EFFICIENT TRANSFORMERS VIA MPO-BASED LOW-RANK FACTORIZATION AND PRUNING

Sam Mikhak **Venkata Sai Gummidi** **Praneeth Medepalli** **Kevin Zhu**
Algoverse AI Research
sammikhak@gmail.com | kevin@algoverseacademy.com

ABSTRACT

We explore the use of matrix product operators (MPOs) to compress transformer-based architectures. By factorizing full-rank weight matrices into tensor-train product, MPOs reduce both memory footprint and computational cost, which is critical for deployment on resource-constrained devices. Our experiments on speaker identification using the LibriSpeech train-clean-360 subset show that MPO-based models, and even their pruned variants, maintain high performance with far fewer parameters than full-rank transformers. We detail the mathematical principles underlying low-rank factorization and unstructured pruning and discuss next steps for extending this approach to more complex tasks such as automatic speech recognition (ASR).

1 INTRODUCTION

Transformers have achieved remarkable performance in audio processing tasks such as speaker identification(1). However, their high computational cost can hinder deployment in low-resource or real-time environments. Although Matrix Product Operator (MPO) techniques have been extensively studied in quantum physics and natural language processing(2)(3), their application to speech tasks remains relatively under-explored. In a tensor train (or MPO) framework, a large, high-dimensional weight matrix is decomposed into a series of smaller, interconnected core tensors. Instead of storing and computing with a single, huge matrix, the model uses several compact matrices that multiply together to approximate the original weights. This dramatically reduces the number of parameters and computational cost (3). In our implementation, we simplify this further by using just two core tensors (i.e., a low-rank approximation), which still retains much of the expressive power of the full model while enabling efficient training and inference. Please refer to the appendix for a more detailed mathematical explanation.5

In this work, we apply MPO-based compression to transformer models for speaker identification to evaluate the effectiveness of MPO compression in reducing computational complexity while maintaining performance. The LibriSpeech dataset, with its well-defined speaker labels (which has been used in training transformers before), provides an ideal environment for this investigation. Our approach compresses the transformer-based speaker identification model by replacing full-rank weight matrices with MPO decompositions applied to either specific sub-modules or the entire model.

Our experiments demonstrate MPO-based compression can make transformer models more efficient on audio tasks, providing a proof of concept for broader applications. Preliminary results suggest synergy with pruning, and although we tested only on an audio benchmark, the method generalizes to other transformers and domains.

2 EXPERIMENTAL SETUP AND RESULTS

2.1 DATA, PREPROCESSING, AND TRAINING PARAMETERS

We perform experiments on the LibriSpeech train-clean-360 subset, which contains 104,014 audio samples from 921 speakers. The dataset is split into training (80%), validation (10%), and test (10%) sets. Each audio file is converted into a log Mel spectrogram using a 1024-point FFT, a hop length

of 160, and 128 Mel bands. Waveforms are padded or trimmed to 30 seconds to ensure uniform input dimensions. In our MPO-based compression, we chose a preliminary core count of $n = 2$ to decompose the weight matrices, striking a balance between compression and performance. Further parameter settings are provided in the Appendix.5

2.2 Model Architectures

In our work, we compare five transformer-based architectures for speaker identification. The first is the Vanilla transformer, which employs full-rank linear layers throughout the network. The second, Full MPO, replaces every linear transformation in the transformers encoder with an MPO-based low-rank factorization. The third variant, MPO (Attention Only), applies MPO compression exclusively to the self-attention projections (i.e., the query, key, and value matrices). The fourth, MPO (Feed-Forward Only), compresses only the feed-forward layers using the MPO framework. Finally, the MPO Pruning model combines MPO factorization with L1-norm unstructured pruning to further reduce the number of non-zero parameters.

3 Results

Our experiments show that MPO-based compression robustly preserves transformer performance at low parameter counts compared to both full-rank and pruning approaches (Refer to Table 1). The vanilla transformers suffers marked degradation below 200K parameters, whereas MPO models maintain high accuracy even around 110K parameters using only 27.51% of the corresponding vanilla weights. For the MPO variants, selective MPO—particularly when applied solely to the encoder layers—yields the best results, while MPO on only the feed-forward layers limits accuracy to 85.38%. Compared to a pruning approach that reduces the model to 113K non-zero parameters with test accuracies up to 97.05%, the full MPO model (approximately 130K parameters) delivers comparable performance. This MPOPruning decomposition not only compresses the model but also stabilizes training dynamics compared to the other MPO variants. (Please refer to graphs in appendix)5. Overall, our results indicate that strategic MPO application, especially within the attention submodules, achieves the optimal balance between efficiency and accuracy.

