Whitened Self-Attention

Anonymous ACL submission

Abstract

Self-attention in Transformer-based generative models such as GPT, implicitly assumes that context tokens are independent and identically distributed. However, this contradicts the very premise of attention: that the meaning of words is influenced by their complex interdependencies. We propose whitened self-attention, a filter that optimally accounts for inter-token correlations and show it enhances representation learning for autoregressive language modeling. Experiments on a small GPT architecture demonstrate an 11% improvement in perplexity, an equivalent performance in 13x fewer iterations, and after optimizations, a lowered training time by up to 42%. This work advances self-attention for generative NLP tasks, based on a theoretically grounded method for handling token dependencies, and our method shows promise for improving generalization in large-scale NLP models.

1 Introduction

The Transformer model (Vaswani et al., 2017) is a popular and successful deep learning architecture used in a wide array of applications areas such as NLP (Kalyan et al., 2021), computer vision (Khan et al., 2022; Han et al., 2022), speech recognition (Gulati et al., 2020), and computational biology (Zhang et al., 2023). That said, the core component, self-attention, is more of a heuristic than a precisely formulated, optimally derived filter. Attention estimates a target vector, $x_n \in \mathcal{R}^d$ based on a weighted sum of its context vectors, $\{x_i\} \in \mathcal{R}^d$ (Bahdanau et al., 2014). The autoregressive formulation used in GPT architectures takes the form

$$\text{Att}(x_N) = \sum_{i=0}^{N-1} \text{softmax}(\frac{x_N^T Q^T K x_i}{\sqrt{d}}) V x_i \quad (1)$$

$$= \sum_{i=0}^{N-1} \left[\frac{\exp(\frac{x_N^T Q^T K x_i}{\sqrt{d}})}{\sum_{j=1}^{N-1} \exp(\frac{x_N^T Q^T K x_j}{\sqrt{d}})} \right] V x_i, \quad (2)$$

where the Q, K, and V are learned matrices. The softmax terms in Equation 1 are positive scalars summing to one, and they estimate the relative information each x_i has about x_N. For this formulation to provide a minimum variance estimate, the Gauss-Markov theorem implies the x_i should be independent and identically distributed random vectors (Shaffer, 1991). If they are not i.i.d. the estimator is suboptimal. When training with very large datasets, it is possible that the variance approaches an optimal value, but intuitively, it is doubtful this occurs uniformly for all token embeddings across the entire input vocabulary.

Whitening is a filtering process that transforms input sequences into stochastically independent outputs, and estimators based on it are optimal, having minimum variance (Kleiner et al., 1979; Kailath, 1970). The rest of this paper develops a computationally feasible whitening operator for self-attention, and presents experimental results showing that whitened attention significantly improves performance when used to train a GPT model.

2 Sequence Whitening

Given an ordered sequence of column vectors $\{x_0, x_1, \ldots, x_{N-1}\}, x_i \in R^D$, a common objective is to autoregressively predict the next vector, x_N, given observations of the preceding context (Akaike, 1969). Typically, the x_i are assumed independent and identically distributed, but if the sequence is correlated it must be whitened to obtain an optimal estimator. Defining the vector $X = [x_0^T, x_1^T, \ldots, x_{N-1}^T]^T \in R^{ND}$, where the x_i are zero-mean random vectors, the covariance matrix $\Lambda_X = E\{XX^T\}$ has a block structure

$$\Lambda_X = \begin{bmatrix} \Lambda_{0,0} & \Lambda_{0,1} & \ldots & \Lambda_{0,N-1} \\ \Lambda_{1,0} & \Lambda_{1,1} & \ldots & \Lambda_{1,N-2} \\ \vdots & \vdots & & \vdots \\ \Lambda_{N-1,0} & \Lambda_{N-1,1} & \ldots & \Lambda_{N-1,N-1} \end{bmatrix}, \quad (3)$$

where $\Lambda_{i,j} = E\{x_i x_j^T\}$. The whitened sequence, $W = [w_0^T, w_1^T, \ldots, w_{N-1}^T]^T \in R^{ND}$, is obtained from X by letting $W = \Lambda_X^{-1/2} X$. That the w_i are independent (that is, whitened) can be verified as follows:

$$\begin{aligned} \Lambda_W &= E\{WW^T\} \\ &= E\{\Lambda_X^{-1/2} X X^T \Lambda_X^{-1/2}\} \\ &= \Lambda_X^{-1/2} E\{XX^T\} \Lambda_X^{-1/2} \\ &= \Lambda_X^{-1/2} \Lambda_X \Lambda_X^{-1/2} \\ &= I. \end{aligned} \tag{4}$$

