# Fisher Information aware dynamic compression of Language Transformer networks using SVD

**S.B.H.C. Hofstee**
**Supervisor: Le Viet Duc**
University of Twente
s.b.h.c.hofstee@alumnus.utwente.nl

## Abstract

We propose a novel approach to Singular Value Decomposition (SVD) for low-rank compression, addressing limitations in previous methods such as Fisher Weighted SVD (FWSVD) and True Fisher Weighted SVD (TFWSVD), which apply uniform compression ratios across layers without considering intra- and inter-layer information characteristics. Unlike FWSVD and TFWSVD, our approach dynamically determines layer-wise compression ratios based on these characteristics, enhancing task performance efficiency. Specifically, we explore three novel methods in which ranks are dynamically determined for low-rank compression based on the inter- or intra-layer Fisher Information (FI): (1) dynamically determining the rank for low-rank compression based on inter-layer Fisher Information (FI), (2) maintaining a fixed percentage of intra-layer FI and (3) optimizing to maximize total (or layer-wise) FI given a fixed overall compression ratio. These methods are evaluated on a transformer-based language model and benchmarked against the state-of-the-art. One of the proposed methods, relying on specifying a fixed percentage of FI to keep per layer, has been shown to outperform the current state of the art on average in excess of 5%, and very significantly for inference and similarity tasks. The work furthermore provides valuable insights for future work to further explore the dynamic compression of layers in transformer networks using FI, in particular by displaying the effectiveness of dynamic compression using intra-layer FI.

## Introduction

During the past few years, Transformers have proven to be of great use for Natural Language Processing tasks, as well as for computer vision tasks (Dosovitskiy et al. 2020). Transformer networks are often vast, and more often than not contain tens of millions to billions of parameters (Dehghani et al. 2023). The mere size of these models makes it rather difficult to deploy said models on widely available hardware, causing access to transformer networks to be restricted to only those institutions that can afford to invest great amounts into extremely performant hardware. For this reason, the compression of transformer-based networks is of great interest.

The reduction of the number of parameters in transformer-based language- and vision-based models has been shown to be able to be achieved by various methods, among which making use of Singular Value Decomposition (SVD) of weight matrices in order to decrease the number of parameters in vision transformers (Hajimolahoseini et al. 2022). Conventional low-rank compression using Singular Value Decomposition (SVD) however assumes that matrix rows with lower singular values carry less importance with respect to the target task. Work by Hsu et al. (2022) shows that this is not necessarily the case, and has advanced low-rank compression using SVD by proposing Fisher Weighted Singular Value Decomposition (FWSVD), a Fisher-Information weighted objective function for SVD in language-based models. While effective, FWSVD makes use of a simplification, making weight matrix rows share information to obtain a closed-form solution. This simplification limitation present in FWSVD was mitigated by Hua et al. (2022), who with True Fisher Weighted Singular Value Decomposition (TFWSVD), numerically optimised the weighted objective function introduced by Hsu et al. (2022). However, these methods apply a fixed compression ratio to each layer within the language transformer network and hence ignore any inter- or intra-layer information characteristics specific to the layers when determining the low-rank compression ratios for these layers. We argue that dynamically determining layer-wise low-rank compression ratios based on these inter- and intra-layer information characteristics can be used in order to increase the task-performance efficiency of (T)FWSVD, where the task-performance efficiency is defined as the number of parameters required to obtain a specific performance on the target task. To the best of our knowledge, this paper is the first to propose utilizing inter- and intra-layer Fisher information characteristics in order to dynamically determine low-rank compression ratios, even more so for its specific application to transformer networks.

This work builds upon the foundational studies (Hsu et al. 2022; Hua et al. 2022), by proposing three distinct methods that take into account the mentioned inter- and intra-layer information characteristics in order to improve the task-performance efficiency of Fisher Weighted SVD. Taking into account the inter- and intra-layer information characteristics is done by exploring dynamically determining the rank for low-rank compression based on the inter-layer Fisher Information (FI), dynamically determining the rank for low-rank compression based on keeping a fixed percent-

age of the intra-layer FI, and dynamically determining the rank for low-rank compression - optimizing to maximize total (or layer-wise) FI given a fixed compression ratio. In particular, the main contributions of this work are as follows.

(1) Improve upon (Hsu et al. 2022) and (Hua et al. 2022) in order to improve task-performance efficiency of Fisher Weighted SVD by proposing novel compression methods which take into account intra- and inter-layer information characteristics to determine layer-wise low-rank compression ratios where previous work compressed all layers equally.

(2) Benchmarking these novel methods by applying them to various natural language processing tasks using BERT.

(3) Extract directions for future work from the experiments in this work in order to improve language transformer compression even further.

# Background

## Using Singular Value Decomposition to compress weight matrices

Singular value decomposition (SVD) is a way to decompose a matrix $W \in R^{M \times N}$ into components $U \in R^{M \times r}$, $V \in R^{N \times r}$ and diagonal matrix $\Sigma$, such that

$$W = U \Sigma V^T, \tag{1}$$

where the values $\sigma_i \in \Sigma \mid \sigma_i = \Sigma_{i,i}$ are referred to as singular values.

Notably, the singular values in $\Sigma$ are arranged in descending order of magnitude, meaning that at rank 1 ($\Sigma_{0,0}$), the largest singular value is present.

Compression by means of using Singular Value Decomposition as a way to create a low-rank approximation of a weight matrix $W \in R^{M \times N}$ is done by decomposing this weight matrix $W$ into components $U \in R^{M \times r}$, $V \in R^{N \times r}$, but this time with the diagonal matrix $\widetilde{\Sigma} \in R^{r \times r} \mid r <= min\{m, n\}$ where $\widetilde{\Sigma}$ contains the same values as $\Sigma$, except $\widetilde{\Sigma}$ being truncated to only include the $r$ singular values of greatest magnitude.

$$W \approx U \widetilde{\Sigma} V^T \tag{2}$$

For $r = min\{m, n\}$ (full rank) $W = U \widetilde{\Sigma} V^T$.

The general idea of using Singular Value Decomposition to compress neural network architectures, is well-established. Using SVD to compress linear and convolutional layers in network architectures has been proposed to work for Convolutional Neural Networks (CNN) as early as 2014 in work by Denton et al. (2014), which tested its performance on an image classification task using the ImageNet 2012 dataset (Deng et al. 2009). The method has not only proven to be promising when considering image classification but has also proven its usefulness for the compression of neural architectures in other tasks, such as in acoustic scene classification (using CNN) (Wang, Li, and Wang 2019) and speech recognition (using Recurrent Neural Network (RNN) architecture) (Prabhavalkar et al. 2016).

Schotthöfer et al. (2022) categorize the decomposition of weight matrices into two distinct categories: Fixed low-rank and dynamic low-rank approaches, where fixed low-rank approaches define a specific rank upfront by for example decomposing a network followed by fine-tuning (Denton et al. 2014; Sainath et al. 2013; Lebedev et al. 2015) or enforcing a fixed low-rank during training for weight matrices (Jaderberg, Vedaldi, and Zisserman 2014; Khodak et al. 2021). Rank-adaptive approaches automatically determine and adapt the matrix rank after training, for example, based on metrics such as a PCA-energy-based metric (Kim, Khan, and Kyung 2019) and a metric using a validation dataset to maximize accuracy - using Variational Bayes Matrix Factorization (VBMF) (Nakajima et al. 2013) to sample ranks. The proposed novel methods in this work fall into this rank-adaptive category.

## Observed Fisher Information

The Fisher information is defined as a way to measure the information that variable $X$ contains with respect to an unknown parameter $\theta$, which in itself parameterises the distribution modelling variable $X$.

The observed Fisher Information used in our compression method is the sample-based version of the Fisher Information $\mathcal{I}$ where

$$\mathcal{I}_\theta = E[(\frac{\delta}{\delta \theta} log f(X; \theta))^2 \mid \theta] \tag{3}$$

For variable $X$ distributed as $f(X; \theta)$.

Now, what one would like to do is approximate the Fisher information solely based on the observed variable $X$, which can be defined as:

$$\mathcal{I}_\theta \approx \widetilde{\mathcal{I}_\theta} = \sum_{i=1}^{|D|} log f(d_i; \theta) = \frac{1}{|D|} \sum_{i=1}^{|D|} (\frac{\delta}{\delta \theta} \mathcal{L}(d_i; \theta))^2 \mid d \in D \subset X \tag{4}$$

Where $\mathcal{L}$ is the loss function used for the model.

In order to see the intuition behind using the Fisher Information to either use it as a measure of importance for a weight matrix $W$, or as to weigh SVD based on it, consider loss function $\mathcal{L}(d_i; \theta)$ in which a sufficiently small change $\Delta\theta$ is added to model parameter $\theta$

$$\mathcal{L}(d_i; \theta + \Delta\theta) \tag{5}$$

We can use the first-order Taylor series expansion of the loss function around $\theta$ to obtain

$$\mathcal{L}(d_i; \theta + \Delta\theta) \approx \mathcal{L}(d_i; \theta) + \frac{\delta \mathcal{L}(d_i; \theta)}{\delta \theta} \Delta\theta \tag{6}$$

This shows that when the gradient of the loss function $\mathcal{L}$ with respect to the parameter $\theta$ is large, even a small perturbation in $\theta$ will lead to a significant change in the loss function. This observation highlights the intuition behind the definition of the observed Fisher Information $\widetilde{\mathcal{I}_\theta}$ as presented in equation 4. Specifically, $\widetilde{\mathcal{I}_\theta}$ effectively captures the average

sensitivity of the loss function $\mathcal{L}$ to changes in the parameter $\theta$ across the sample dataset $D$.

The idea of utilizing the Fisher information in order to improve neural network compression, not necessarily using SVD, is likewise well-established. Tu et al. (2016) proposed a scheme to reduce model size by utilizing Fisher information in order to discard lower-importance parameters and allocate more quantization bits to those parameters with higher Fisher information in deep neural networks. Theis et al. (2018) show that using a combination of Fisher pruning and knowledge distillation (Hinton, Vinyals, and Dean 2015), distilling knowledge from an ensemble of models into a single distilled model, a 10x speedup could be obtained for human gaze prediction. The Fisher pruning herein consisted of removing those feature maps in case of convolutional architectures (as typically, implementations of convolution operations might have difficulty exploiting sparse filters in order to speed up the operation) that do not significantly contribute to the performance of the model. In this approach by Theis et al. parameters (or feature maps in the case of CNN architectures) are pruned one by one in ascending order of Fisher information. The mentioned works show the effectiveness of utilizing FI as a measure to determine the influence a given parameter or structure has on task performance, which is an assumption this work relies on.

