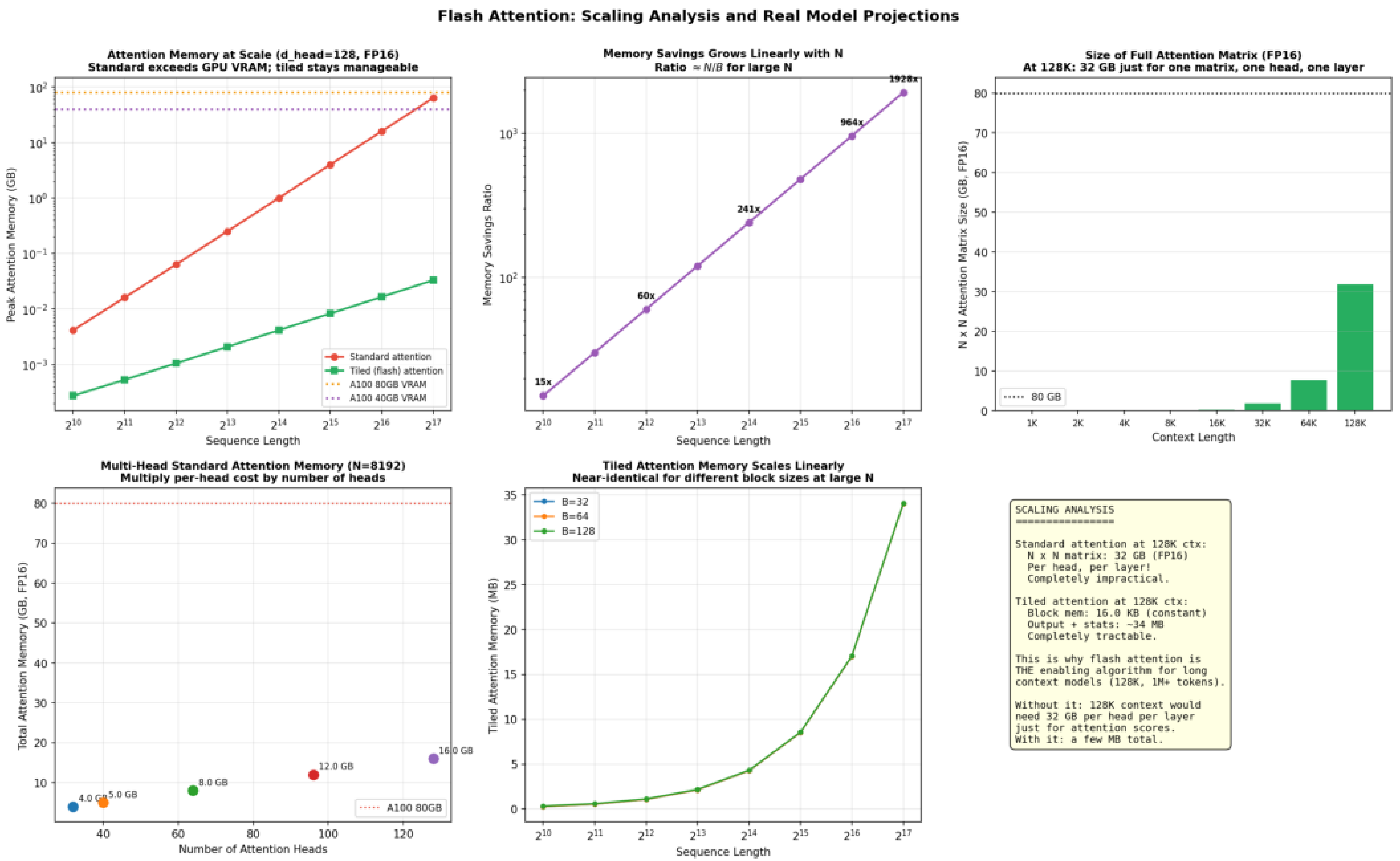
Example 4: Tiling Visualization



Example 5: Causal Masking Verification



Example 6: Scaling Analysis and Real Model Projections



Example 7: No-Materialization Proof