Eigen Analysis of Self-Attention and its Reconstruction from Partial Computation

Srinadh Bhojanapalli, Ayan Chakrabarti, Himanshu Jain, Sanjiv Kumar, Michal Lukasik, Andreas Veit

Google Research, New York

Abstract

State-of-the-art transformer models use pairwise dot-product based self-attention, which comes at a computational cost quadratic in the input sequence length. In this paper, we investigate the global structure of attention scores computed using this dot product mechanism on a typical distribution of inputs, and study the principal components of their variation. Through eigen analysis of full attention score matrices, as well as of their individual rows, we find that most of the variation among attention scores lie in a low-dimensional eigenspace. Moreover, we find significant overlap between these eigenspaces for different layers and even different transformer models. Based on this, we propose to compute scores only for a partial subset of token pairs, and use them to estimate scores for the remaining pairs. Beyond investigating the accuracy of reconstructing attention scores themselves, we investigate training transformer models that employ these approximations, and analyze the effect on overall accuracy. Our analysis and the proposed method provide insights into how to balance the benefits of exact pair-wise attention and its significant computational expense.

1 Introduction

Transformers deliver state-of-the-art performance in several tasks in natural language processing [20, 6, 12–14], and are beginning to show promise in other domains [7]. Their success is attributed to their self-attention mechanism. While previous architectures based on convolutional or recurrent layers had a-priori fixed structures for interaction between different positions in the input, self-attention in transformers allows arbitrary input-dependent interaction through attention scores based on dot product similarities between learned hidden feature representations. The importance of this flexible input-dependent attention has been established both empirically [24, 15, 18] and theoretically [25].

While transformers have the expressive capacity to output arbitrary attention scores between all the input token pairs, these scores are computed based on the inputs to these models. In the case of natural language, we know that these inputs are structured and have a more restricted distribution than random sequences of words [2]. It follows, then, that attention scores computed from inputs drawn from a structured natural distribution will themselves exhibit some structure. In this paper, we analyze the distribution of attention scores computed by transformer models from natural language.

Our analysis shows that the attention scores—either the global attention matrix of all scores across all pairs of tokens, or scores corresponding to a given query token position—are not arbitrary and much of their variability can be explained by a relatively small number of principal components. Moreover, surprisingly, we find these components to be reasonably shared by different layers and heads in a single model, by different models, and across different input distributions (of the same language—English). Based on these insights, we investigate how this structure could be used to reconstruct the

^{*}Authors are ordered alphabetically. Corresponding email: bsrinadh@google.com

full set of attention scores without computing all of them explicitly. We propose an approach for selecting a partial set of scores for exact computation, and for reconstructing all scores from this partial set. Our experiments using this approximate computation approach within transformer models show an encouraging trade-off between network accuracy and attention computation cost.

In summary, our contributions in this paper are as follows:

- Through eigen analysis of attention scores of different variants of BERT transformer models [6], we establish that these scores lie in low-dimensional subspaces. We show that, surprisingly, these subspaces are largely shared across different layers, models, and datasets.
- We propose an approximate attention computation approach that selects a subset of token pairs to compute attention scores exactly, and estimates the remaining from these computed values.
- We show that this partial computation approach yields attention score estimates with low mean squared error, and conduct preliminary experiments to train networks featuring these approximations.

1.1 Background

Transformers Each transformer block comprises two components: 1) a multi-head self-attention block; and 2) a token-wise feed-forward multi-layer perceptron (MLP). The input to these blocks is a sequence of vectors $\boldsymbol{X} \in \mathbb{R}^{f \times n}$, where n is the sequence length, and the columns of \boldsymbol{X} represent the f-dimensional embedding of different tokens. The self-attention layer updates these embeddings by a linear combination of values using the pairwise dot product similarities of per token query and key vectors, where values, queries, and keys are all computed by linear transforms applied to \boldsymbol{X} . The attention scores $\boldsymbol{A}_{\boldsymbol{X}}$ are computed as:

$$\boldsymbol{A}_{\boldsymbol{X}} = \boldsymbol{X}^{\top} \boldsymbol{W}_{O}^{\top} \boldsymbol{W}_{K} \boldsymbol{X} / \sqrt{d}, \tag{1}$$

where W are trainable parameter matrices, and W_QX and $W_KX \in \mathbb{R}^{d \times n}$ denote the d- dimensional query and key projections. These attention scores are used to linearly combine inputs as follows: $Z = \sigma\left(A_X\right) \cdot X^\top W_V^\top \cdot W_0$, where W_V denotes the value projection, and σ is a row-wise softmax operator. Multi-head attention involves multiple such trainable attention heads in a single layer—using dimensionality d of the queries, keys, and values in each head, being equal and summing up to f. The output of the attention block is fed into a tokenwise feedforward layer: $W_2 \Phi\left(W_1Z^\top\right)$, with Φ denoting a non-linear activation. Both the self-attention and MLP blocks employ layer-normalization and residual connections.

Related Work Given the popularity of transformers, there have been many works on understanding their behavior in natural language tasks. Clark et al. [5], Hewitt and Manning [9], Vig et al. [21], Michel et al. [11] used several language tasks as probes to understand how the language representation evolves over the layers of transformer models. They demonstrated that different heads specialize in particular linguistic sub-tasks such as parts of speech tagging. These works focused on the ability of attention to capture linguistic knowledge. We encourage the reader to see Rogers et al. [16] for an excellent overview of such analyses.