### Weighing SVD approximation with observed Fisher Information in language models

Conventional low-rank approximation has an objective function that is defined as minimizing the Frobenius norm between original matrix $W$ and the (low-rank) approximation $AB$

$$\min_{A,B} ||W - AB||_2 \tag{7}$$

where for SVD, $A = U\tilde{\Sigma}$ and $B = V^T$.

Where conventional low-rank approximation tries to approximate $AB$ as close to $W$ as possible, Hsu et al. (2022) have identified that matrix rows with lower singular values do not necessarily carry less importance with respect to the target task, a notion previously supported by Lyu et al. (2023). This is shown by splitting the singular values up in ordered groups of 10%, where the first group contains the highest 10% of the singular values, whereas the last group contains the smallest 10% of singular values, as can be seen in Figure 1. Hsu et al. (2022) have identified that one can obtain the importance of each element $W_{ij}$ by weighing the conventional low-rank approximation objective with the observed Fisher information $\widetilde{\mathcal{I}}_\theta$ named Fisher Weighted SVD (FWSVD) such that the weighted objective function be

$$\min_{A,B} \sum_{i,j} \widetilde{\mathcal{I}}_\theta (W_{ij} - (AB)_{ij})^2. \tag{8}$$

The authors of (Hsu et al. 2022) note that this weighted SVD objective function does in general not have a closed form solution, and therefore simplify by making a single row in $W$ share one importance such that $\widetilde{\mathcal{I}}_{\theta W_i} = \sum_j \widetilde{\mathcal{I}}_{\theta W_{ij}}$, causing the objective function to be

$$\min_{A,B} ||\widetilde{\mathcal{I}}_\theta W - \widetilde{\mathcal{I}}_\theta AB||_2. \tag{9}$$

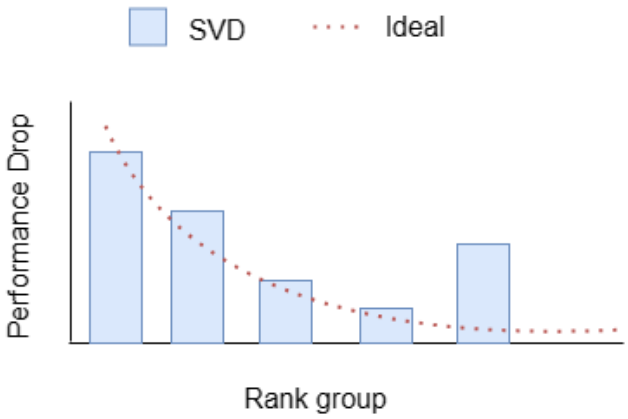


Figure 1: Lower rank groups do not necessarily have less effect on task performance. Adaptation from Hsu et al. (Hsu et al. 2022)

This can then be solved using standard SVD on $\widetilde{\mathcal{I}}_\theta W$, where $svd(\widetilde{\mathcal{I}}_\theta W) = (U^*, \Sigma^*, V^*)$, making the solution for $A$ and $B$ from Equation 9 be $A = \widetilde{\mathcal{I}}_\theta{}^{-1} U^* \tilde{\Sigma}^*$ and $B = V^{*T}$.

The objective function of equation 8 can however directly be optimised using an optimization algorithm. This numerical optimization has previously been done by Hua et al. (2022) and was named True Fisher Weighted SVD (TFWSVD), as an improvement on the work by Hsu et al. (2022), showing that this numerical performance is able to outperform the state-of-the-art achieved by Hsu et al. (2022). In the approach by Hua et al. Equation 8 is directly optimized using a combination of Adam (Kingma and Ba 2015) and Stochastic Gradient Descent (SGD) (Kiefer and Wolfowitz 1952). In this scheme, the simplified objective function as posed in Equation 9 is used as a threshold in order to determine when to switch the optimizer from Adam to SGD: when a loss that is equal to the value of Equation 9, after applying SVD (analytical solution) on $\widetilde{\mathcal{I}}_\theta W$, the optimizer is switched to SGD.

Note that both FWSVD and TFWSVD do not take into account any inter- or intra-layer information characteristics when determining the ranks for low-rank compression, both equally compressing every layer. This work, in contrast, does take into account said inter- and intra-layer Fisher information characteristics, dynamically determining the compression ratio for each layer based on said characteristics.

## Method

### Determining weight matrix compression based on observed Fisher Information

The first approach proposed in this work is using the summed Fisher information matrix $\widetilde{\mathcal{I}}_{\theta W}$ for an entire weight matrix $W$, in order to dynamically determine the rank $r$ given compression ratio $c$. This method utilizes the inter-layer information characteristics of the layers within the network by using the share of information a layer has as a proxy for its importance with regard to task performance.

The approach follows a number of steps. First, the summed Fisher information matrix $\widetilde{\mathcal{I}}_{\theta W}$ for an entire weight matrix is calculated

$$\widetilde{\mathcal{I}}_{\theta W} = \sum_{ij} \widetilde{\mathcal{I}}_{\theta W_{ij}}. \tag{10}$$

After this, the total Fisher information $F_{total}$ over all weight matrices, is calculated

$$\widetilde{\mathcal{I}}_{total} = \sum_{W \in Q} \widetilde{\mathcal{I}}_{\theta W} \tag{11}$$

where $Q$ is the set of all weight matrices.

Next, the Fisher Information is normalized to create a weight factor $p_W$ for each $W \in Q$

$$p_W = \frac{\widetilde{\mathcal{I}}_{\theta W}}{\widetilde{\mathcal{I}}_{total}} \tag{12}$$

such that $\sum_{W \in Q} p_W = 1$.

Now, a problem arises, which is determining which compression ratio $c_W$ to use for each weight matrix $W \in Q$. Clearly, the Fisher information-based weight factor $p_W$ should determine the compression ratio, while the sum of all compression ratios should sum to $c$, $\sum_{W \in Q} c_W = c$.

At the same time, the sum of all compression ratios equaling to $c$, does not mean that the overall compression ratio is in fact equal to $c$, as each weight matrix $W \in Q$ may have a different size, and hence when a smaller sized weight matrix $J$ would get compressed more, and a larger sized weight $K \mid \sum dim(K) > \sum dim(J)$ matrix would get compressed less, while $\frac{1}{2}\sum c_K + c_J = c$, the total number of parameters might not be the same.

For this reason, one can identify two distinct optimization objectives to determine the compression ratios.

(1) Optimizing for a fair compression ratio for all layers, that is, those with a higher Fisher information should obtain a lower compression ratio, disregarding the absolute size of the weight matrix of the layer. The average of these compression ratios should equal to $c$. This can be achieved by defining a loss function

$$\mathcal{L}_{fair}(\alpha) = \left( c - \frac{1}{|Q|} \sum_{W \in Q} \alpha(1 - p_W) \right)^2. \tag{13}$$

Where $\alpha$ is some parameter to be optimized, and notably, the same for all $W \in Q$. Additionally, note that a higher $p_w$ leads to less compression, which is important as one would like to compress those weight matrices with a higher Fisher information less.

The fair optimization function as shown in Equation 13, has an analytical solution, which makes it much easier to determine $\alpha$. Note that by $\sum_{W \in Q} p_W = 1$, we have

$$\alpha = \frac{c}{(1 - \frac{1}{|Q|})} \tag{14}$$

(see derivation in Appendix ).

Note that despite $a$ not being dependent on $p_W \mid W \in Q$, the compression ratio for a weight matrix $W$ is defined as $\alpha(1 - p_W)$, and hence does take into account the share of information of each layer.

(2) Optimizing for an overall compression ratio of $c$, hence also taking into account the sizes $m_w$, $n_w$ of the weight matrices $W$ in a layer. For this, one can define the loss function

$$\mathcal{L}_{overall}(\alpha) = \left( \frac{S_{total}}{c} - \sum_{W \in Q} \frac{(m_W \times n_W)}{\alpha(1 - p_W)} \right)^2. \tag{15}$$

Where $\alpha$ is some parameter to be optimized and $S$ is the total number of parameters in the to-be-compressed layers. Note that a higher $p_w$ leads to less compression, as $\alpha(1 - p_W)$ becomes smaller.

Note that the overall optimization function shown in Equation 15, also has an analytical solution

$$\alpha = \frac{c}{S_{total}} \sum_{W \in Q} \frac{(m_W \times n_W)}{(1 - p_W)}. \tag{16}$$

(see derivation in the Appendix).

Note that for $\mathcal{L}_{overall}(\alpha)$ and $\mathcal{L}_{fair}(\alpha)$, $0 \leq p_W \leq 1 \mid W \in Q$ and $\sum_{W \in Q} p_W = 1$.

The compression ratio $c_W$ is related to the (reduced) rank $r_W$ as follows

$$c_W = \frac{m_W \times n_W}{(m_W \times r_W) + (n_W \times r_W)} = \frac{m_W \times n_W}{r_W(m_W + n_W)}. \tag{17}$$

Where solving for $r_W$ gives

$$r_W = \frac{m_W \times n_W}{c_W(m_W + n_W)}. \tag{18}$$

This means that a matrix $W \in Q$, is split up into matrices of size $m_W \times r_W$, $r_W \times r_W$ and $r_W \times n_W$ respectively.

## Weight matrix compression retaining a percentage of FI

Another approach, which utilizes the intra-layer information characteristics of a given layer, is to keep a fixed percentage of the Fisher Information per layer and dynamically have the compression ratio be inferred from this.

