Efficient Self-Attention Mechanisms Via Vector Quantization

George Ankeney
ICME, Stanford University
ankeneyg@stanford.edu

Juan Muneton Gallego
ICME, Stanford University
jmuneton@stanford.edu

Abstract

- **Motivation**: Transformers revolutionized NLP, audio, and computer vision but face quadratic time and space complexity for long-sequence tasks.
- **Challenges with existing solutions**:
  - Sparse factorizations, LSH, and linear attention reduce complexity.
  - Issues include gradient instability and degraded inference performance.
- **Our contribution**:
  - Introduced a novel self-attention mechanism using **VQ-VAEs** to achieve sub-quadratic runtime.
  - Compresses keys and queries in self-attention using vector quantization for efficient, stable attention over long contexts.
- **Performance highlights**:
  - Superior throughput, memory efficiency, and computational performance.
  - Scalable across sequence lengths: 10^3 to 10^5.

Methodology

- **Overview**:
  - Developed a self-attention mechanism that quantizes both keys ($\mathbf{K}$) and queries ($\mathbf{Q}$) using codebook representations.
  - Introduced two learnable vector quantizers to map rows of $\mathbf{Q}$ and $\mathbf{K}$ to their respective quantized representations.

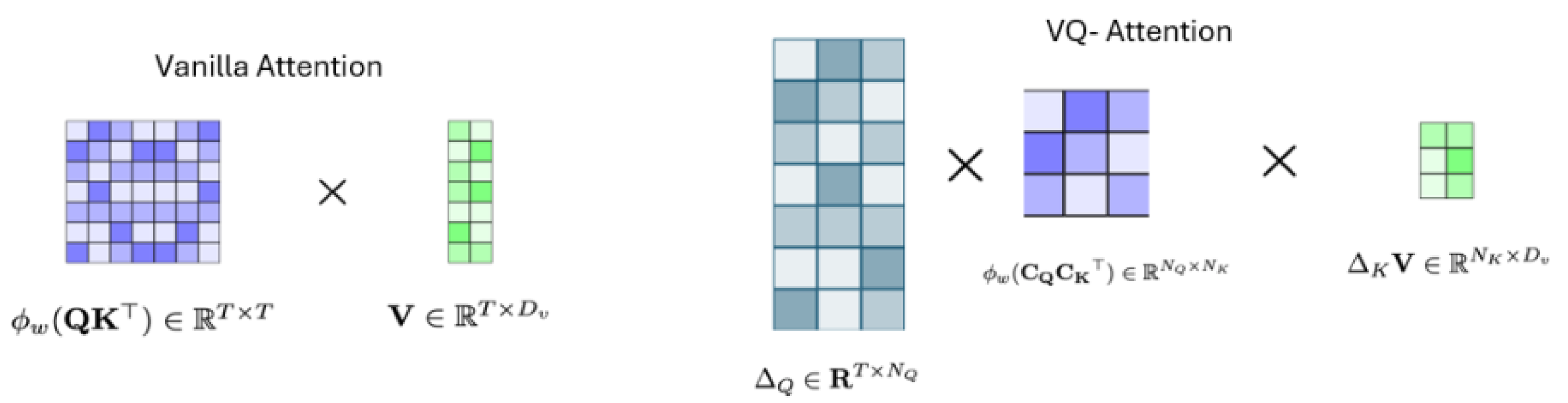

Figure 1. Visual representation of vector quantization

- **Proposed Self-Attention Mechanism**:
  - For queries ($\mathbf{Q}$) and keys ($\mathbf{K}$), quantized representations ($\hat{\mathbf{Q}}$, $\hat{\mathbf{K}}$) enable efficient attention computation.
  - Attention weights are approximated as:

$$\mathbf{W} \approx \phi_w(\mathrm{VQ}(\mathbf{Q};\mathbf{C_Q})\mathrm{VQ}(\mathbf{K};\mathbf{C_K})) = \phi_w(\widehat{\mathbf{Q}}\widehat{\mathbf{K}}^\top). \tag{1}$$

  - Final computation leverages the associative property of matrix multiplication for sub-quadratic time complexity:

$$\mathbf{W} = \mathrm{Diag}\left(\Delta_Q \mathbf{M} \Delta_K \mathbf{1}\right)^{-1} \Delta_Q \mathbf{M} \Delta_K, \tag{2}$$

where $\mathbf{M} = \exp(\mathbf{C_Q C_K}^\top)$ and Δ matrices select code vectors.