Speeding up attention computation in transformers using different approximations has also been an important research direction. One popular approach is to perform a sparse computation of attention scores. Several works have explored different sparsity patterns and shown their utility for long sequence lengths [3, 1, 8, 26]. Others have explored using clustering, hashing, etc. to group tokens and compute attention based on these groupings [10, 17]. [4] explored using a low rank and kernel approximation of attention scores. We refer to Tay et al. [19] for a more detailed discussion of these approaches. However, these works are not based on analysis of the distribution of attention scores on inputs from natural datasets. In contrast, our work is primarily focused on analyzing the variation of attention on real-world datasets, which can serve as a starting point for approximation approaches (including the one we propose).

Recently, Raganato et al. [15] explored using fixed attention patterns with different patterns in each head, and one learnable attention head per layer. Tay et al. [18] propose using a learnable, but input independent, attention matrix and evaluate such models on language tasks. While these works are similar in spirit they do not analyse attention scores learned by transformers, as we do in this paper. Moreover we keep attention computation input dependent, and exploit the low dimensional structure for reconstruction from partial computation.

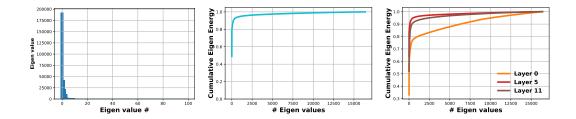


Figure 1: **Eigen values of** C_a . Left: Barplot of the top 100 eigen values of the covariance matrix C_a of attention scores aggregated over the entire network of a BERT_{BASE} model. Middle: Cumulative sum of eigen values of C_a . Both show that C_a is approximately low rank with top 200 (1.2%) eigen values capturing > 90% of the energy. Right: Cumulative sum of eigen values of attention scores covariance matrix C_a^l for different layers of a BERT_{BASE} model. We notice later layers in the network have smaller rank.

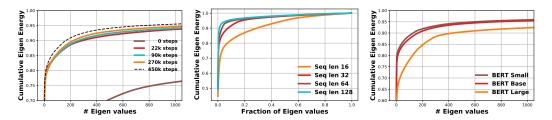


Figure 2: **Eigen values of** C_a . Cumulative sum of eigen values of attention scores covariance matrix C_a of a BERT_{BASE} model - Left: after varying number of training steps. We notice that the rank slightly decreases throughout training with a large reduction in the beginning. Middle: for different sequence length inputs. Note that x-axis here denotes the fraction of eigen values. Right: for varying model sizes. Note that rank slightly increases with model size.

2 Eigen Analysis of the Attention Scores

In this section we present our analysis of attention scores on a pre-trained BERT_{BASE} model [6]². This model has 12 transformer layers and is pre-trained using a Masked Language Modeling (MLM) task on English Wikipedia and Books datasets [27]. We follow the same setting as in Devlin et al. [6] and use their codebase³. We use an input sequence length of 128 for our experiments⁴. We refer to Appendix for our detailed experimental setup.

Our goal is to analyse the principal components of the subspace that captures the variation of attention scores. Towards this we first compute the covariance matrix of the attention scores for the entire network across all inputs. Let $a_{\boldsymbol{X}}^{l,h} = \operatorname{Vectorize}(\boldsymbol{A}_{\boldsymbol{X}}^{l,h})$ be the vectorized form of attention scores from layer l and head h for a given input \boldsymbol{X} . For a given input sequence length of n, $a_{\boldsymbol{X}}^{l,h} \in \mathbb{R}^{n^2}$. The covariance matrix of the attention scores is given below⁵.

$$C_a = \mathbb{E}_{\boldsymbol{X}} \left[\frac{1}{L \cdot H} \sum_{l \in [L], h \in [H]} a_{\boldsymbol{X}}^{l,h} (a_{\boldsymbol{X}}^{l,h})^{\top} \right].$$
 (2)

Here L denotes the number of layers and H the number of heads per layer. For a given sequence length n, C_a is a $n^2 \times n^2$ dimensional matrix. In practice, we estimate C_a by computing an empirical

²Please see Appendix for a similar analysis of the BERT_{LARGE} model.

³https://github.com/google-research/bert.

⁴Note that BERT uses a sequence length of 512. However the attention scores covariance matrix in that case has $512^2 \times 512^2 \approx 68B$ entries making it challenging to compute in practice.

⁵Note that we use the outer-product matrix without mean subtraction, since we want to analyze the span of the attention scores themselves.

average over all the training examples. We compute this for the Wikipedia dataset by averaging over 2,500M words.

Let the eigen decomposition of C_a be $\sum_i \lambda_i v_i v_i^{\top}$, $v_i \in \mathbb{R}^{n^2}$. Eigenvalues capture the variation of the attention scores distribution along different principal components. We plot the top 100 eigenvalues of this matrix in Fig. 1. We also plot the cumulative sum of eigenvalues $(\sum_{i=1}^k \lambda_i / \sum \lambda_i)$, referred to as cumulative eigen energy in the middle of Fig. 1. We first observe that attention scores lie in an approximately low rank subspace with few eigenvalues dominating over the rest. Even though attention scores are represented in a 16384 dimensional space, top 200 (1.2%) eigenvalues capture > 90% of the total energy. Note that this is different from the rank of the attention scores matrix $(\operatorname{rank}(A_X))$, and instead captures the subspace dimension of attention scores across different inputs.