For this, a mapping must be made between the Fisher Information associated with the original weight matrix $\widetilde{\mathcal{I}}_{\theta W}$ and the SVD of the original weight matrix. This is done by first summing the Fisher Information values in $\widetilde{\mathcal{I}}_{\theta W}$ across each row to obtain a vector where each element represents the sum of the Fisher Information values for the corresponding row

$$\tilde{I}_{W_{reduced_i}} = \sum_{j=0}^{n} \tilde{I}_{W_{i,j}}. \tag{19}$$

Then we diagonalize $\tilde{I}_{W_{reduced}}$ and project it onto the space defined by $V$

$$\tilde{I}_{W_{projected}} = V^T diag(\tilde{I}_{W_{reduced}}) V. \tag{20}$$

After which the singular values in $\Sigma$ are weighed with $\tilde{I}_{W_{projected}}$

$$\Sigma_{weighed} = \Sigma \odot \tilde{I}_{W_{projected}}. \quad (21)$$

$\Sigma_{weighed}$ is then re-ranked to have the weighted singular values be ordered decreasingly.

In order to then retain the specific percentage of information per layer, the algorithm iterates over each re-ranked row, starting from the first row (that is the row with the highest FI associated with it), and keeps including rows until the desired FI threshold has been reached or exceeded. After this, a list of sorted indices $l$ is made, containing the indices of weighed singular values in $\Sigma_{weighted}$, in descending order. The top $k$ indices from the list of sorted indices $l$ are then used in order to determine the top-k components in $\Sigma$, determining $\tilde{\Sigma}$.

The intuition behind this approach is that for each compressed layer, a percentage of information is kept, where the hypothesis is that this ensures for each layer, information required to perform the task is retained. At the same time, this approach has an apparent downside which is one not being directly in control as to how much a model is compressed. Although for a specific trained model, one would be able to map the percentage of information kept to a corresponding compression ratio by trial and error, however, this exact mapping would be different for every trained model.

At the same time, a rough general mapping between the percentage of information kept and the compression ratio obtained from said percentage kept might be found. Such mapping, combined with the performance of this approach might lead one to still prefer this approach over others.

## Weight matrix compression by dynamically determining the rank for low-rank compression - optimizing to maximize total FI given a fixed compression ratio

Having a fixed compression ratio and optimizing to maximize the total Fisher Information is a method utilizing the inter-layer information characteristics - optimizing compression ratios for all layers such that the overall information kept is maximized. This can be done by defining a loss function as follows

$$L(C,\lambda) = -\sum_{W\in Q}\frac{\tilde{I}_W}{c_W} + \lambda\left(\sum_{W\in Q}\frac{m_W\cdot n_W}{c_W} - \frac{S_{total}}{c}\right)^2 = -\sum_{W\in Q}\frac{\tilde{I}_W}{c_W} + \lambda\left(\sum_{W\in Q}\frac{m_W\cdot n_W}{c_W S_{total}} - \frac{1}{c}\right)^2. \quad (22)$$

Where overall compression $c \geq 1$ and $\lambda$ is a Lagrange multiplier to enforce the constraint that the total number of parameters should be equal to $\frac{S_{total}}{c}$. $c_w$ here is the compression ratio that differs for each linear layer, $c_W \in C$, where $c_W \geq 1$. We normalize the Lagrange multiplied part of the equation by dividing by $S_{total}$.

Note however that the optimization function from Equation 22, which tries to find the most optimal $c_W \in C$, is similar to the simple mathematical function $\frac{1}{x}$. This means that the function is discontinuous at $x = 0$, and hence no derivative exists at $x = 0$. Similarly, no derivative exists for any $c_W = 0 \mid c_W \in C$.

Despite the constraint that $c_W \geq 1$, during optimization, some $c_W \in C$ could be set equal to 0 during some intermediate update. For this reason, the loss function is defined as the following piecewise loss function

$$\begin{cases} L(C,\lambda) = -\sum_{W\in Q}\frac{\tilde{I}_W}{c_W} + \lambda\left(\sum_{W\in Q}\frac{m_W\cdot n_W}{c_W S_{total}} - \frac{1}{c}\right)^2 & \forall c_W \in C,\ c_W \neq 0 \\ L(C,\lambda) = \infty & \exists c_W \in C \text{ such that } c_W = 0. \end{cases} \quad (23)$$

The idea behind this approach is that the total information, which consists of all information in those layers that are subject to compression, is maximized and hence the hypothesis is that the task performance is also maximized.

**Plotting the problem space** The space of this problem for a given compression ratio is one of dimensions $\mid Q \mid$, given there exists a compression ratio $c_W$ for each $W \in Q$.

To get an idea of this problem space, a Monte Carlo sampling has been made of the space, given a set of 12 layers $\tilde{W} \in \tilde{Q}$ simulating $W \in Q$ of sizes $m_W, n_W \mid 1 \leq [m_W, n_W] \leq 100$, where these $m_W, n_W$ have been sampled from a uniform distribution. Additionally, for each simulated layer $\tilde{W} \in \tilde{Q}$, an accompanying $\tilde{\tilde{I}} \mid 0.1 \leq \tilde{\tilde{I}} \leq 10.0$ simulating $\tilde{I}$, the simulated observed information, was sampled from a uniform distribution.

Then, $\tilde{c}_W$ for each layer $\tilde{W} \in \tilde{Q}$ are uniformly sampled, where $1 \leq c_W \leq 10$.

A sample set $\tilde{C} \mid \tilde{c}_W \in \tilde{C}$ is found to be acceptable in case it allows for the desired overall compression ratio $c \pm 5\%$.

As one might expect, uniform sampling of $\tilde{C}$ might make it take rather long to reach those $\tilde{c}_W \in \tilde{C}$ such that the sampled total compression $\tilde{c} = c \pm 5\%$ holds, where

$$\tilde{c} = \sum_{W\in Q}\frac{\frac{\tilde{m}_W \times \tilde{n}_W}{\tilde{c}_W}}{\tilde{S}_{total}} \mid \tilde{c}_W \in \tilde{C}, \tilde{W} \in \tilde{Q}. \quad (24)$$

For this reason, after every sample taken, a mechanism to adjust the sampling of $\tilde{C}$ is used.

When compression ratio $\tilde{c} \neq c$, all items $c_W \in \tilde{C}$ are scaled by a factor $z = \frac{c}{\tilde{c}}$. This means that in case $\tilde{c} > c$ (too much compression), this scaling factor $z < 1$, whereas in case $\tilde{c} < c$ (too little compression), this scaling factor $z > 1$.

This sampling strategy leads to a collection of samples , where PCA (Wold, Esbensen, and Geladi 1987) is used to reduce the dimensionality of these vectors to 2 such that these can be used to plot a 3-D space in which the x and y axes are the two principal components, and the z-axis is the amount of total information kept. These plots can be observed in Figure 2.

Figure 2 clearly shows that it might prove difficult to optimize for maximum total information given a compression ratio when using a standard numerical optimizer, such as Adam or RMSProp, as the space contains a very large number of local optima. Therefore not only the numerical optimization methods are used in order to find an optimal solution to this problem which maximizes the total information

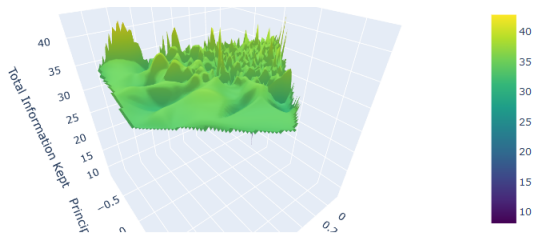


Figure 2: Plot of Monte Carlo uniformly sampled problem space: PCA of compression ratios vs total information

given a specific compression ratio, but, a sampling method such as described in order to plot the problem space is also to be considered. In this, the same Monte-Carlo method utilizing a mechanism to uniformly adjust the sampled $\tilde{c} \neq c$ is used to move $\tilde{c}$ into the direction of $c$, where the initial samples $c_W \in \tilde{C}$ are drawn from a uniform distribution where $1 \leq c_W \leq 10$.

## Experiments

### Language tasks and datasets

The benchmark tasks used in this work are the tasks present in the General Language Understanding Evaluation (GLUE) (Wang et al. 2019), which is commonplace to use in order to assess model compression performance (Khetan and Karnin 2020).

The GLUE benchmark consists of several tasks that can be categorized into three distinct classes

(1) single-sentence classification tasks
(2) similarity & paraphrase tasks
(3) inference tasks.

**Single-sentence tasks** Two specific tasks fall under the single-sentence tasks umbrella: The Corpus of Linguistic Acceptability (CoLA) (Warstadt, Singh, and Bowman 2019) single-sentence task, and the Stanford Sentiment Treebank (Socher et al. 2013) single-sentence task (Socher et al. 2013).

CoLA - The corpus for the CoLA task consists of acceptability judgements from books and journal articles on linguistic theory in the English language. The Matthews correlation coefficient (MCC)(Matthews 1975), where the MCC is a balanced measure taking into account true positives, true negatives, false positives and false negatives. The MCC is calculated as

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{25}$$

The value of the MCC can vary between -1 and 1, where 1 indicates a perfect prediction, 0 indicating a prediction no better than random and -1 indicating maximum disagreement between prediction and labels.

For the CoLA task, the measure is calculated using a combination of both the in- and out-of-domain sections of the test set.

SST-2 - The mentioned Stanford Sentiment Treebank (Socher et al. 2013) is comprised of sentences from movie reviews, which are sentiment-labeled by human annotators. For the task, sentence-level labels are considered, and a two-way class split (positive, negative) is used.

**Similarity and paraphrase tasks** The next group of tasks that make up the GLUE benchmark are the similarity and paraphrase tasks, which include the Microsoft Research Paraphrase Corpus (Dolan and Brockett 2005) (MRPC) task, the Quora Question Pairs dataset (Quora 2012) (QQP) task and the Semantic Textual Similarity Benchmark (Cer et al. 2017) (STS-B) task.

MRPC - The MRPC corpus consists of sentence pairs which were automatically extracted from news sources online, where semantic equivalences between pairs have been labelled by human annotators (positive or negative; semantically equivalent or not). As the classes are imbalanced (68% positive), both F1 score and accuracy are calculated, where both (Hsu et al. 2022) and (Hua et al. 2022) report the F1 score in their results.

QQP - The QQP dataset is comprised of pairs of questions from the question-answer-based community forum Quora. The task here is, similarly to MRPC, to determine whether a pair of questions is semantically equivalent or not. Again, given the classes in this dataset are imbalanced (63% negative), both F1 score and accuracy are calculated, where both (Hsu et al. 2022) and (Hua et al. 2022) report the F1 score in their results.