The whitened sequence, $\{w_i\}$, spans the same subspace as the $\{x_i\}$ but are independent of each other. When substituted into the self-attention expression from Equation 2, the result is an optimized estimator of x_N we call whitened attention (WA),

$$\text{WA}(x_N) = \sum_{i=0}^{N-1} \text{softmax}(\frac{x_N^T Q^T K w_i}{\sqrt{d}}) w_i \tag{5}$$

$$= \sum_{i=0}^{N-1} \left[\frac{\exp(\frac{x_N^T Q^T K w_i}{\sqrt{d}})}{\sum_{j=1}^{N-1} \exp(\frac{x_N^T Q^T K w_j}{\sqrt{d}})} \right] w_i. \tag{6}$$

A superficial difference between the expression for standard attention in Equation 2 and the whitened one in Equation 6 is that the latter has no V matrix. It has been absorbed into the w_i, as is explained in more detail in the next section.

3 Modeling the Covariance Structure

As the matrix Λ_X is $ND \times ND$, its inverse is computationally challenging and memory intensive. For example, in current production-quality LLMs, $ND \approx 10^7$, meaning this single matrix could require more than a petabyte of memory. Some of the computational and memory requirements can be mitigated by recognizing that Λ_X, a covariance matrix, is symmetric, and additional efficiencies can be had by assuming the modeled sequences are wide-sense stationary. The cross-covariance blocks, Λ_{ij}, of a wide-sense stationary process depend only on their separation, $|i-j|$ (Van Trees, 2004; Papoulis and Pillai, 2002). This makes the structure of Λ_X block Toeplitz:

$$\Lambda_X = \begin{bmatrix} \Lambda_0 & \Lambda_1 & \ldots & \Lambda_{N-1} \\ \Lambda_1 & \Lambda_0 & \ldots & \Lambda_{N-2} \\ \vdots & \vdots & & \vdots \\ \Lambda_{N-1} & \Lambda_{N-2} & \ldots & \Lambda_0 \end{bmatrix}. \tag{7}$$

We can further simplify the covariance model if we assume the process has compact support. For example, if $\Lambda_k = 0$ for $k > 0$ then Equation 7 is block diagonal. For this trivial case, the whitening filter, $\Lambda_X^{-1/2}$ is also block diagonal,

$$\Lambda_X^{-1/2} = \begin{bmatrix} \Lambda_0^{-1/2} & & & \\ & \Lambda_0^{-1/2} & & \\ & & \ddots & \\ & & & \Lambda_0^{-1/2} \end{bmatrix}. \tag{8}$$

This makes the whitened vectors $w_i = \Lambda_0^{-1/2} x_i$, and we identify the V matrix in Equation 2 as $\Lambda_0^{-1/2}$, which clarifies why the expression in Equation 6 does not need to explicitly represent it.

A less trivial case is when $\Lambda_k = 0$ for $k > 1$, making Λ_X block tridiagonal,

$$\Lambda_X = \begin{bmatrix} \Lambda_\phi & \Lambda_1 & & & & \\ \Lambda_1 & \Lambda_0 & \Lambda_1 & & & \\ & \Lambda_1 & \Lambda_0 & \Lambda_1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \Lambda_1 & \Lambda_0 & \Lambda_1 \\ & & & & \Lambda_1 & \Lambda_0 \end{bmatrix}. \tag{9}$$