One may wonder if this low dimensional nature of attention scores is due to some constraint in the transformer architecture. We note that transformers have $2 \cdot n \cdot d$ degrees of freedom in the query and key projections used for attention computation per layer. For BERT_{BASE} model this translates to $2 \cdot 128 \cdot 768$ degrees of freedom, much larger than the low rank we observed in Fig. 1. Additionally we will see that at initialization the spectrum is quite flat, and model learns this low dimensional structure during training.

Individual layers We next look at covariance matrix of attention scores from individual layers.

$$\boldsymbol{C}_{a}^{l} = \mathbb{E}_{\boldsymbol{X}} \left[\frac{1}{H} \sum_{\boldsymbol{X} \in \mathcal{X}, h \in [H]} a_{\boldsymbol{X}}^{l,h} (a_{\boldsymbol{X}}^{l,h})^{\top} \right].$$
 (3)

Note that the global covariance matrix (C_a) is the mean of individual layer covariances (C_a^l) across all layers. We plot the cumulative eigen spectrum for layers 0, 5 and 11 of a BERT_{BASE} model in Fig. 1. Again we notice a similar low rank⁶ structure even in the individual layer covariance matrices. However we do notice difference between layers, with earlier layers having a flatter spectrum than later layers. This can potentially be attributed to earlier layers of the model being more input sensitive than the later layers.

Training steps We next plot the evolution of the eigen spectrum at different steps during the training in Fig. 2. We first notice that initially the spectrum is quite flat with top 100 eigenvalues capturing only 35% of the total energy. It however quickly concentrates in 1/4th of the total training steps, leading finally to an approximate low rank structure with almost 90% energy captured by top 200 eigenvalues.

Sequence length We next study the effect of the input sequence length on the eigen values of the covariance matrix. We consider four different values of input sequence lengths and plot the eigen spectrum in Fig. 2. We notice that as we increase the sequence length the rank of the covariance matrix remains relatively small even though the dimension increases quadratically.

Model size To study the effect of model size on the eigen spectrum we consider two additional models $BERT_{SMALL}$ and $BERT_{LARGE}$ with 6 and 24 layers respectively. We plot their eigenspectrum in Fig. 2. We notice that though the rank of the covariance matrix increases with model size, it is still relatively small. Even for a large model ($BERT_{LARGE}$) top 200 eigen values capture greater than 85% of the total energy.

Dataset Finally we compute attention score distributions from different datasets and plot their eigen spectrum in Fig. 4. We notice that the same low rank behavior holds across different datasets.

2.1 Subspace Similarity

We next study the similarity between the principal components of the global and different layer covariance matrices. To measure this we project different layer covariance matrices onto the eigenspace of the global covariance matrix. Let $V \in \mathbb{R}^{n^2 \times k}$ be the projection matrix onto a k dimensional

⁶Note that we will drop the "approximate" qualifier in the remainder of the paper and simply refer to such matrices as low rank.

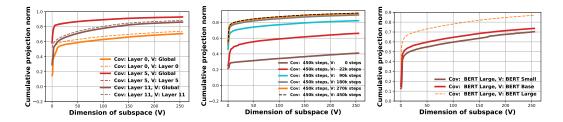


Figure 3: **Subspace similarity**. We plot the cumulative projection norm (eq 4) to measure subspace overlap for different choices of attention scores C_a and subspaces V for varying subspace dimension. Left: We project covariance matrices (C_a^l) of different layers projected onto the top 256 eigen vectors of global covariance matrix C_a of a BERT_{BASE} model. We notice that there is substantial overlap in eigen subspaces of global and per layer attention scores. Middle: We project attention scores covariance matrix after full training onto subspaces of attention scores after different number of training steps. We notice that the overlap increases quickly as training progresses. Right: Covariance matrix of a BERT_{LARGE} model projected onto eigen spaces of models with varying sizes.

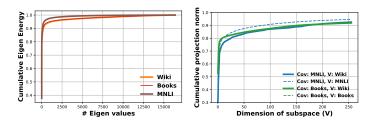


Figure 4: **Variation across datasets**. Left: We plot the cumulative eigen spectrum of the attention scores variation on different datasets, and notice similar low dimensional structure. Right: Similarity between subspaces (eq 4) of attention scores from different datasets, for varying subspace dimension. We notice a large subspace overlap across different datasets.

subspace. Then for a given attention score a_{X} , we are interested in the following projection norm.

$$\mathbb{E}_{\boldsymbol{X}}\left[\frac{1}{L\cdot H}\sum_{l,h}\|\boldsymbol{V}^{\top}\boldsymbol{a}_{\boldsymbol{X}}^{l,h}\|^{2}\right] = \mathbb{E}_{\boldsymbol{X}}\left[\frac{1}{L\cdot H}\sum_{l,h}\operatorname{Tr}(\boldsymbol{V}^{\top}\boldsymbol{a}_{\boldsymbol{X}}^{l,h}(\boldsymbol{a}_{\boldsymbol{X}}^{l,h})^{\top}\boldsymbol{V})\right] = \operatorname{Tr}(\boldsymbol{V}^{\top}\boldsymbol{C}_{a}\boldsymbol{V}). \quad (4)$$

Here Tr computes the trace of a matrix. Note that if V spans the top-k eigenspace of C_a , then this is exactly the sum of its top-k eigen values. The above projection measures how much the principal components V capture the variation in attention scores C_a .