STS-B - STS-B consists of a collection of sentence pairs from natural language inference data, news headlines and video- and image captions, where each pair is humanly labelled with a similarity score between 1 and 5, which are to be predicted for this task. For this task, the Pearson correlation coefficient (PCC) is used as a performance metric.

**Inference tasks** The last class of tasks are the inference tasks, consisting of the Multi-Genre Natural Language Inference Corpus (Williams, Nangia, and Bowman 2018) (MNLI) task, the Stanford Question Answering Dataset (Rajpurkar et al. 2016) (QNLI) task, the Recognizing Textual Entailment (Wang et al. 2019) (RTE) task and the Winograd Schema Challenge (Levesque, Davis, and Morgenstern 2012) (WNLI) task.

MNLI - The MNLI task consists of a collection of sentence pairs with their corresponding textual entailment annotations. Each pair consists of a premise- and hypothesis sentence, where the task at hand is to predict whether the premise entails or contradicts the hypothesis (or neither). Premise sentences are sourced from ten different sources, among which are speech transcriptions and fiction. Additionally, the SNLI corpus (Bowman et al. 2015) is used for additional pre-training data.

QNLI - The QNLI corpus used for this task is a question-answering dataset consisting of pairs of questions with a corresponding paragraph, in which the corresponding paragraph contains the answer to the given question. Wang et

al. (Wang et al. 2019) have modified the dataset such that a pair is created between each question and each sentence in the corresponding paragraph, where notably pairs with a high lexical overlap are removed. The task then is to predict whether a given sentence answers the corresponding question. As Wang et al. (2019) mention, despite this removing the task of the model selecting the answer that contains the correct question, it does also remove the assumption that a given paragraph would always contain an answer to the question as well as, due to the fact that lexically similar pairs were removed, the assumption that lexical overlap between the question and sentence would correlate to the sentence containing the answer.

RTE - The RTE task datasets are a combination of different textual entailment challenges, where datasets are based on Wikipedia articles and news items. The dataset combination is then split into two distinct classes: *entailment* and *non-entailment*. For any datasets that make up the combined dataset which have more than two classes, *neutal* and *contradiction* classes are simply put into the *non-entailment* class. The task then is to correctly predict *entailment* or *non-entailment*.

WNLI - The WNLI is based on the Winograd Schema Challenge (Levesque, Davis, and Morgenstern 2012), which is a reading comprehension task where in sentences with pronouns, the referent of said pronoun must be selected. Examples are manually created in order to prevent simple statistical methods from being effective. Wang et al. (Wang et al. 2019) have converted the problem into a sentence classification one, where sentence pairs have been created which replace the pronoun that could be referred to by multiple referents, by each of the possible referents. It is then predicted whether the sentence in which the pronoun has been replaced by the possible referent is entailed by the original sentence. Wang et al. (2019) have furthermore created an additional test set which consists of new examples sourced from fiction books.

## Baseline model

The baseline language model used for the experiments in this work is a standard 12-layer BERT model (Devlin et al. 2019). BERT is a well-known language representation model and is designed to encode deep bidirectional representations from unlabeled text. It is an excellent choice to test novel compression methods given both its high performance in numerous natural language tasks (Devlin et al. 2019; Choi et al. 2021), as well as due to the fact that it can be easily fine-tuned for a range of natural language processing tasks, such as language inference and question answering, by adding a single additional output layer (Devlin et al. 2019).

At the same time, using BERT in order to determine the performance of compression schemes on natural language models appears to be commonplace (Khetan and Karnin 2020; Wang et al. 2022; Sanh, Wolf, and Rush 2020; Cao et al. 2020), and is also the choice of language model for the work on Fisher Weighted SVD by Hsu et al. (2022), which this works both makes direct comparisons with as well as improves upon.

BERT notably has an encoder-only transformer architecture. This means that the architecture does not contain a decoder as shown in Figure 4.

## Layers to compress

For the experiments, all linear layers are to be compressed using low-rank compression using the share-of-information, keeping a specific share of information or optimizing for overall information kept respectively. Despite the compression of embedding layers has however in previous work shown to be both feasible and effective (Chen et al. 2018), where said word embedding layer occupies 21.3% of the standard BERT model, this work does not compress the word embedding layer in order to maintain comparability with other work, akin to the work by Hsu et al. (2022).

Most layers that are to be compressed are an encoder-level layer. That is, these layers exist in one of the 12 encoder blocks of the BERT architecture. These layers are the query, key, value and output linear layers of the attention mechanism, as well as the intermediate and output linear layers of the encoder block.

For clarity, Figure 3 highlights these to be compressed encoder-level layers, which furthermore shows the architectural location of these to-be compressed layers within the BERT architecture.

Next, any linear layers that are not present in the encoder component of the BERT architecture are furthermore compressed, which is dependent on the downstream task on which BERT is fine-tuned.

## Common experiment setup [1]

Each of the proposed novel compression methods is run separately on the SST-2, MRPC, QQP, STS-B, MNLI and QNLI GLUE benchmark tasks, which have been described in the language tasks and datasets section of the experiments section. Additionally, a BERT-base (Devlin et al. 2019) benchmark is run without any compression, in order to obtain a benchmark for the exact unaltered base model used. For each task, the BERT-base model (Devlin et al. 2019) is first fine-tuned for three (3) training epochs, using a batch size of 32, a learning rate of 2e-5 and a max sequence length of 128 tokens.

Given that the approaches mentioned in the methodology section are ways to dynamically determine the compression ratio for a given layer $W \in Q$, the decomposition itself is set to be either using FWSVD (Hsu et al. 2022)-, or TFWSVD (Hua et al. 2022). For each set of these experiments, a set using either FWSVD or TFWSVD as a decomposition basis, the baseline is reported. That is, it is reported how well given the current configuration, these approaches work. For the TFWSVD experiment set, the Adam (Kingma and Ba 2015) optimizer is used using a learning rate of $0.001$, taking $200$ optimization steps. A clear overview of the experiments done per compression level can be seen in Figure 3, where firstly one of two Fisher-weighted SVD approaches is used,

[1]The implementation of all experiments mentioned in this paper are available at https://github.com/SebastiaanDev/transformer-compression-bert

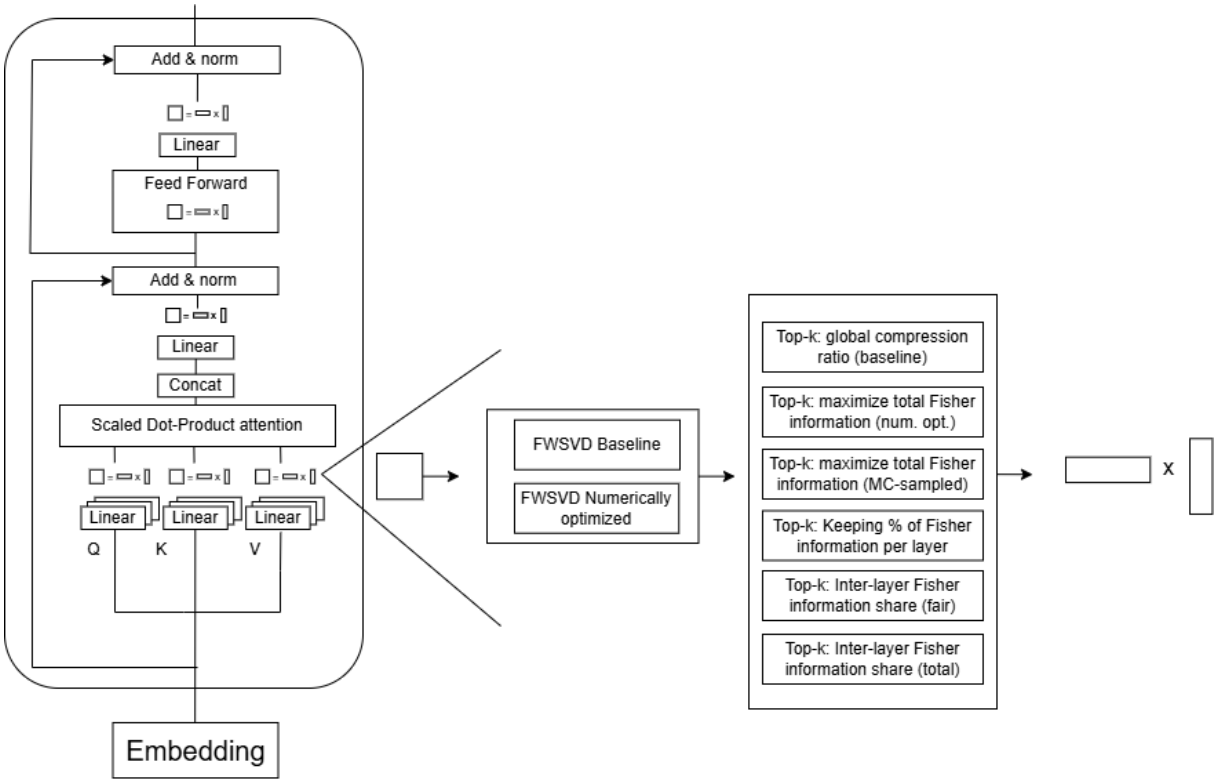


Figure 3: Experimental setup: Encoder-level BERT compression and compression approaches

after which a top-k is determined dynamically by means of the approaches described in the methodology.

All GLUE tasks are used to determine how well the proposed approaches work under the FWSVD and TFWSVD low-rank compression schemes, except for the RTE and WNLI tasks. These tasks are not used for benchmarking as previous work by Hsu et al. (2022) as well as by Hua et al. (2022) do not use these tasks in their benchmarking.

In order to compare the approaches among each other, compression ratios are tweaked such that a specific number of parameters can be reached. For the approaches that have a fixed total compression ratio $c$, a specific compression is obtained by fixing this parameter. These fixed-compression approaches being the inter-layer Fisher information (with loss function $\mathcal{L}_{overall}(\alpha)$) and weight matrix compression, maximizing the observed Fisher information approaches, as elaborated on in the methodology section. Notably, for the latter approach maximizing the observed Fisher information, the compression constraint might not be fully reliable due to the fact that in this approach, the compression is regulated by a Lagrange multiplier in the loss, and hence is 'soft-constrained'.