Model Type	**Test Accuracy (%)**	**Parameter Count/ Compression Ratio**	**Hidden Dim**
Vanilla transformer	92.55	433K	128
Full MPO	93.78	128K / 27.51%	128
MPO+ L1 Pruning (Attention)	97.05	113K / 26.09%	128
Vanilla transformer	88.34	217K	64
MPO (Feed-Forward)	85.18	110K / 50.65%	64
MPO (Attention)	94.26	115K / 52.95%	64

Table 1: Test Accuracy, Parameter Count for Vanilla, MPO Full, MPO Attention, MPO Feed Forward, and MPOPruning models with LibriSpeech clean-360 test dataset.

4 Conclusion and Next Steps

Our experiments demonstrate that MPO-based compression enables transformer models to maintain high performance with drastically reduced parameter counts. In particular, while the full MPO model (compressing both attention and feed-forward layers) achieves test accuracies above 90%, selective MPO—especially when applied to the attention submodules—yields even better performance. In contrast, applying MPO solely to the feed-forward layers limits accuracy to below 85.38%. Compared to traditional pruning, which discards weights based on magnitude (3; 4), the MPO approach factorizes weight matrices into lower-rank components, preserving essential information while ensuring stable training dynamics. Future work will focus on further fine-tuning and hybrid compression techniques to scale these methods to more complex tasks, such as but not limited to automatic speech recognition (ASR). ASR systems must capture complex acoustic patterns and long-range dependencies, which challenges model capacity and stability (5; 6). We acknowledge

the reviewers' observations about our use of only a two-core approach—which effectively resembles the application of LORA adapters—in this work. It's important to note that this experiment was a preliminary investigation aimed at testing the viability of the approach in its simplest form. Further work will focus on increasing the number of cores considerably. We will also integrate MPO-based compression into pre-trained models, leveraging their rich, transferable representations to achieve efficient compression through MPO and pruning techniques.

REFERENCES

[1] K. Wei, P. Guo, and N. Jiang, "Improving transformer-based conversational asr by inter-sentential attention mechanism," 2022. [Online]. Available: https://arxiv.org/abs/2207.00883

[2] B. Žunkovič, "Deep tensor networks with matrix product operators," *Quantum Machine Intelligence*, vol. 4, no. 2, p. 21, 2022.

[3] Z.-F. Gao, S. Cheng, R.-Q. He, Z. Y. Xie, H.-H. Zhao, Z.-Y. Lu, and T. Xiang, "Compressing deep neural networks by matrix product operators," *Physical Review Research*, vol. 2, no. 2, Jun. 2020. [Online]. Available: http://dx.doi.org/10.1103/PhysRevResearch.2.023300

[4] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," 2019. [Online]. Available: https://arxiv.org/abs/1905.09418

[5] H. Kawano and S. Shimizu, "An effective transformer-based contextual model and temporal gate pooling for speaker identification," 2023. [Online]. Available: https://arxiv.org/abs/2308.11241

[6] W. Kwon, S. Kim, M. W. Mahoney, J. Hassoun, K. Keutzer, and A. Gholami, "A fast post-training pruning framework for transformers," 2022. [Online]. Available: https://arxiv.org/abs/2204.09656