As Λ_X is symmetric positive semi-definite, it can be factored with a block Cholesky decomposition, $\Lambda_X = LL^T$ (Golub and Van Loan, 2013). Note we have introduced Λ_ϕ into the covariance model, so Λ_X deviates from being strictly block Toeplitz, but this trick simplifies L to block bidiagonal,

$$L = \begin{bmatrix} L_0 & & & & & \\ L_1 & L_0 & & & & \\ & L_1 & L_0 & & & \\ & & \ddots & \ddots & & \\ & & & L_1 & L_0 & \\ & & & & L_1 & L_0 \end{bmatrix}. \tag{10}$$

Similar to Equation 4, we can verify that $W = L^{-1}X$ is white. Thanks to its structure, the inverse of L can be efficiently computed using block Gaussian elimination (Golub and Van Loan, 2013). The solution, as illustrated graphically in Figure 1, results in the following recursion:

$$\begin{aligned} w_0 &= L_0^{-1} x_0 \\ w_1 &= L_0^{-1}(x_1 - L_1 w_0) \\ &\vdots \\ w_{N-1} &= L_0^{-1}(x_{N-1} - L_1 w_{N-2}). \end{aligned} \tag{11}$$

The elements of the matrices L_0^{-1} and L_1 are learned directly as part of training the attention model, and the matrix L_0 need never be inverted.

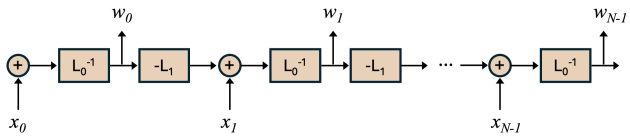

Figure 1: The whitening filter for the block tridiagonal covariance matrix is a recursion taking the correlated x_i as inputs and producing independent w_i at the output.

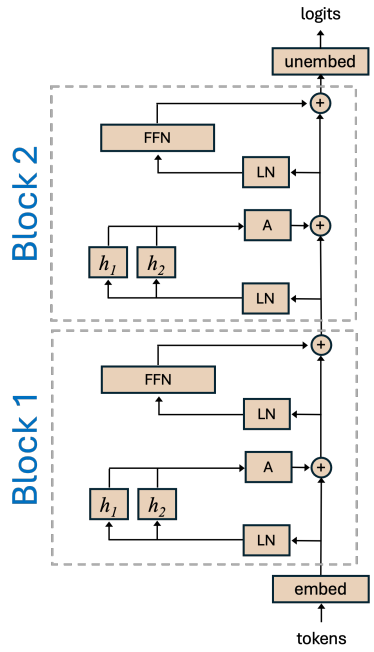

Figure 2: Decoder Transformer architecture with two blocks, each with a two-head (h) attention and a feed-forward layer (FFN). Each layer includes a layer norm (LN), and a projection (A). Tokens are converted to embeddings at the input, and to logits at the output.

4 Experiments

To test the efficacy of our model, we implement whitened attention using Equations 6 and 11 and compare the results to experiments using standard attention, as specified in Equation 2. The experiments are based on the small Transformer architecture shown in Figure 2, consisting of two Transformer blocks, each with two attention heads. Positional information is incorporated using rotary positional embeddings (RoPE) (Su et al., 2024).

The data used in the experiments is the collected works of Charles Dickens, obtained from the Project Gutenberg website (Dickens, 2018), and we used a character-based tokenization strategy (Banar et al., 2020). The corpus contains 13m characters, with a total vocabulary of 93 unique tokens. This approach significantly simplifies the language preprocessing required when training language models with Transformers. It has the advantage of providing a small, well-represented vocabulary without having to map rare tokens to a catchall such as <UNK>. The model implements the standard Transformer blocks: layer norms, linear projections, multilayer perceptrons, and the embedding and unembedding layers.[1] Our experiments are implemented with a context window of length 256 and a token embedding dimension of 256. We use the mean cross-entropy (MCE) loss applied to the validation data as our performance metric.

To evaluate whitened attention, we ran four experiments, each for 100k iterations. The first was the standard GPT implementation of self-attention with RoPE. The batch parameter was 256 and the model size worked out to 1.6m parameters. Its MCE loss on the validation data is represented by the blue curve in Figure 3. The final value of loss for this experiment was 1.39, corresponding to a perplexity of 4.0 versus 93 for the untrained model.