In Fig. 3 we plot this projection norm of the per-layer covariance matrices (C_a^l) of a BERT_{BASE} model with the top-k eigen space of C_a referred to as Global in the plot. We also plot the exact eigenspectrum of C_a^l for comparison. We notice that projection onto the eigen space of C_a preserves most of the eigen spectrum of C_a^l , showing that their principal components are very similar. We repeat this analysis using the covariance matrices computed at different steps of the training, and notice in Fig. 3 that by 1/3rd into training, the eigenspace is highly similar to that of the fully trained model. We also compare eigenspace across different sized models and notice that there is a substantial overlap, with top 250 eigen vectors of BERT_{BASE} capturing more than 70% of energy of a BERT_{LARGE} model. Finally we compare the subspace similarity of attention scores from different datasets in Fig. 4 and notice they have high overlap as well.

2.2 Per-Query Attention Scores

So far we have analysed the attention scores computed for the entire input. However in transformers, attention is computed independently for each token/query in the input. We now study the eigen spectrum of per-query attention scores which are the rows of the attention scores matrix.

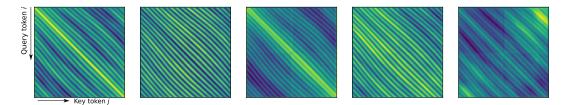


Figure 5: Global principal components. Visualization of attention score patterns captured by the top 5 eigenvectors of C_a of a BERT_{BASE} model. We notice that the leading principal components capture predominantly shifted diagonal patterns.

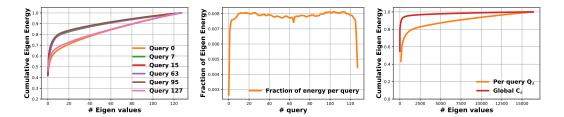


Figure 6: **Per-query eigen spectrum.** Left: Cumulative eigen values sum for different queries. We notice that first and last queries, which are special tokens in BERT, have consistently higher rank compared to rest. Middle: Fraction of total eigenvalue sum for each query. We again notice that the special tokens have lower energy. Right: for each $k=i\times 128$, we take top i eigenvalues of each per-query covariance matrix (Q_a) , and plot their cumulative eigen spectrum. This is contrasted against the eigen spectrum of the global attention scores (C_a) . We notice that per-query attention scores capture majority of the variation in the global attention scores.



Figure 7: **Per-query principal components**. Visualization of per-query attention score patterns from the top 5 eigenvectors (1st for the leftmost Figure, then 2nd etc.) from each query stacked into their corresponding rows. Note the similarities between the top eigenvectors of the per-query attention scores and global attention scores (Fig 5).

We first define the per-query attention scores covariance matrices as follows. Let $A_{\boldsymbol{X}}^{l,h}[i,:]$ denote the ith row of $A_{\boldsymbol{X}}^{l,h}$. Then $a_{\boldsymbol{X}}^{l,h,i}=A_{\boldsymbol{X}}^{l,h}[i,:]$ is the ith query attention score from layer l and head h for a given input \boldsymbol{X} . For a give input sequence length of n, $a_{\boldsymbol{X}}^{l,h,i}\in\mathbb{R}^n$. The covariance matrix of the per-query attention scores, a $n\times n$ matrix, for row $i\in[n]$ is given below.

$$\boldsymbol{Q}_{a}^{i} = \mathbb{E}_{\boldsymbol{X}} \left[\frac{1}{L \cdot H} \sum_{l,h} a_{\boldsymbol{X}}^{l,h,i} (a_{\boldsymbol{X}}^{l,h,i})^{\top} . \right]$$
 (5)

We plot the eigenvalues of this matrix in Fig. 6 for different input queries/rows, and the fraction of total eigenvalue sum for different queries. We notice that special tokens 0 and 127, which correspond to [CLS] and [SEP] tokens in BERT, have higher rank and lower fraction of energy compared to other tokens. We notice that the eigenspectrum of most other queries is similar with them being predominantly concentrated in first eigenvalue.

Note that Q_a^i are block diagonals of C_a , and do not capture cross query variation. To compare how much the variation in individual rows captures the global variation of attention scores we plot the cumulative eigen energy of global attention scores C_a and of the per token attention scores aggregated across rows. We notice that per token eigenspectrum captures most of the variation of attention scores.

Finally we visualize the top 5 eigenvectors of each query stacked across rows in Fig. 7. Note the similarity with the corresponding eigen vectors from the global attentions scores in Fig. 5.

3 Reconstructing Attention from Partial Computation

In the last section, we saw that attention scores tend to lie in low-dimensional subspaces. This implies that the attention score matrix, for a given input at a given self-attention layer, can be represented with fewer coefficients than the total number of elements in that matrix. We now investigate whether this property can be used to compute attention scores efficiently, without explicit computation of inner products between all query-key pairs in the input sequence.