The total compression to compare with is chosen to be 1.65, as this is the same total compression ratio used in the work by Hsu et al. (2022). Note that 21.3% of the parameters in the model ($\sim$ 23M parameters) are in fact part of the embedding layer of the model, and are not compressed, hence a total compression of 1.65 translates to a compression applied to the encoder- and downstream linear layers that is larger than this.

Furthermore, note that reported performances on the baseline approach by Hsu et al. (2022) and Hua et al. (2022) might not be exactly the same as reported in their respective works. This could be due to slight differences between this work and the work by Hsu et al. (2022) and Hua et al. (2022) with regard to the base BERT model, such as the pre-training weights and slight architecture differences of which this work is unaware. Additionally, the layers compressed might be slightly different between the approaches, such as potentially the compression of linear layers specific to the downstream task, given the nature of the experiment design descriptions of these works do not specify this in great detail. This possible difference does however not affect the validity of this work, as (1) the FWSVD compression method as presented by Hsu et al. (2022) and TFWSVD by Hua et al. (2022) are utilized for the baseline experiment, and form the foundation for experiments on the novel approaches presented in this work and (2) the same network architecture is used for all experiments for a given task. Given the model is constant for all experiments, the relative differences between approaches on the different GLUE tasks are valid.

The benchmark total compression of 1.65 is reported on with a maximum deviation of $\pm 0.3\%$, for which individual methods have been tweaked, which shall be elaborated on below. Note that tweaking might differ slightly per task within the experiment set of each approach, as the size of the task-specific part of the BERT model (the task head) differs slightly between tasks.

The compression approaches that are especially difficult to fix the total number of parameters after compression for are most notably, the method retaining a percentage of the FI for each layer, and to a lesser extent the inter-layer compression ratio approach with loss function $\mathcal{L}_{fair}(\alpha)$, as elaborated on in the methodology section.

## Compression based on FI between layers

Compression based on the Fisher Information between layers has two distinct sub-methods. The first one being the *fair* approach, which optimizes for Fisher-Information based layer compression ratios that keep the average compression ratio $\frac{1}{|Q|}\sum_{W\in Q}\alpha(1-p_W)$ as close as possible to the desired compression ratio $c$, independent of the weight matrix size of said layer. The loss objective for this optimization problem can be seen in equation 13. As previously discussed, a downside of this approach is that the total number of parameters after compression might not after compression be equal to $\frac{S_{total}}{c}$, despite the average of the layer compression ratios being equal to $c$. For example in the case when layers with a smaller weight matrix have been assigned a larger compression ratio, and layers with a larger weight matrix have been assigned a smaller compression ratio. As mentioned, given that it appears harder for this approach to directly fix a compression ratio for the total number of parameters, despite tweaking this approach for each task to have a benchmark value for the total compression ratio of 1.65, the compression ratio for this approach might deviate slightly from the target compression ratio within a margin of $\pm 0.3\%$.

The *overall* approach on the other hand, optimizes for Fisher-Information-based layer compression ratios that keep the total number of parameters after compression $\sum_{W\in Q}\frac{(m_W\times n_W)}{\alpha(1-p_W)}$ as close as possible to the desired number of parameters after compression $\frac{S_{total}}{c}$. The loss function for this optimization problem can be seen in equation 15. As discussed, the downside to this approach might be that given the compression ratio for a layer is also dependent on its size, the effectiveness of the method is compromised by the fact that compression ratios are not only dependent on the Fisher Information.

Unlike the *fair* approach, the *overall* approach does not have the issue of it being difficult to fix a total number of parameters after compression. This effectively means that for this approach, compression ratios for all experiments can be reached more precisely, more easily.

**Compression by optimizing for retaining FI percentage** For the retaining of a specific percentage of the Fisher Information for each layer, as elaborated on in the methodology section, the selected information retaining percentage must be such that the required total compression ratio of 1.65 is reached. As the observed Fisher information depends on the fine-tuning run that precedes the task evaluation, it requires special tweaking of the percentage of information retained, even more so than as described for the *fair* inter-layer information difference-based approach. As before, for each task, the approach is tweaked to obtain a total compression ratio of 1.65, with a margin of $\pm 0.3\%$.

For this approach, it must be noted that when using the TFWSVD low-rank compression by Hua et al. (2022), a slight difference in implementation exists compared to using FWSVD low-rank compression by Hsu et al. (2022) (FWSVD). With TFWSVD, optimization of $A$ and $B$ (as defined in Equation 7) occurs after the projection of information onto $V$ and subsequent weighting of the singular values using the diagonal elements of the projected Fisher information. The approach using FWSVD on the other hand utilizes the simplification that a single row in W shares one importance such that $\widetilde{\mathcal{I}}_{\theta W_i} = \sum_j \widetilde{\mathcal{I}}_{\theta W_i j}$, causing the objective function to be as in Equation 9. As mentioned in the related work section, Hsu et al. (2022) then solve Equation 9 analytically by using standard SVD on $\widetilde{\mathcal{I}}_\theta W$, where $svd(\widetilde{\mathcal{I}}_\theta W) = (U^*, \Sigma^*, V^*)$. This then makes the solution for $A$ and $B$ from Equation 9 be $A = \widetilde{\mathcal{I}}_\theta^{-1} U^*\Sigma^*$ and $B = V^{*T}$.

At the same time, the approach retaining a given percentage of the information projects $\widetilde{\mathcal{I}}_\theta$ onto $V$, which is now $V^*$, such that $\tilde{I}_{W_{projected}} = V^{*^T} diag(\tilde{I}_{W_{reduced}})V^*$, where $\tilde{I}_{W_{reduced}}$ is defined as in Equation 19. Singular values are then weighed using the diagonal elements of the projected Fisher information: $\Sigma^*_{weighed} = \Sigma^* \odot \tilde{I}_{W_{projected}}$. After which a list of sorted indices $l$ is made, containing the indices of weighed singular values in descending order. $k$ is then determined by iterating over the weighted singular values until the threshold $p\%$ of the sum of all weighted singular values has been reached. The top $k$ indices from the list of sorted indices $l$ are then used in order to determine the top-k components in $\Sigma^*$, determining $\tilde{\Sigma}^*$. Note that still, due to the FWSVD low-rank compression method (which uses the simplification that each row shares one Fisher Information value), $A = \widetilde{\mathcal{I}}_\theta^{-1} \times U^* \times \tilde{\Sigma}^*$.

## Compression by maximizing total FI

For the compression by maximizing the total FI, as previously elaborated in the methodology section, two distinct approaches are taken in order to solve this problem, namely utilizing numerical optimization as well as using Monte Carlo sampling.

For the numerical optimization method, the piece-wise loss as shown in Equation 23, and finding an optimal $C \mid c_W \in C, W \in Q$. For the optimization, the Adam (Kingma and Ba 2015) optimizer is used. The learning rate is set to $0.01$ and $1000$ optimization steps are taken. Note that given the constraint on requiring the total number of parameters after compression to be equal to $\frac{1}{c}$, is a soft constraint enforced by the Lagrange multiplier and the total number of parameters after optimization might not necessarily be equal to $\frac{1}{c}$ after optimization. Hence for this approach to optimize for maximum information being kept given a certain overall compression ratio, in order to report on the task performances given the 1.65x experiment compression, specific tweaking of the compression ratio is again required in order to be able to report on the task performances under a

total compression of $1.65 \pm 0.3\%$.

In order to try to mitigate the potential problem of the optimizer getting stuck at a local minimum, which is a reasonable assumption given the problem space as plotted in Figure 2, as well as to have more stringent control on the total compression of the model, Monte Carlo sampling is utilized. 100,000 samples for $C \mid c_W \in C$ are taken, where for each layer $W \in Q$, $c_W$ is sampled from a uniform distribution where $1 \leq c_W \leq 10$. Any sampled set $C \mid c_w \in C$ is considered valid if the total compression $\tilde{c}$ using the sampled compression map $C$ does not differ from the target total compression $c$ by more than 5%, i.e., $\tilde{c} = c \pm 5\%$. Here $\tilde{c}$ is defined as

$$\tilde{c} = \sum_{W \in Q} \frac{\frac{m_W \times n_W}{c_W}}{S_{total}} \mid \tilde{c_W} \in C, W \in Q. \tag{26}$$

As touched upon in the methodology section, it might be difficult to obtain a sampled compression map $C$ which adheres to this criterion. For this reason, similar to the sampling used to plot an example of the problem space, when compression ratio $\tilde{c} \neq c$, all items $c_W \in C$ are scaled by a factor $z = \frac{c}{\tilde{c}}$ meaning that in case $\tilde{c} > c$ (too much compression), this scaling factor $z < 1$, whereas in case $\tilde{c} < c$ (too little compression), this scaling factor $z > 1$. The experiment is run multiple times in order to find a solution such that the total compression is equal to $c \pm 0.3\%$.

## Results and discussion

In this section, results are shown for which compression has been fixed to be 1.65x, causing the resulting compressed model to have a size of 65.5M parameters, with a deviation $\leq \pm 0.3\%$. The section is split up into two parts, one going over the results obtained from experiments carried out with novel approaches op top of FWSVD (Hsu et al. 2022), and the second going over the results obtained from experiments carried out with novel approaches on top of TFWSVD (Hua et al. 2022).

### Novel methods on top of FWSVD

Firstly, results are discussed with experiments of novel approaches on top of FWSVD (Hsu et al. 2022). FWSVD makes use of a simplified Fisher Weighting of the singular value decomposition, the objective function being defined as $\min_{A,B} ||\widetilde{\mathcal{I}_\theta} W - \widetilde{\mathcal{I}_\theta} AB||_2$, which is solved using standard SVD on $\widetilde{\mathcal{I}_\theta} W$, as elaborated on in the background section. These results are shown in a separate section of Table 1. For the FI-based, dynamic SVD compression approaches built on top of the FWSVD baseline work, one can observe that clearly, for all tasks, at least one Fisher information-informed method utilizing intra- or inter-layer information characteristics outperforms the baseline - which appears to be promising.