Results

Varying Sequence Length

- Evaluated performance of hyperattention, vanilla attention, and VQ attention as sequence input length increased.
- Setup: Batch size = 1, head size = 1, embedding dimension = 512, codebook size = 128.
- Pre-computed keys, queries, values, and Δ_Q, Δ_K matrices to isolate computational cost of attention step.

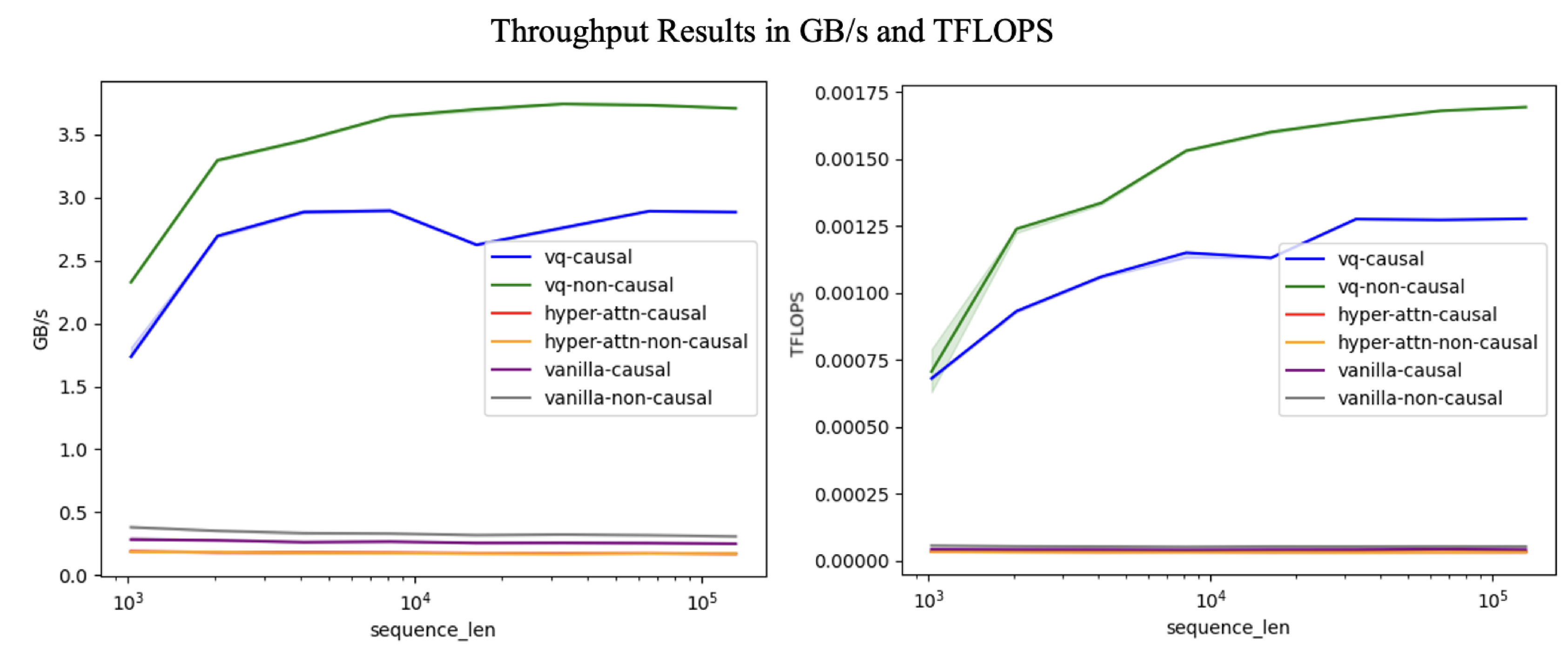

Figure 2. Left figure shows the comparison of the GB/s and right figure provides TFLOPs comparison over increasing sequence lengths