3.1 Formulation

Let $a \in \mathbb{R}^{l \times l}$ denote a vector of attention scores and $C \in \mathbb{R}^{l \times l}$ its corresponding covariance matrix. Here, a corresponds to either a single row of the attention matrix or to its flattened version, with l being number of tokens n or its square n^2 in each case. We seek to explicitly compute the values of only a partial subset of $k \ll l$ elements of a with query-key inner products, and then reconstruct the remaining elements from these computed values.

Accordingly, we let $P\subset\{1,2,\dots l\}, |P|=k$ denote the indices of the elements of a to be computed exactly, and $\overline{P}=\{1,2,\dots l\}\setminus P$ the set of remaining indices. Moreover, we let $a_P\in\mathbb{R}^{k\times 1}$ and $a_{\overline{P}}\in\mathbb{R}^{(l-k)\times 1}$ denote vectors containing the corresponding elements of a. Thus, after explicitly computing a_P , we seek to compute an estimate $\hat{a}_{\overline{P}}$ of the remaining attention scores $a_{\overline{P}}$ from a_P . We will do so using a linear transform $\mathbf{R}\in\mathbb{R}^{(l-k)\times k}$ as $\hat{a}_{\overline{P}}=\mathbf{R}\,a_P$.

Optimal Reconstruction Given a choice of P, the average of the squared error $||a_{\overline{P}} - R a_P||^2$ in the estimates $\hat{a}_{\overline{P}}$ is given by trace of the matrix $C_{\overline{P}|P}$, which is defined as⁷

$$\boldsymbol{C}_{\overline{P}|P} = \mathbb{E}_{\boldsymbol{X}} (a_{\overline{P}} - \boldsymbol{R} a_{P}) (a_{\overline{P}} - \boldsymbol{R} a_{P})^{\top} = \boldsymbol{C}_{\overline{PP}} - \boldsymbol{R} \boldsymbol{C}_{P\overline{P}} - \boldsymbol{C}_{\overline{P}P} \boldsymbol{R}^{\top} + \boldsymbol{R} \boldsymbol{C}_{PP} \boldsymbol{R}^{\top}.$$
 (6)

Here, C_{AB} denotes a "crop" of the covariance matrix C containing the rows and columns with indices in sets A and B. It is easy to see that the trace of $C_{\overline{P}|P}$ is minimized by setting R as

$$R = C_{\overline{P}P}C_{PP}^{-1},\tag{7}$$

and the expression for $C_{\overline{P}|P}$ simplifies to the Schur's complement of C_{PP} in C, i.e.,

$$C_{\overline{P}|P} = C_{\overline{PP}} - C_{\overline{P}P}C_{PP}^{-1}C_{P\overline{P}}.$$
(8)

Selecting Partial Set Our choice of the partial set of indices P, for a given choice of its size k, should be such that it yields a low reconstruction error (from (8)). Unfortunately, finding the globally optimal choice of P would require evaluating all possible subsets of size k, with computational cost $O(\exp(k))$. But, we find a greedy selection approach as described below to work well in practice.

We form a series of matrices $P^1, P^2, \dots P^k$, with $|P^{k'}| = k'$ and $P^{k'} \supset P^{k'-1}$ (with P^0 the empty set), and then set $P = P^k$ for our desired choice of k. Given $P^{k'}$ and the corresponding residual covariance matrix $C^{k'} = C_{\overline{P}^{k'}|P^{k'}}$, we set $P^{k'+1} = P^{k'} \cup \{i\}$, choosing $i \in \overline{P}^{k'}$ as

$$i = \arg\min_{i} \text{Tr}(\mathbf{C}^{k'+1}) = \arg\min_{i} \sum_{j \in \overline{P}_{k'} \setminus \{i\}} \mathbf{C}_{jj}^{k'} - \frac{(\mathbf{C}_{ij}^{k'})^{2}}{\mathbf{C}_{ii}^{k'}} = \arg\max_{i} \frac{\sum_{j \in \overline{P}_{k'}} (\mathbf{C}_{ij}^{k'})^{2}}{\mathbf{C}_{ii}^{k'}}. \quad (9)$$

Computational Cost The computational savings of this approach will depend on whether it is applied to the whole attention matrix, or to each row independently. We let \bar{k} denote the total number of scores we compute exactly, with $\bar{k}=nk$ and k for the per-query and whole matrix settings. Then, for d-dimensional query and key vectors, the combined computational cost of exact computation

⁷Note that these expressions correspond to to the conditionals of multivariate zero-mean Gaussian distributions with covariance C.

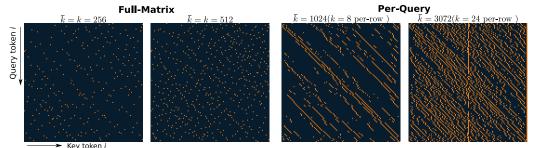


Figure 8: Pairs Selected for Partial Computation. We visualize examples of the partial sets P of query-token pairs selected by our greedy algorithm (eq. (9)), given total number of pairs \bar{k} . We show examples of both whole matrix and per-query sets, with $k = \bar{k}/n$ pairs for each row for the latter.