For the ColA task, the approaches maximizing for total information using both numerical optimization and Monte Carlo simulation as well as the percentage of total information kept all outperform the baseline approach based on Matthew's correlation metric. In this, the approach maximizing total information kept using Monte Carlo simulation outperforms all other methods with about an order of magnitude difference. This appears rather impressive, given this single-sentence task appears rather challenging to solve after compression without fine-tuning. A possible explanation for the Monte Carlo total information maximization method outperforming the others could be that it can more easily find a global optimum for total information kept when compared to the numerical optimization method, which is a theory posed in the methodology section as well.

Next, for the MNLI inference task, all dynamic rank selection approaches outperform the FWSVD baseline in terms of accuracy. Here it must be noted that for all dynamic rank selection approaches with the exception of the approach retaining a fixed percentage of the FI per layer, these do not deviate more from the baseline FWSVD than the maximum margin of error in the number of parameters that the models contain after compression in these experiments: $2 \times 0.3\% = 0.6\%$. On the other hand, the approach retaining a fixed percentage of the FI per layer clearly outperforms all other approaches, performing about twice as well on the task. A possible hypothesis as to why this approach outperforms all others by such a significant amount might be that inter-layer information might be distributed rather evenly i.e. different layers might contain similar amounts of information in total, while the intra-layer information is not i.e. information within a layer might be contained within only a few rows of its weight matrix, hence the approach that focuses on intra-layer information retention could perform better in such cases.

Considering the MRPC task, it can be observed that this similarity task appears rather challenging, where the F-1 score for the total information maximization approaches as well as the intra-layer information retention approach is zero. A possible explanation for the information maximization approaches performing sub-par could be due to certain layers being rather information-dense, while others contain relatively little information in absolute terms, but are still important to the task at hand. At the same time, sub-par performance by the intra-layer FI retention approach could be explained by the specific task having rows within weight matrices be important to the task, while these did not contain enough information to be retained. At the same time, the fair- and overall dynamic compression methods that define compression ratio based on the inter-layer FI both outperform the FWSVD baseline by a difference that is larger than the maximum margin of error in the number of parameters that the models contain after compression in these experiments. The difference in performance between these approaches and the approaches maximizing total information in the compressed network could be due to some normalization effect by utilizing the shares of total information when calculating the compression ratios, although this remains speculation.

Evaluating the QNLI task, an analysis similar to the MNLI task can be made. Again, for this inference task, the approach retaining a percentage of FI per layer greatly outperforms all other approaches, and a similar reasoning could be applied as was done for the MNLI task. Interestingly, the approaches maximizing total information now also show re-

Table 1: Results of FI-based dynamic compression approaches on top of FWSVD, TFWSVD, and BERT$_{base}$ baselines

| Approach | #Param | ColA (MCC) | MNLI (Acc) | MRPC (F1) | QNLI (Acc) | QQP (F1) | SST2 (Acc) | STSB (PCC) | G-Avg |
|---|---|---|---|---|---|---|---|---|---|
| BERT$_{base}$ (Devlin et al. 2019) | 108.3M | 59.81 | 83.79 | 87.74 | 90.68 | 87.45 | 92.55 | 88.68 | 84.4 |
| FWSVD (Hsu et al. 2022) based | | | | | | | | | |
| FWSVD (Hsu et al. 2022) | 65.55M | 0 | 31.73 | 33.61 | 53.19 | 0 | 83.26 | 20.64 | 31.8 |
| Total info kept MC (Ours) | 65.55M | **13.00** | 31.82 | 0 | 59.25 | 0 | 82.00 | 28.77 | 30.7 |
| Total info kept (Ours) | 65.55M | 3.78 | 31.99 | 0 | 54.37 | 0 | 82.34 | 15.88 | 26.9 |
| % Info kept (Ours) | 65.55M | 2.07 | **62.08** | 0 | **70.58** | **66.47** | **84.17** | **38.01** | **46.2** |
| Dynamic (fair) (Ours) | 65.55M | 0 | 31.85 | **36.02** | 53.54 | 0 | 83.72 | 24.79 | 32.8 |
| Dynamic (total) (Ours) | 65.55M | 0 | 31.88 | 35.68 | 53.56 | 0 | 83.83 | 24.86 | 32.8 |
| TFWSVD (Hua et al. 2022) based | | | | | | | | | |
| TFWSVD (Hua et al. 2022) | 65.55M | **33.78** | 44.14 | 0.71 | 60.33 | **74.77** | 85.32 | 34.82 | 47.7 |
| Total info kept (Ours) | 65.55M | 33.69 | 43.78 | 0 | 58.48 | 65.54 | 83.72 | 34.17 | 45.6 |
| Total info kept (Ours) MC | 65.55M | 25.15 | 39.45 | 0 | 60.74 | 73.03 | 84.17 | **62.10** | 49.2 |
| % Info kept (Ours) | 65.55M | 26.31 | **53.60** | **12.38** | **67.69** | **74.77** | 84.86 | 50.00 | **52.8** |
| Dynamic (fair) (Ours) | 65.55M | 33.27 | 44.20 | 0.71 | 60.61 | 74.00 | **85.67** | 33.93 | 47.5 |
| Dynamic (total) (Ours) | 65.55M | 33.27 | 44.23 | 0 | 60.59 | 74.01 | 85.55 | 34.01 | 47.4 |

sults that improve upon the FWSVD baseline, both exceeding the maximum margin of error in the number of parameters of the compressed models. The Monte Carlo sampled version of the total information maximalization outperforms the numerical optimization method, where again it is theorized that it is able more easily find a global optimum for total information kept when compared to the numerical optimization approach when considering the difficult landscape to find a global optimum in shown in Figure 2.

Next, for the QQP task, the F1 score of the approach that keeps a certain percentage of information for each layer clearly outperforms all other methods on this semantic equivalence task, achieving an F1 score of $66.47\%$. The difference with the MRPC task, another similarity task, is rather stark. This shows the perspective that the lower performance of the intra-layer FI retaining approach might not necessarily be worse on similarity tasks fundamentally, but that the low performance on the MRPC task might be attributed to other factors, such as possibly not having enough epochs to estimate the Fisher Information on.

Continuing to the SST2 task, where single sentences are to be sentiment-classified, here one can observe task performances across approaches that are rather similar to each other, compared to the other tasks. At the same time, the approach where a percentage of information is retained again outperforms the baseline by a margin that is greater than the mentioned margin of error in the number of parameters in the experiment.

When considering the STS-B similarity task, both the total information maximization approach using Monte Carlo sampling as well as the approach retaining a specific percentage of intra-layer information, clearly outperform the baseline approach, showing again that the retaining intra-layer information approach can perform well on similarity tasks (where this task is posed as a classification problem, akin to the other similarity tasks, a similarity for a pair being a discrete value between 1 and 5). At the same time, the maximization of total information approach appears to also have an ability to perform well on such similarity tasks, where it must again be noted that the Monte Carlo sampling approach outperforms the method utilizing numerical optimization, possibly due to the difficulty to find good optima in the problem space, as mentioned before.

Overall, considering also the average Glue score (G-Avg) for all approaches experimented with when using the FWSVD baseline, it appears that the dynamic determination of compression ratios based on the share of the total information between layers, the maximalization of total information in the network as well as retaining a certain percentage of the FI within a given layer, are all methods that are able to outperform the FWSVD baseline approach proposed by Hsu et al. (2022). This shows that both utilizing the inter- and intra-layer information characteristics are able to outperform previous work. Retaining a certain percentage of FI within layers appears to perform the best overall, showing a clear ability to perform well on all task types, where it appears to be particularly strong on the inference and similarity tasks (with the exception of MRPC). Dynamic determination of compression ratios based on the share of the total information between layers, furthermore has shown to work rather well on the similarity tasks.

## Novel methods on top of TFWSVD

Next, the results that make use of an improved version of FWSVD (Hsu et al. 2022), TFWSVD, as was introduced by Hua et al. (2022). Here, instead of making use of the simplification that makes a single row in W share a single importance, Equation 8 is directly optimized. These results are again shown in a separate section of Table 1. When considering the results as shown in Table 1 that make use of

TFWSVD (Hua et al. 2022), the current state of the art, one can observe a number of differences with results obtained when using FWSVD (Hsu et al. 2022) as a foundation for the novel approaches.

Firstly, results from experiments based on TFWSVD achieve, on average, higher performance on their respective task metrics, which in itself is expected as TFWSVD was posed as an improvement on FWSVD. It is however considered relevant to determine the differences between using FWSVD or TFWSVD as a base, as novel approaches might be more effective when utilizing each as a base.

For the ColA single sentence task, one can observe that in contrast to the experiments that use FWSVD as a base, here the baseline approach appears to perform best. However, given the maximum margin of error in the number of parameters that the models contain after compression in these experiments, this cannot be concluded given the approach maximizing total information using numerical optimization as well as the dynamic compression approaches based on inter-layer Fisher information obtained similar task metrics. Possibly, the effect of the approaches that worked best for the ColA task when using FWSVD, is mitigated by directly optimizing Equation 8, which does not consider information characteristics in a row-wise manner - something which the intra-layer information retaining approach does. Additionally, the Monte Carlo sample for the experiment maximizing total information might have simply been sub-par by chance, especially considering the performance of the same experiment utilizing numerical optimization.

When considering the MNLI inference task, one can observe that, similarly to when using FWSVD, the only task meaningfully outperforming the TFWSVD baseline approach is the approach retaining a certain percentage of intra-layer information. As mentioned previously, a possible explanation for this is that possibly, inter-layer information might be distributed rather evenly, while the intra-layer information is not, hence the approach that focuses on intra-layer information retention could perform better. Additionally, one must notice the fact that on the MNLI task, the percentage of intra-layer FI retaining approach on top of FWSVD outperforms even the same approach utilizing TFWSVD, which could initially be regarded as an unexpected result. At the same time, knowing that the percentage of inter-layer FI retention approach projects information onto $V$ in a row-wise fashion, and knowing that FWSVD similarly considers FI in a row-wise fashion, this result might not be as surprising as initially thought.