The green curve in the figure is the result for our whitened attention formulation. The batch parameter for this experiment was 256, and the model size 2.15m parameters. As indicated in the figure, it rapidly outstrips standard attention, dropping to the same MCE loss in just 5,784 iterations (17x less). The loss value continues to improve, achieving a value of 1.24 after 100k iterations. This corresponds to a perplexity of 3.47, a better than 13% improvement over the result for standard attention.

As the whitened attention experiment benefited from more parameters, a third experiment was run for standard attention, but with an equivalent model capacity of 2.15m parameters, achieved by increasing the embedding dimension from 256 to 296. Again the batch size was 256. The results are represented by the orange curve in the figure. The final MCE loss for this trial was 1.36, a slight improvement over the smaller standard attention experiment. By comparison, the whitened attention run achieves the same level of loss in 7,593 iterations (13x less), while delivering an 11% improvement in perplexity at 100k iterations.

These results are summarized in the first three rows of Table 1, which also provide the compute time for the full 100k iterations.[2] The whitened attention model required 10.4x more time than was needed for standard attention with equivalent ca-

[1] See https://transformer-circuits.pub/2021/framework/

[2] All computations were performed on an Nvidia RTX 4090

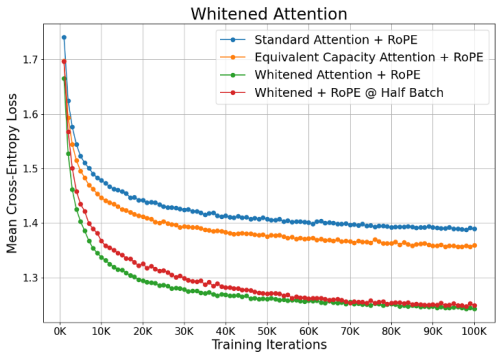

Figure 3: Comparing whitened and standard attention of GPT training on the validation data (10% split).

pacity, but as discussed, it achieves the same level of MCE loss in 13x fewer iterations, corresponding to 21% less time (see Table 2). This motivated a fourth experiment, rerunning the whitened model with half the batch size, 128. The results are represented by the red curve in the figure, which shows a final MCE loss almost identical to the one with the larger batch size. However, as shown in Table 1, in comparison to full batch whitened attention, it completed in 51% of the compute time. It achieves the same level of MCE loss as the equivalent capacity attention at iteration 10,879 and in 42% less time (see Table 2). Equivalent attention performance times are recapped in Table 2. These experiments demonstrate that whitening is a powerful technique that significantly improves the model's ability to explain the data while reducing the compute time for an equivalent MCE loss.

Experiment	Model Size	MCE Loss	Compute Time	Perplexity
Standard Attention	1.6m	1.39	56 min	4.00
Equivalent Capacity	2.15m	1.36	73 min	3.90
Whitened Attention	2.15m	1.24	761 min	3.47
Whitened Half Batch	2.15m	1.25	387 min	3.49

Table 1: Performance summary, comparing standard and whitened attention models for 100k iterations. Equivalent capacity refers to the larger standard attention trial.

5 Conclusions and Future Research

In this paper, we show that whitening improves the performance of self attention for a small GPT architecture. It delivered 11% improvement in perplexity and achieved standard attention's best result in

	Batch Size	Iters	Compute Time	%Time
ECSA	256	100k	73 min	100%
WA	256	7,592	58 min	79%
WAHB	128	10,879	42 min	58%

Table 2: Number of iterations and time to attain an MCE loss of 1.36: ECSA is for Equivalent Capacity Standard Attention and is the baseline, WA is for Whitened Attention, and WAHB is for Whitened Attention Half Batch. Each model was trained with 2.15m weights.

42% less time. If this carries over to training larger LLM models it would mean savings in compute time, improvements in performance, or a combination of both. Furthermore, as whitening removes inter-symbol correlations, it will likely affect the weights of attention heads and MLPs, which in turn, will have a knock-on effect on results reported by papers on mechanistic interpretability (Bereska and Gavves, 2024; Frankle and Carbin, 2018; Naim and Asher, 2024; Scherlis et al., 2022), making this an important topic for future investigation.