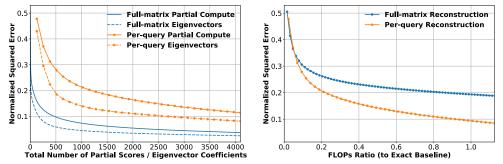


Figure 9: **Reconstruction Errors from Partial Computation**. We show average squared error of attention scores—normalized by their total variance—reconstructed using our partial computation approach. (Left) We plot these for both whole matrix and per-query reconstruction, as a function of the total number \bar{k} of pairs computed exactly. We compare these to reconstruction from projections on to an equivalent number of dense eigenvectors. (Right) We analyze the accuracy-computation trade-off in whole matrix vs. per-query reconstruction, but plotting errors as a function of floating point operations needed for partial computation and reconstruction, to those required for full exact computation (FLOPs ratio).

and reconstruction using our approach is $O(\bar{k}d+\bar{k}n)$ and $O(\bar{k}d+\bar{k}n^2)$ in the per-query and whole matrix settings respectively. In contrast, full exact computation has a cost of $O(n^2d)$. Thus, although the whole matrix setting may yield better reconstructions by exploiting correlations across rows for the same number of partial computations \bar{k} , it also entails a higher computational cost. As we see next, the per-row setting yields a better trade-off between accuracy and computational cost.

3.2 Reconstruction Error

We begin by evaluating the partial pairs P selected by our method for exact computation, and the corresponding average squared reconstruction errors, for attention scores from a typical network. Note that the reconstruction errors can be computed directly (using (8)) from the covariance matrices estimated in Sec. 2. We report results for the BERT_{BASE} model trained on the standard pre-training task of masked language modeling (MLM) [6], using average covariance across layers and heads.

In Fig. 8, we show the query-token pairs selected by our greedy algorithm for a few choices of k—for both the whole matrix and per-query settings. We notice that, like the eigenvectors in Sec.2, these patterns often cluster along shifted diagonals. This effect is more pronounced in the per-query patterns, where each row can rely only on its own exact computations, while those for the whole matrix are less coherent. We next characterize the reconstruction error in both settings in Fig. 9. First, we plot reconstruction errors, normalized by total variance of the full covariance matrix, for a range of values for number of exact scores \bar{k} (equal to k for whole matrix patterns, and nk for per-query). For reference, we compare these to error from approximation by the same number of top whole matrix and per-query eigenvectors. As expected, the constraint on sampling a subset of entries rather than projecting to dense eigenvectors leads to a gap in error. Nevertheless, we find that our approach yields reasonable reconstructions with increasing numbers of coefficients. When comparing with an equivalent number of exact computations, using whole matrix reconstruction yields better results.

Table 1: **Network Performance with Partial Computation**. We train BERT models with partial attention computation and per-row restoration, for different values of per-row exact scores k. We report accuracies of models trained for MLM and fine-tuned for MNLI (averaging three runs for the latter), and FLOPs ratios for attention computation. We select the optimal set P and initialize R as per (7), and consider three training regimes—(F) where the R matrix is kept fixed to its initialization; (C) where a common R is trained for all layers; and (P) where a separate R is trained for each layer.

	Exact	k = 16		k = 24			k = 32			
	Baseline	\mathbf{F}	C	P	\mathbf{F}	C	P	\mathbf{F}	C	P
Test Acc	uracy									
MLM	66.0	63.9	64.5	63.6	63.1	65.5	64.7	64.5	64.7	65.6
MNLI	81.6	75.8	77.1	76.7	76.5	79.3	78.7	77.2	79. 7	79. 7
FLOPs Ratio	1.0	0.375		0.5625			0.75			

But Fig. 9 also provides a comparison in terms of equivalent computational cost, and we see here that the per-query setting affords a better trade-off due to its lower cost of reconstruction.

3.3 Network Performance

We next look beyond squared errors in reconstructed attention scores, and evaluate the effect of this approximation on overall network performance. As expected, simply introducing the approximation in a network that has been trained with exact attention performs poorly (see supplement). Instead, we consider training transformer models with the partial computation and reconstruction built-in.

We evaluate approximation performance in the BERT_{BASE} model, on the pre-training MLM task [6] followed by fine-tuning for entailment classification on the Multi-Genre NLI corpus [23] (a part of the GLUE benchmark [22]). We introduce approximate attention computation in all but the last layer (where typically only the embedding of a single token is retained, and approximation would offer no computational benefit), and use per-query reconstruction from partial sets of different numbers of exact query-key pair scores k. We select these sets P using our greedy algorithm, and then initialize the reconstruction matrix R to its optimal value computed using (7)—based on a covariance matrix computed from all layers and heads of a baseline exact BERT_{BASE} model.

All weights of the network are trained end-to-end, back-propagating through the partial selection of exact dot products in P, and reconstruction with the linear transform R. Moreover, while the set P is kept constant, we evaluate different approaches for R. In one approach, we keep R fixed to its optimal initialized value. We also consider updating R during training—as a common matrix for all layers, as well as learning a different R for each layer.

These results are summarized in Table 1. We find that training \boldsymbol{R} rather than keeping it fixed is beneficial—suggesting that its initial value, being optimized for squared error, may not be optimal for accuracy. Interestingly, training a common \boldsymbol{R} for all layers appears to be beneficial for smaller values of k, while larger k benefits from per-layer training. Overall, we find only a modest drop in accuracy for a reduction of 25-45% in the attention computation cost (for k=32 and 24).