Next, considering the MRPC similarity task when utilizing TFWSVD, it is interesting to note that where the baseline approach as well as the dynamic compression approaches determining compression ratios based on inter-layer information characteristics associated with layers (both fair and overall), performed best on the MRPC task when using FWSVD, all approaches other than retaining a percentage of intra-layer information appear to have near-zero performance. While this approach which is now the exception, it had for the FWSVD-based experiment a near-zero performance itself. A possible explanation for this could be that the solution found by the TFWSVD optimizer is sub-par in itself and that the row-wise information projection onto $V$ has been successful in mitigating this, although this remains up for debate.

Considering the QNLI inference task, it is interesting to observe that when using FWSVD both the information maximization approaches as well as the intra-layer information retention approach all outperformed the baseline by more than the maximum margin of error in the number of parameters that the models contain after compression when using, now, when utilizing TFWSVD, only the intra-layer information retention approach meaningfully improves upon the TFWSVD baseline. This might again be due to the possibility that performance gains using the proposed total information maximization method due to row-wise information in FWSVD causing information to possibly be more spread over a parameter matrix, being mitigated by the direct optimization of Equation 8. While the intra-layer information share retention approach remains the best-performing approach, as with the MNLI inference task, this approach performed even better when utilizing FWSVD as a base for the method, possibly due to this approach projecting information onto $V$ in a row-wise fashion, given FWSVD considers FI in a row-wise fashion as was previously mentioned.

Considering the QQP task, in great contrast to the experiments utilizing FWSVD, the TFWSVD baseline is not outperformed by any dynamic compression approach. Therefore, it might be simply possible that for a given number of parameters, a more optimal solution than the solution posed by TFWSVD cannot be achieved, which could be supported by the fact that the approach retaining a percentage of the intra-layer FI performed exactly as well as TFWSVD, while this approach often outperforms the baseline - also on similarity tasks such as QQP.

Next, the SST single sentence task shows the dynamic compression method utilizing the inter-layer shares of the total information performing best. However, this performance cannot be determined to be definitively better than the baseline experiment, as the improvement does not exceed the maximum margin of error in the number of parameters that the models contain after compression. Similarly to the ColA single sentence task, possibly, the effect of the intra-layer FI share retention approach when using FWSVD, is mitigated by directly optimizing Equation 8, which does not consider information in a row-manner - something which the intra-layer information retaining approach does.

Lastly, considering the STSB similarity task, both the approach maximizing total information using Monte Carlo sampling of compression ratios, as well as the intra-layer FI retention approach, outperform the TFWSVD baseline, such as was the case for the FWSVD-based experiments. Again, the Monte Carlo approach for total information maximization outperformed the numerical optimization, given the difficult-to-optimize problem landscape seen in Figure 2.

Overall, considering the average Glue score (G-Avg) for all approaches experimented with when using the TFWSVD baseline, it appears that both maximizing the total information kept under a given compression constraint, compression ratios being Monte Carlo sampled, as well as retaining a certain percentage of the FI within a given layer, are both

methods that can outperform the TFWSVD SOTA baseline approach proposed by Hua et al. (2022). For the approach retaining a certain percentage of the FI for each layer, utilising the inta-layer information characteristics, outperforming the baseline is a result that was also found when utilizing FWSVD by Hsu et al. (2022). The magnitude of outperforming the baseline is however higher when utilizing the the FWSVD baseline approach, which as mentioned before is theorized to be due to the approach retaining a percentage of inter-layer FI projecting information onto $V$ in a row-wise fashion, and the FWSVD approach considering FI in a row-wise fashion as well, in contrast to TFWSVD.

### Overall across both experiment sets

The clearly best-performing method is keeping a specified percentage of information on an intra-layer basis. This is the only approach that determines the compression ratio not on the total information for a layer but determines the compression ratio for a layer based on the individual row-wise information in said layer, iteratively keeping rows until the information threshold is reached. This approach has been shown to outperform the baseline in all but one task when using FWSVD, and for the SOTA TFWSVD improving on all of the inference tasks, two out of three of the similarity tasks (the last one being equal to SOTA), and only having one of two single sentence tasks be more than maximum margin of error in the number of parameters that the models contain after compression away from the SOTA approach.

While still outperforming the SOTA on most tasks, the performance improvement from the approach retaining a percentage of intra-layer information is less drastic when using TFWSVD when compared to FWSVD. This could be due to the fact that in contrast to TFWSVD, FWSVD considers FI in a row-wise fashion, just as the approach retaining a percentage of intra-layer FI projects information onto $V$ in a row-wise fashion. Additionally, this could explain the performance below the SOTA level for the ColA task when using the TFWSVD baseline as well as the intra-layer information retention approach on top of FWSVD at times improving on the SOTA. Even more so than when using the the intra-layer information retention approach on top of the SOTA approach (TFWSVD).

## Future work

This work has shown that both taking into consideration the inter- and intra-layer information characteristics when using SVD to compress a language transformer network, are greatly beneficial to task performance, outperforming the state of the art on almost all GLUE tasks. This work therefore acts as a precursor for additional research into utilizing inter- and intra-layer information characteristics in order to optimise layer compression in transformers - for example, using metrics that are different from the Fisher information. At the same time, this work can be improved upon by extending results to new modalities, such as vision transformers, combining inter- and intra-layer approaches as well as finding a way around the row-wise constraint currently present for the approach which considers the intra-layer information characteristics by means of row-wise information retention.

## Conclusion

This work has introduced a novel approach to Fisher-weighted Singular Value Decomposition (SVD) for low-rank compression, addressing limitations in previous methods by dynamically determining layer-wise compression ratios based on intra- and inter-layer Fisher Information (FI). Unlike previous methods, our approach enhances task performance efficiency through three dynamic rank determination methods. One of the proposed methods, relying on specifying a fixed percentage of Fisher information to keep per layer, has been shown to outperform the current state of the art in excess of 5% on average, and outperform the current state of the art very significantly on inference and similarity tasks. The work furthermore provides valuable insights for future work to further explore the dynamic compression of layers in transformer networks using Fisher Information characteristics, in particular by displaying the effectiveness of dynamic compression using intra-layer Fisher information.

## References


Ba, L. J.; Kiros, J. R.; and Hinton, G. E. 2016. Layer Normalization. *CoRR*, abs/1607.06450.

Bengio, Y.; Simard, P.; and Frasconi, P. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2): 157–166.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In Màrquez, L.; Callison-Burch, C.; Su, J.; Pighin, D.; and Marton, Y., eds., *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 632–642. The Association for Computational Linguistics.

Cao, Q.; Trivedi, H.; Balasubramanian, A.; and Balasubramanian, N. 2020. DeFormer: Decomposing Pre-trained Transformers for Faster Question Answering. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 4487–4497. Association for Computational Linguistics.

Cer, D. M.; Diab, M. T.; Agirre, E.; Lopez-Gazpio, I.; and Specia, L. 2017. SemEval-2017 Task 1: Semantic Textual Similarity - Multilingual and Cross-lingual Focused Evaluation. *CoRR*, abs/1708.00055.

Chen, P.; Si, S.; Li, Y.; Chelba, C.; and Hsieh, C.-J. 2018. Groupreduce: Block-wise low-rank approximation for neural language model shrinking. *Advances in Neural Information Processing Systems*, 31.

Choi, H.; Kim, J.; Joe, S.; and Gwon, Y. 2021. Evaluation of bert and albert sentence embedding performance on downstream nlp tasks. In *2020 25th International conference on pattern recognition (ICPR)*, 5482–5487. IEEE.

Dehghani, M.; Djolonga, J.; Mustafa, B.; Padlewski, P.; Heek, J.; Gilmer, J.; Steiner, A. P.; Caron, M.; Geirhos, R.; Alabdulmohsin, I.; et al. 2023. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, 7480–7512. PMLR.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Denton, E. L.; Zaremba, W.; Bruna, J.; LeCun, Y.; and Fergus, R. 2014. Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in neural information processing systems*, 27.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Dolan, B.; and Brockett, C. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *CoRR*, abs/2010.11929.

Hajimolahoseini, H.; Ahmed, W.; Rezagholizadeh, M.; Partovinia, V.; and Liu, Y. 2022. Strategies for applying low rank decomposition to transformer-based models. In *36th Conference on Neural Information Processing Systems (NeurIPS2022)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *CoRR*, abs/1503.02531.

Hsu, Y.-C.; Hua, T.; Chang, S.; Lou, Q.; Shen, Y.; and Jin, H. 2022. Language model compression with weighted low-rank factorization. In *International Conference on Learning Representations*.

Hua, T.; Hsu, Y.; Wang, F.; Lou, Q.; Shen, Y.; and Jin, H. 2022. Numerical Optimizations for Weighted Low-rank Estimation on Language Models. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 1404–1416. Association for Computational Linguistics.

Jaderberg, M.; Vedaldi, A.; and Zisserman, A. 2014. Speeding up Convolutional Neural Networks with Low Rank Expansions. In Valstar, M. F.; French, A. P.; and Pridmore, T. P., eds., *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*. BMVA Press.

Khetan, A.; and Karnin, Z. S. 2020. schuBERT: Optimizing Elements of BERT. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2807–2818. Association for Computational Linguistics.

Khodak, M.; Tenenholtz, N. A.; Mackey, L.; and Fusi, N. 2021. Initialization and Regularization of Factorized Neural Layers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Kiefer, J.; and Wolfowitz, J. 1952. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 462–466.

Kim, H.; Khan, M. U. K.; and Kyung, C.-M. 2019. Efficient neural network compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12569–12577.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Lebedev, V.; Ganin, Y.; Rakhuba, M.; Oseledets, I. V.; and Lempitsky, V. S. 2015. Speeding-up Convolutional Neural Networks Using Fine-tuned CP-Decomposition. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Levesque, H.; Davis, E.; and Morgenstern, L. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Lyu, Z.; Yu, T.; Pan, F.; Zhang, Y.; Luo, J.; Zhang, D.; Chen, Y.; Zhang, B.; and Li, G. 2023. A survey of model compression strategies for object detection. *Multimedia Tools and Applications*, 1–72.

Matthews, B. W. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2): 442–451.

Nakajima, S.; Sugiyama, M.; Babacan, S. D.; and Tomioka, R. 2013. Global analytic solution of fully-observed variational Bayesian matrix factorization. *The Journal of Machine Learning Research*, 14(1): 1–37.

Pascanu, R.; Mikolov, T.; and Bengio, Y. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, 1310–1318. Pmlr.