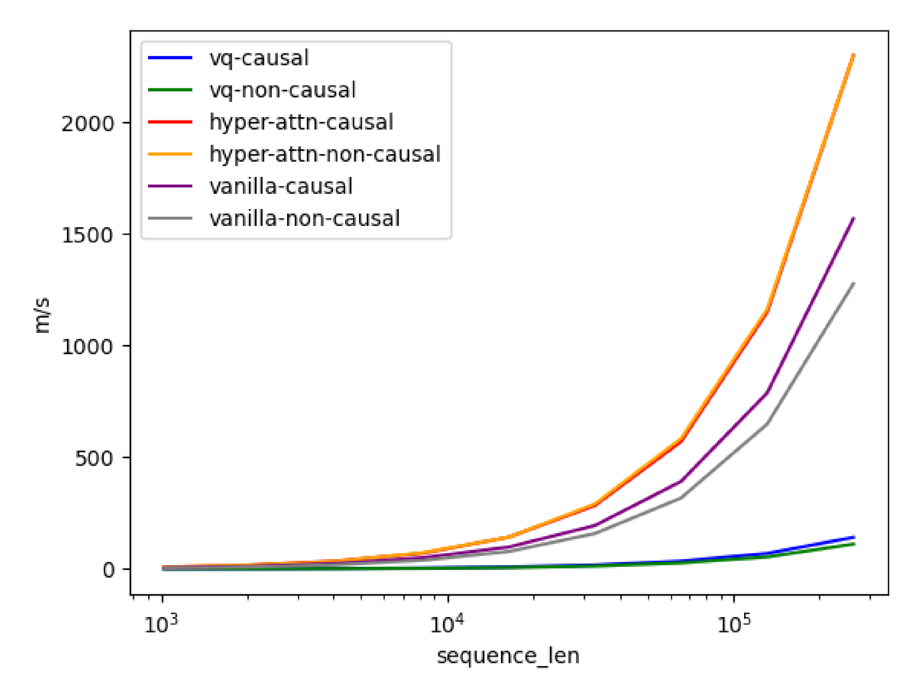

Figure 3. Runtime comparison for varying sequence lengths

- VQ-non-causal models slightly outperformed VQ-causal models in throughput test.
- Both configurations significantly surpassed vanilla and hyperattention baselines.
- VQ attention consistently demonstrated superior throughput and efficiency for varying sequence lengths.

Results (Cont.)

Varying Codebook Size

- Fixed sequence length (16k) and varied codebook size from approximately 10^2 to 10^4.

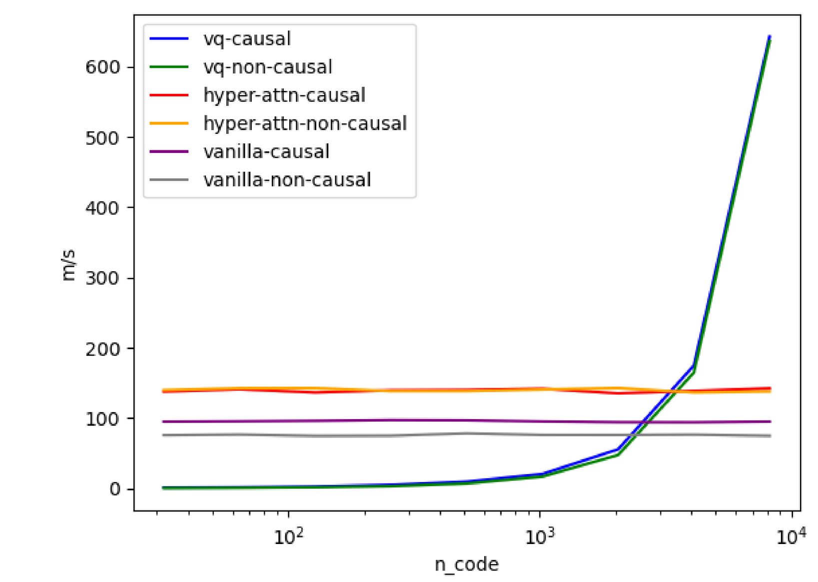

Figure 4. Runtime comparison for varying codebook sizes

- Larger codebooks improved throughput and efficiency but caused performance degradation when excessively large.

Discussion and Future Work

Key Takeaway

This study demonstrates that vector quantization of queries and keys offers a promising alternative for optimizing self-attention mechanisms. By mitigating runtime dependence on sequence length, this approach enables more efficient and scalable implementations of self-attention in large-scale models. However, careful selection of the codebook size is critical for optimizing self-attention performance.

Future Work

- Explore CUDA-enabled evaluations and utilizing tools such as Triton and Faiss to widen the scope of analysis and optimize the algorithm.
- Develop formal scaling laws to predict the feasibility of this approach for larger models.
- Examine the impact of varying quantized query and key sizes on model performance.

References

[1] Lucas D. Lingle. *Transformer-VQ: Linear-Time Transformers via Vector Quantization*. Feb. 25, 2024. arXiv: 2309.16354[cs]. URL: http://arxiv.org/abs/2309.16354 (visited on 10/05/2024).

[2] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. *Neural Discrete Representation Learning*. May 30, 2018. arXiv: 1711.00937[cs]. URL: http://arxiv.org/abs/1711.00937 (visited on 10/05/2024).