4 Discussion

In this paper, we analyzed the distribution of attention scores generated by transformers on natural language inputs, and found them to lie in a relatively low-dimensional subspace. We found this behavior to hold across different layers and models, and found significant overlap between their eigen subspaces—indicating that this phenomenon is fundamentally a product of the underlying language structure. Our analysis can serve as a useful and principled foundation for approximate attention approaches, and we propose one such approach based on partial computation followed by reconstruction. Our results, both in terms of squared reconstruction error and trained network performance, indicate that this is a promising direction for future research.

We want to emphasize, however, that our specific partial computation and reconstruction method is only one possible way of exploiting this low-dimensional variance structure, and we expect future work will explore others. Moreover, while our analysis was restricted to English language

datasets, a natural question to ask is whether these findings also hold for other languages, and to other domains—such as computer vision, where transformers act on tokens representing image patches instead of words. Finally, a limitation of our method is that it assumes a fixed sequence length and is challenging to employ when this length is large. Generalizing our approach to work on longer sequences—potentially by modeling correlations within sub-sequences and applying it on all translated sub-sequences—is another interesting direction of future work.

References

- [1] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document Transformer. *arXiv* preprint arXiv:2004.05150, 2020.
- [2] Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, Jennifer C Lai, and Robert L Mercer. An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1):31–40, 1992.
- [3] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse Transformers. *arXiv preprint arXiv:1904.10509*, 2019.
- [4] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. arXiv preprint arXiv:2009.14794, 2020.
- [5] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does BERT look at? an analysis of BERT's attention. *arXiv* preprint arXiv:1906.04341, 2019.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional Transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations, ICLR*, 2021.
- [8] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-Transformer. arXiv preprint arXiv:1902.09113, 2019.
- [9] John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, 2019.
- [10] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient Transformer. arXiv preprint arXiv:2001.04451, 2020.
- [11] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/2c601ad9d2ff9bc8b282670cdd54f69f-Paper.pdf.
- [12] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *Technical Report, OpenAI*, 2018.
- [13] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *Technical Report, OpenAI*, 2019.
- [14] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.
- [15] Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. Fixed encoder self-attention patterns in transformer-based machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 556–568, 2020.
- [16] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. Transactions of the Association for Computational Linguistics, 8:842–866, 2020.
- [17] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing Transformers. *arXiv preprint arXiv:2003.05997*, 2020.
- [18] Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention in transformer models. arXiv preprint arXiv:2005.00743, 2020.

- [19] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *arXiv* preprint arXiv:2011.04006, 2020.
- [20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, pages 5998–6008, 2017.
- [21] Jesse Vig, Machine Learning, and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. *ACL* 2019, page 63, 2019.
- [22] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In 7th International Conference on Learning Representations, ICLR 2019, 2019.
- [23] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.
- [24] Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. *arXiv* preprint arXiv:1901.10430, 2019.
- [25] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are Transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020.
- [26] Chulhee Yun, Yin-Wen Chang, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. O(n) connections are expressive enough: Universal approximability of sparse transformers. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 13783–13794. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/9ed27554c893b5bad850a422c3538c15-Paper.pdf.
- [27] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

Supplementary Material

Table 2: Details of the BERT models used in this paper.

Model	Layers	Hidden size	Num heads
BERT _{SMALL}	6	768	12
$BERT_{BASE}$	12	768	12
$BERT_{LARGE}$	24	1024	16

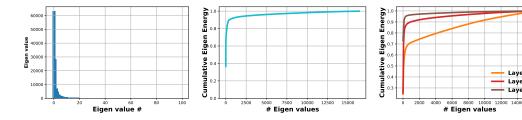


Figure 10: **Eigen values of** C_a . Left: Barplot of the top 100 eigen values of the covariance matrix C_a of attention scores aggregated over the entire network of a BERT_{LARGE} model. Middle: Cumulative sum of eigen values of C_a . Both show that C_a is approximately low rank with top 125 eigen vectors capturing > 80% of the energy. Right: Cumulative sum of eigen values of attention scores covariance matrix C_a^l for different layers of a BERT_{LARGE} model. We notice that later layers in the network have smaller rank.

A Network Training Details

We used the same setting as in BERT [6], including using their codebase⁸, to train the various Transformer models. We pre-trained the models on English Wikipedia and Books datasets [27]. We used inputs of sequence length 128 and trained the model using the Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks. We trained these models for 450k steps with a batch size of 1024, using the Adam optimizer with a peak learning rate of 1e-4, a linear warmup for the first 10k steps, followed by linear decay. We also used weight decay of 1e-4 and dropout of 0.1.

B Eigen analysis of BERT_{LARGE} model

Figures 10-15 present eigen analysis results analogous to those shown in Sec. 2 for a BERT_{LARGE} model. We notice similar behavior as the BERT_{BASE} model with approximate low rank attention scores variation and large subspace overlap across different settings.

C Mean-subtracted Attention Scores

All of our analysis in Sec. 2 was on the variability of raw pre-softmax attention scores. Although the softmax operation is invariant to the mean value of each row in the attention score matrix, we did not subtract this mean in our main analysis in order to characterize the variability in the raw query-key similarities, and because during partial computation, the value of this mean would be unknown.