5 APPENDIX

5.1 GRAPHS

Graphs are for the tabular results in section 4

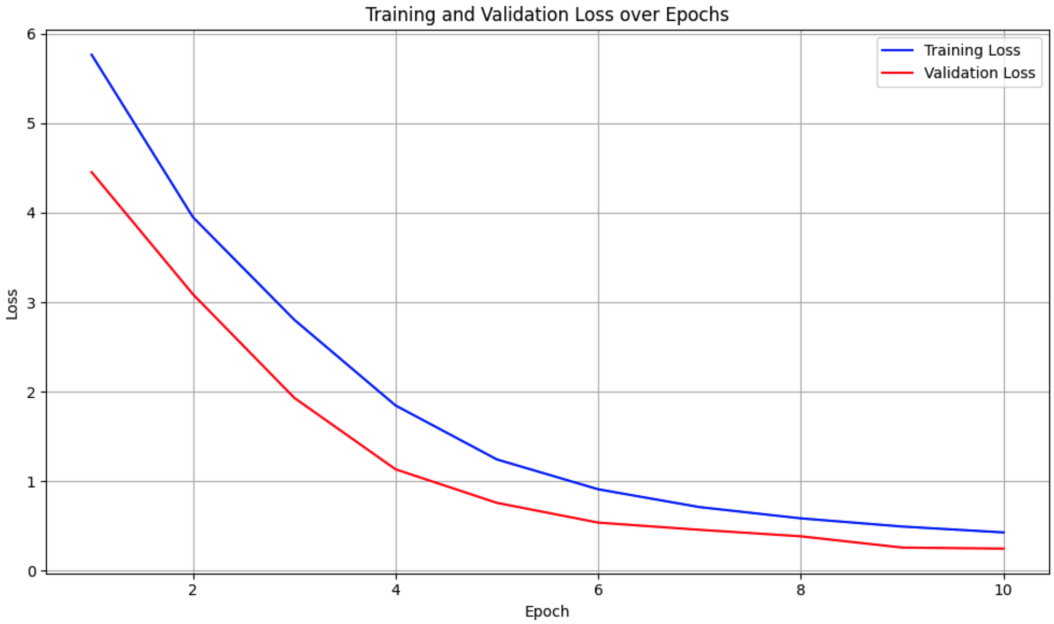

Figure 1: Full MPO Graph

Figure 2: MPO Feed Forward Only

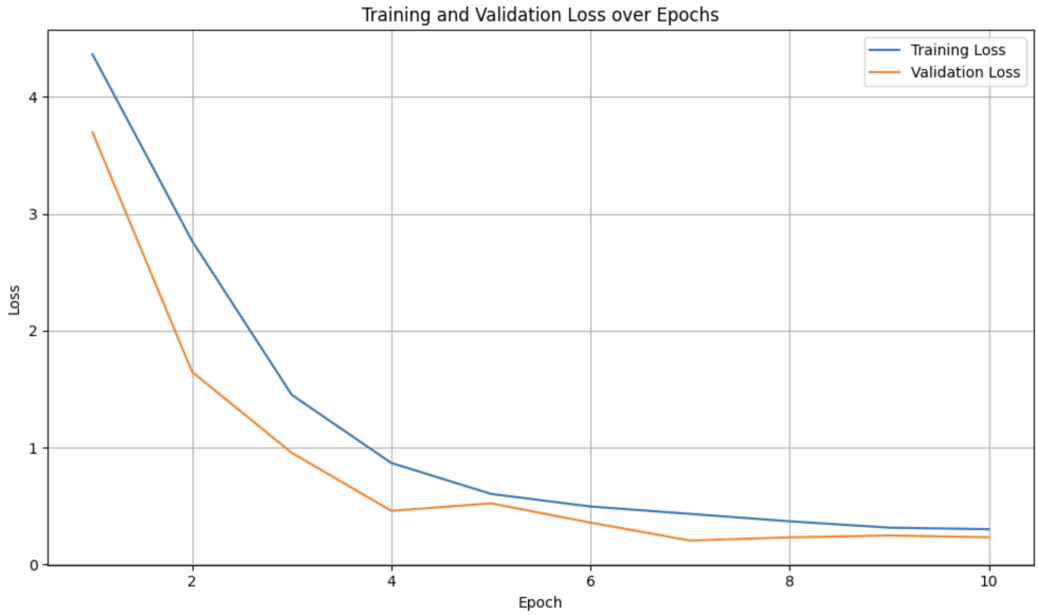

Figure 3: MPO Encoder Only

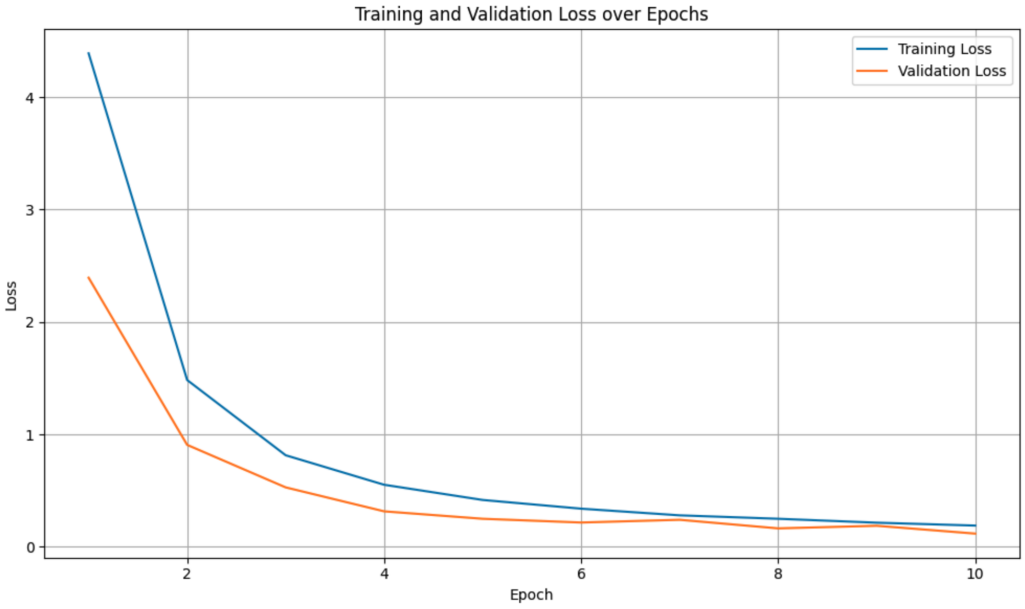

Figure 4: MPOPruning Model Graph

5.2 MPO-Based Compression

Forward Pass Computation: Forward Pass Computation: In our approach, a full weight matrix