Our next steps will focus on scaling to larger corpora, using more sophisticated tokenization strategies, and implementing larger GPT models. We saw that a simple change in batch dimension significantly reduced compute time, so additional hyperparameter optimization is one direction of subsequent investigation. Our developments also highlight the potential of covariance modeling, and we plan to explore additional ideas at both the global and block levels. For covariance blocks, matrix series truncations such as Neumann series expansions and Krylov subspaces (Strang, 2000) could help reduce computational burden, as could diagonal plus low rank matrices (Saunderson et al., 2012). At the global level, we plan to extend the block tridiagonal model in Equation 9 to higher orders, such as the pentadiagonal case ($\Lambda_k = 0$ for $k > 2$). Finally, for longer sequences, we plan on implementing the convolutional recursion in Equation 11 with an FFT.

6 Limitations

The main limitation of our approach is the computational burden introduced by recursion needed for the implementation of whitening. This interferes with a fully parallelized implementation on a GPU, however, it may be possible to mitigate this by leveraging CUDA optimizations such as tiling, memory coalescing, kernel fusion, and so on (Hijma et al., 2023; Yang et al., 2011).

Acknowledgments

Removed for anonymity

References

Hirotugu Akaike. 1969. Fitting autoregreesive models for prediction. In *Selected Papers of Hirotugu Akaike*, pages 131–135. Springer.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Nikolay Banar, Walter Daelemans, and Mike Kestemont. 2020. Character-level transformer-based neural machine translation. In *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, pages 149–156, New York, NY, USA. Association for Computing Machinery.

Leonard Bereska and Stratis Gavves. 2024. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*.

Charles Dickens. 2018. Index of the project gutenberg works of Charles Dickens. https://www.gutenberg.org/ebooks/58157. Public domain. Accessed: 2025-05-15.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Gene H Golub and Charles F Van Loan. 2013. *Matrix Computations*. JHU press.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and 1 others. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Proc. Interspeech 2020*, pages 5036–5040.

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and Dacheng Tao. 2022. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):87–110.

Pieter Hijma, Stijn Heldens, Alessio Sclocco, Ben Van Werkhoven, and Henri E Bal. 2023. Optimization techniques for GPU programming. *ACM Computing Surveys*, 55(11):1–81.

Thomas Kailath. 1970. The innovations approach to detection and estimation theory. *Proceedings of the IEEE*, 58(5):680–695.

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2021. AMMUS: A survey of transformer-based pretrained models in natural language processing. *Language*, 4.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 54(10s):1–41.

Beat Kleiner, R Douglas Martin, and David J Thomson. 1979. Robust estimation of power spectra. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 41(3):313–338.

Omar Naim and Nicholas Asher. 2024. On explaining with attention matrices. In *ECAI 2024*, pages 1035–1042. IOS Press.

Athanasios Papoulis and S Unnikrishna Pillai. 2002. *Probability*. McGraw-Hill.

James Saunderson, Venkat Chandrasekaran, Pablo A Parrilo, and Alan S Willsky. 2012. Diagonal and low-rank matrix decompositions, correlation matrices, and ellipsoid fitting. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1395–1416.

Adam Scherlis, Kshitij Sachan, Adam S Jermyn, Joe Benton, and Buck Shlegeris. 2022. Polysemanticity and capacity in neural networks. *CoRR*.

Juliet Popper Shaffer. 1991. The Gauss-Markov theorem and random regressors. *The American Statistician*, 45(4):269–273.

Gilbert Strang. 2000. *Linear Algebra and its Applications*. Academic Press.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Harry L Van Trees. 2004. *Detection, Estimation, and Modulation Theory, Part I: Detection, Estimation, and Linear Modulation Theory*. John Wiley & Sons.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Xintian Yang, Srinivasan Parthasarathy, and P Sadayappan. 2011. Fast sparse matrix-vector multiplication on gpus. *Proceedings of the VLDB Endowment*, 4(4):231–242.

Shuang Zhang, Rui Fan, Yuti Liu, Shuang Chen, Qiao Liu, and Wanwen Zeng. 2023. Applications of transformer-based language models in bioinformatics: A survey. *Bioinformatics Advances*, 3(1):vbad001.