Prabhavalkar, R.; Alsharif, O.; Bruguier, A.; and McGraw, L. 2016. On the compression of recurrent neural networks with an application to LVCSR acoustic modeling for embedded speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5970–5974. IEEE.

Quora. 2012. First Quora Dataset Release Question Pairs.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In Su, J.; Carreras, X.; and Duh, K., eds., *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2383–2392. The Association for Computational Linguistics.

Sainath, T. N.; Kingsbury, B.; Sindhwani, V.; Arisoy, E.; and Ramabhadran, B. 2013. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *2013 IEEE international conference on acoustics, speech and signal processing*, 6655–6659. IEEE.

Sanh, V.; Wolf, T.; and Rush, A. 2020. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in neural information processing systems*, 33: 20378–20389.

Schotthöfer, S.; Zangrando, E.; Kusch, J.; Ceruti, G.; and Tudisco, F. 2022. Low-rank lottery tickets: finding efficient low-rank neural networks via matrix differential equations. *Advances in Neural Information Processing Systems*, 35: 20051–20063.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Theis, L.; Korshunova, I.; Tejani, A.; and Huszár, F. 2018. Faster gaze prediction with dense networks and Fisher pruning. *CoRR*, abs/1801.05787.

Tu, M.; Berisha, V.; Woolf, M.; Seo, J.-s.; and Cao, Y. 2016. Ranking the parameters of deep neural networks using the fisher information. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2647–2651. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Wang, J.; Li, S.; and Wang, W. 2019. SVD-based channel pruning for convolutional neural network in acoustic scene classification model. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 390–395. IEEE.

Wang, N.; Liu, C.-C. C.; Venkataramani, S.; Sen, S.; Chen, C.-Y.; El Maghraoui, K.; Srinivasan, V. V.; and Chang, L. 2022. Deep compression of pre-trained transformer models. *Advances in Neural Information Processing Systems*, 35: 14140–14154.

Warstadt, A.; Singh, A.; and Bowman, S. R. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7: 625–641.

Williams, A.; Nangia, N.; and Bowman, S. R. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In Walker, M. A.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, 1112–1122. Association for Computational Linguistics.

Wold, S.; Esbensen, K.; and Geladi, P. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3): 37–52.

## Additional background

### The transformer

Transformers, as proposed first by Vaswani et al. (2017), are a type of network architecture which is completely based on attention mechanisms and are much faster to train than conventional network architectures. The transformer network architecture (Vaswani et al. 2017) is shown in Figure 4.

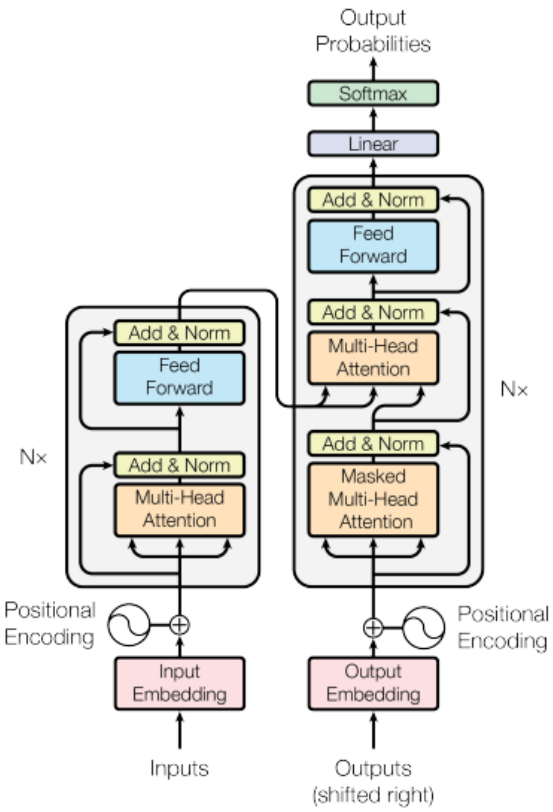


Figure 4: Transformer model architecture, as shown in (Vaswani et al. 2017).

The transformer architecture consists of positional encodings of size $d_{model}$ being added to input or output embeddings which are also of size $d_{model}$, which is done to add information about the position of a word in the sequence.

The distinct blocks repeated N times, shown in Figure 4 are the encoder- and decoder stacks. The encoder stack consists of N layers, where each of these identical layers consists of a multi-head self-attention process as well as a simple feed-forward network. Additionally, a residual connection (also often referred to as a skip-connection), as initially proposed by He et al. (2015), is added to mitigate the vanishing gradient problem (Bengio, Simard, and Frasconi 1994; He et al. 2015; Pascanu, Mikolov, and Bengio 2013). This has the added effect that using this skip-connection keeps layer-local information intact given the self-attention process can perform computations that do not necessarily preserve any information from the input data. Lastly, both the multi-head attention process as well as the simple feed-forward network are followed by layer normalization (Ba, Kiros, and Hinton 2016).

Similar to the encoder, the decoder stack also consists of N identical layers, whereas for the decoder, the multi-head self-attention process is masked such that it is enforced that only previous tokens are considered when predicting the next token. Additionally, output embeddings are offset by one position for the same purpose.

An additional multi-head attention process is added to the decoder, which takes the output of the encoder stack as input. Akin to the encoder, a simple feed-forward network is present, and around each multi-head attention block as well as the feed-forward network, a skip connection is present followed by normalization.

### Attention

The attention mechanisms utilized in the transformer architecture is defined as a function which maps a query, keys and values to an output that is a sum of the values weighted by the dot product between the query and the key corresponding to the value:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (27)$$

In this context, $Q$ (query) is derived from the decoder hidden state and represents a linear transformation applied to the word embedding of a specific word in a sentence. Similarly, $K$ (key) results from a different linear transformation applied to the embedding of another word in the same sentence. The dot product $QK^T$ measures the similarity between these two words within the given context.

A value ($V$) here is the result of a separate linear transformation applied to the scored dot product of the query ($Q$) and a key ($K$). This linear transformation is optimized to find the next word in a sequence.

An enumeration summarizing this can be seen below.

- **Query (Q)**: Derived from the decoder hidden state; a linear transformation applied to the word embedding of a specific word in a sentence.
- **Key (K)**: Derived from a different linear transformation applied to the embedding of another word in the same sentence as the query.
- **Value (V)**: Result of a separate linear transformation applied to the scored dot product of the query and key.
- The value transformation is optimized to predict the next word in a sequence.
- The dot product of the query and key represents the similarity between the two words in a given context.

An example figure of this concept can be seen in Figure 5.

### Transformer compressability

For this work, it is important to note the compressibility of each of these components of the transformer, where we utilize the fact that linear layers are excellent for weight matrix compression using Singular Value Decomposition.

Given that the keys, values and queries are merely linear transformations applied to word embeddings, these are linear layers and hence lend themselves well to compression. Additionally, the intermediary- and output feed-forward layers in the transformer architecture as shown in Figure 4, are linear as well, and hence also lend themselves to compression.

## Proofs and derivations

### Derivation of the analytical solution to Equation 13

The derivation of the analytical solution to Equation 13 is as follows.

$$\left(c - \frac{1}{|Q|}\sum_{W\in Q}\alpha(1-p_W)\right)^2 = \left(c - \alpha\frac{1}{|Q|}(|Q|-1)\right)^2 = \left(c - \alpha(1-\frac{1}{|Q|})\right)^2 \quad (28)$$

As $\sum_{W\in Q} p_W = 1$. Setting Equation 28 equal to 0 gives

$$c - \alpha(1-\frac{1}{|Q|}) = 0 \quad (29)$$

$$c = \alpha(1-\frac{1}{|Q|}) \quad (30)$$

$$\alpha = \frac{c}{(1-\frac{1}{|Q|})} \quad (31)$$

### Derivation of the analytical solution to Equation 15

The derivation of the analytical solution to Equation 15 is as follows.

$$\left(\frac{S_{total}}{c} - \sum_{W\in Q}\frac{(m_W \times n_W)}{\alpha(1-p_W)}\right)^2 = \left(\frac{S_{total}}{c} - \frac{1}{\alpha}\sum_{W\in Q}\frac{(m_W \times n_W)}{(1-p_W)}\right)^2 \quad (32)$$

Setting Equation 32 equal to 0 gives

$$\frac{S_{total}}{c} - \frac{1}{\alpha}\sum_{W\in Q}\frac{(m_W \times n_W)}{(1-p_W)} = 0 \quad (33)$$

$$\frac{S_{total}}{c} = \frac{1}{\alpha}\sum_{W\in Q}\frac{(m_W \times n_W)}{(1-p_W)} \quad (34)$$

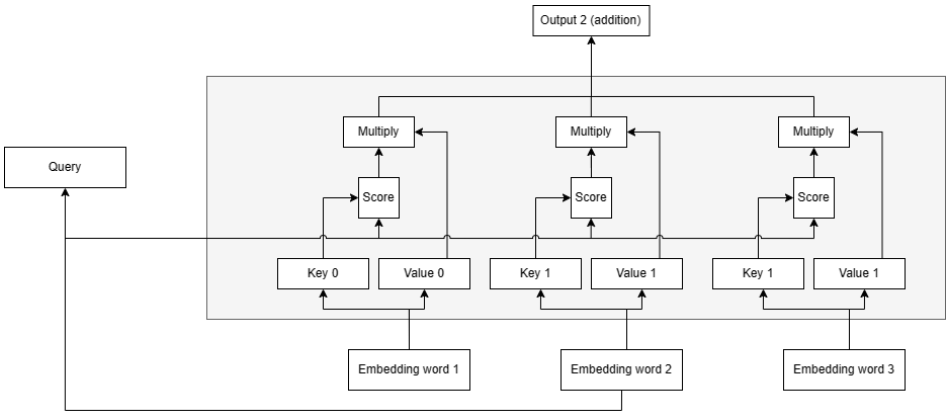


Figure 5: Self-attention concept.

$$\alpha \frac{S_{total}}{c} = \sum_{W \in Q} \frac{(m_W \times n_W)}{(1 - p_W)} \tag{35}$$

$$\alpha = \frac{c}{S_{total}} \sum_{W \in Q} \frac{(m_W \times n_W)}{(1 - p_W)} \tag{36}$$