$$\mathbf{W} \in \mathbb{R}^{M \times N}$$

is approximated via an MPO decomposition as

$$\mathbf{W} \approx \mathbf{c}_1 \, \mathbf{c}_2 \cdots \mathbf{c}_n,$$

where each factor

$$\mathbf{c}_i \in \mathbb{R}^{d_{i-1} \times d_i}, \quad d_0 = M, \quad d_n = N,$$

and the bond dimension (i.e., the maximum of the intermediate dimensions) controls the trade-off between compression and accuracy. The effective weight is then given by

$$\mathbf{W}_{\text{eff}} \approx \mathbf{c}_1 \, \mathbf{c}_2 \cdots \mathbf{c}_n,$$

and the layer output is computed as

$$\mathbf{y} \approx \mathbf{x} \, (\mathbf{c}_1 \, \mathbf{c}_2 \cdots \mathbf{c}_n)^\top + \mathbf{b}.$$

For computational efficiency, we fuse these operations via Einstein summation:

$$\mathbf{y} = \text{einsum}\Big(\text{"}bi, ij, jk, \ldots, \ell m \to bm\text{"}, \, \mathbf{x}, \, \mathbf{c}_n^\top, \, \mathbf{c}_{n-1}^\top, \, \ldots, \, \mathbf{c}_1^\top\Big) + \mathbf{b}.$$

Integration in Transformer Layers: This MPO-based compression replaces the full-rank weight matrices in both the self-attention submodule (for computing queries, keys, values, and the output projection) and the feed-forward submodule of each Transformer encoder layer. This replacement yields substantial parameter savings and reduced computational complexity while maintaining the model's expressive capacity.

5.3 Computational Complexity Comparison

In a standard fully-connected layer, the forward pass involves a matrix multiplication between an input vector $\mathbf{x} \in \mathbb{R}^N$ and a weight matrix $\mathbf{W} \in \mathbb{R}^{M \times N}$, which has a computational complexity of $\mathcal{O}(M \cdot N)$. In contrast, the MPO approach decomposes $\mathbf{W}$ into a product of n smaller matrices:

$$\mathbf{W} \approx \mathbf{c}_1 \, \mathbf{c}_2 \cdots \mathbf{c}_n,$$

where each factor $\mathbf{c}_i \in \mathbb{R}^{d_{i-1} \times d_i}$ with the boundary conditions $d_0 = M$ and $d_n = N$. The forward pass is computed as

$$\mathbf{y} \approx \mathbf{x} \, (\mathbf{c}_1 \, \mathbf{c}_2 \cdots \mathbf{c}_n)^\top + \mathbf{b}.$$

Rather than performing a single $\mathcal{O}(M \cdot N)$ multiplication, this formulation replaces it with a sequence of multiplications with a total cost of

$$\mathcal{O}\Big(\sum_{i=1}^{n} d_{i-1} \, d_i\Big),$$

which is considerably lower when the intermediate dimensions d_i (controlled by the bond dimension) are significantly smaller than M and N. Furthermore, by implementing these operations as fused tensor contractions (e.g., via Einstein summation),

$$\mathbf{y} = \mathrm{einsum}\Big(\text{"}bi, ij, jk \to bk\text{"}, \, \mathbf{x}, \, \mathbf{c}_n^\top, \, \cdots, \, \mathbf{c}_1^\top\Big) + \mathbf{b},$$

the MPO method leverages efficient parallelization on modern GPU architectures while preserving the expressive power of the original full-rank matrix.

6 Training Parameters

Our training configuration includes a batch size of 8, 10 epochs, the AdamW optimizer with a learning rate of 5×10^{-4}, and a bond dimension $r = 2$ for the MPO factorization. These hyperparameters were chosen based on preliminary experiments that balanced computational efficiency, training stability, and overall performance under a resource-constrained environment. In particular, the learning rate of 5×10^{-4} yielded the best convergence and results when computational resources were limited. Additionally, we employ a hidden dimension of 128, 2 attention heads, 1 encoder layer, and a feed-forward dimension of 1024 to maintain sufficient model capacity while ensuring compactness.