For completeness, we also present the eigenspectrum of covariance matrices where all per-row means have been removed in Fig. 16, showing the fraction of total energy captured by the principal eigenvectors (here, both total energy and eigenvectors and values are computed from the modified covariance matrix). We compare this to the energy profile of the original covariance matrix, and find them to be largely similar. We also plot the fraction of energy contribution from variability of the per-row means in the original covariance matrix, and find it to be roughly half of the first eigenvalue. Thus, inclusion of per-row means in our analysis has no meaningful effect on the conclusions.

⁸https://github.com/google-research/bert

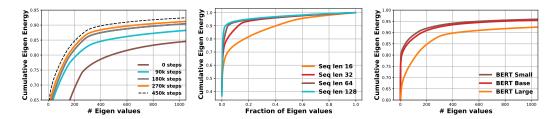


Figure 11: **Eigen values of** C_a . Cumulative sum of eigen values of attention scores covariance matrix C_a of a BERT_{LARGE} model - Left: after varying number of training steps. We notice that the rank slightly decreases throughout training with a large reduction in the beginning. Middle: for different sequence length inputs. Note that x-axis here denotes the fraction of eigen values. Right: for varying model sizes. Note that rank slightly increases with model size.

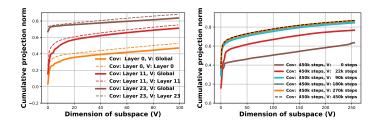


Figure 12: **Subspace similarity**. We plot the projection norm (cumulative energy eq 4) to measure subspace overlap for different choices of attention scores C_a and subspaces V. Left: Cumulative energy of covariance matrices (C_a^l) of different layers projected onto the top 256 eigen vectors of global covariance matrix C_a of a BERT_{LARGE} model. We notice that there is substantial overlap in eigen subspaces of global and per layer attention scores. Right: Cumulative energy of covariance matrix after full training projected onto eigen vectors of covariance matrices after different numbers of training steps. We notice that the overlap increases quickly as training progresses.

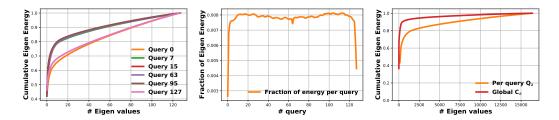


Figure 13: **Per-query eigen spectrum.** Left: Cumulative eigen values sum for different queries for a BERT_{LARGE} model. We notice that first and last queries, which are special tokens in BERT, have consistently higher rank compared to rest. Middle: Fraction of total eigenvalue sum for each query. We again notice that the special tokens have lower energy. Right: for each $k = i \times 128$, take top i eigenvalues of each per-query covariance matrices. This is contrasted against the energy plot for the global attention scores patterns.

D Approximate Attention during Inference

In Table 1, we considered the effect on accuracy from using approximate attention scores (due to reconstruction from partial computation), with models that were trained with this approximation. In Table 3, we report the effect of taking a standard model trained with exact attention scores and introducing the use of approximate attention only during inference. We show results for using reconstructions from partial computation, as well as from replacing the exact attention scores in each row with their best approximation from a limited number of per-query eigenvectors. We find that the performance in this case is notably worse than in Table 1 where, unlike in this case, all layers in the models had the opportunity to adapt to the attention approximation.

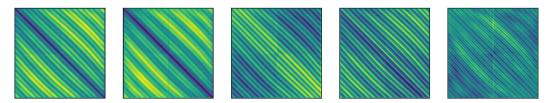


Figure 14: **Global principal components**. Visualization of attention score patterns captured by the top 5 eigenvectors of C_a of a BERT_{LARGE} model. We notice that the leading principal components capture predominantly shifted diagonal patterns.



Figure 15: **Per-query principal components**. Visualization of per-query attention score patterns from the top 5 eigenvectors (1st for the leftmost Figure, then 2nd etc.) from each query stacked into their corresponding rows for a BERT_{LARGE} model. Note the similarities between the top eigenvectors of the per-query attention scores and global attention scores (Fig 14).

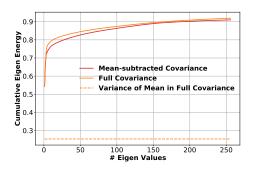


Figure 16: **Analysis after Per-row Mean Subtraction.** Here, we compare the eigen value energies for a covariance matrix from the attention scores of a BERT_{BASE} model, to a version of the covariance matrix computed after subtracting the mean score value in each row of each sample. We find that both versions of the covariance matrix exhibit similar low rank behavior, and that the energy of per-row means makes a negligible contribution to the overall variability of attention scores.

Table 3: **Performance of Pre-trained Network with Approximate Attention during Inference.** We evaluate the effect of using approximate attention scores during inference with a standard BERT_{BASE} model trained with exact attention computation. We report results for two approaches to attention approximation: **EP:** which approximates attention scores using projections onto the per-query top-k eigenvectors, and **PC:** where scores are reconstructed from k exact attention scores per query.

	Exact Baseline	k = EP	= 16 PC	k = EP		k = EP	64 PC
MLM Accuracy	66.0	50.8	51.6	55.5	54.9	61.0	59.7