Addressing Token Uniformity in Transformers via Singular Value Transformation (Supplementary Material)

Hanqi Yan  Lin Gui  Wenjie Li  Yulan He

1Department of Computer Science, University of Warwick, United Kingdom
2Department of Computing, The Hong Kong Polytechnic University, China
3The Alan Turing Institute, United Kingdom

A PROOF OF THEOREM IN SECTION 3

Theorem: \( \forall x \in X^l, \exists x' \in S_{[1,k]}^l \), where the subspace \( S_{[1,k]}^l \) is defined based on \( \lambda_k \geq C \geq \lambda_{k+1} \), then \( \|x - x'\|_2 \leq C \).

Proof We assume that \( X^l \) can be represented as a \( n_l \times m \) matrix:

\[
X^l = \begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2 \\
\vdots \\
\tilde{x}_{n_l}
\end{bmatrix},
\]

where \( \tilde{x}_i \in \mathbb{R}^m \) is an \( m \)-dimensional embedding of a token in the output of \( l \)-th layer. After performing SVD on \( X^l \), we have:

\[
\begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2 \\
\vdots \\
\tilde{x}_{n_l}
\end{bmatrix} = \begin{bmatrix}
\tilde{u}_1 \\
\tilde{u}_2 \\
\vdots \\
\tilde{u}_{n_l}
\end{bmatrix} \begin{bmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \lambda_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_m
\end{bmatrix} \begin{bmatrix}
\tilde{v}_1 \\
\tilde{v}_2 \\
\vdots \\
\tilde{v}_{n_l}
\end{bmatrix},
\]

where the unitary matrix \( U = [\tilde{u}_1^T, \tilde{u}_2^T, \ldots, \tilde{u}_{n_l}^T]^T \), \( V = [\tilde{v}_1^T, \tilde{v}_2^T, \ldots, \tilde{v}_{n_l}^T]^T \) are \( n_l \times n_l \) left singular matrix and \( m \times m \) right singular matrix, respectively. Therefore, the two collections of vectors, i.e. \( \tilde{u}_i = \{u_{i1}, u_{i2}, \ldots, u_{im}\} \) and \( \tilde{v}_i = \{v_{i1}, v_{i2}, \ldots, v_{im}\} \), are two subsets of basis for the \( m \)-dimensional vector space \( \langle m \ll n_l \rangle \). Without loss of generality, we assume \( x_i \in X^l \) can be represented by its corresponding left singular vector, singular values, and the right singular matrix \( V \), which yields:

\[
\tilde{x}_i = \tilde{u}_i \cdot \begin{bmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \lambda_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_m
\end{bmatrix} \cdot \begin{bmatrix}
\tilde{v}_1 \\
\tilde{v}_2 \\
\vdots \\
\tilde{v}_{n_l}
\end{bmatrix}
\]

If we separate the singular values into two parts by \( \lambda_k \geq C \geq \lambda_{k+1} \geq 0 \), we can rewrite Eq. [1] by:

\[
\tilde{x}_i = \sum_{j=1}^{m} \lambda_j \cdot u_{ij} \cdot \tilde{v}_j
\]

By defining \( \tilde{x}_i' = \sum_{j=1}^{k} \lambda_j \cdot u_{ij} \cdot \tilde{v}_j \), where singular values are taken from the larger group, we have:

\[
||\tilde{x}_i - \tilde{x}_i'|| = ||\sum_{j=k+1}^{m} \lambda_j \cdot u_{ij} \cdot \tilde{v}_j||
\]

Where \( || \cdot || \) is the norm, \( \otimes \) is the pairwise product, and \( \langle \cdot, \cdot \rangle \) is the inner product in a vector space, \( \lambda^{[k+1,m]} \) and \( \tilde{u}_i^{[k+1,m]} \) are the sub-vectors of singular values and \( \tilde{u}_i \) from \( k+1 \)-th to \( m \)-th dimensions, respectively, and \( V^{(m-k-1)\times m} \) is the corresponding right singular sub-matrix. According to Hölder inequality, we have:

\[
||\tilde{x}_i - \tilde{x}_i'|| \leq ||\lambda^{[k+1,m]} \otimes \tilde{u}_i^{[k+1,m]}|| \cdot ||V^{(m-k-1)\times m}||
\]

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).
Since $V$ is a unitary matrix, $V^T \cdot V = I$, which yields $\|V_{(m-k)\times m}\| = 1$. Hence,
\[
\|\vec{x}_i - \vec{x}_i'\| \leq \sqrt{\Sigma_{j=k+1}^m \lambda_j^2 \cdot u_{ij}^2}
\]

Considering $\|\vec{u}\| = 1$ and $\lambda_{k+1} \leq C$, obviously we have $\|\vec{u}_{[k+1,m]}\| \leq 1$ and $\lambda_j \leq C$, when $j \geq k + 1$. Therefore,
\[
\|\vec{x}_i - \vec{x}_i'\| \leq C \cdot \sqrt{\Sigma_{j=k+1}^m u_{ij}^2} \leq C
\]

**A case study where the vectors in the unitary matrix $U$ follows a uniform distribution in a $L_2$-norm based metric space**

The **theorem** states that the learned features from a transformer-based language model can be represented as a closure which is defined as a $C$-neighbour of a $k$-dimensional space. Here, we present a case study, assuming the vectors $\vec{u}_i$ in the unitary matrix $U$ follows a uniform distribution within a $L_2$-norm based metric space.

Under such an assumption, the probability of $P(\Sigma_{j=k+1}^m u_{ij}^2 \leq d)$ is the integral of the probability density function in the corresponding area of an $n$-sphere, denoted as $S_{n-1}$, defined by $\Sigma_{j=k+1}^m u_{ij}^2$. It is clear that $P(\Sigma_{j=k+1}^m u_{ij}^2 \leq d) \geq 0$. Hence, we only discuss the upper boundary of $P$ in the following. We denote the sub-area of $\Sigma_{j=k+1}^m u_{ij}^2 \leq d$ as $S_\phi$. To simplify the notation, without loss of generality, we re-order the elements in $\vec{u}_i \in \mathbb{R}^n$ such that its last $k$ dimensions correspond to the small singular values. Then, we have

\[
P(\vec{u}_i \in S_\phi) = \int_{S_\phi} \frac{\Gamma(n/2)}{2\pi^{n/2}} \Pi_{i=1}^{n-k} \sin^{n-1-i}(\psi_i) d\psi_1 \ldots d\psi_{(n-1)}
\]

\[
\leq \frac{\Gamma(n/2)}{2\pi^{n/2}} \int_{S_\phi} \Pi_{i=1}^{n-k} \sin^{n-k-i}(\psi_i) d\psi_1 \ldots d\psi_{(n-k-1)}
\]

\[
\leq \frac{\Gamma(n/2)}{2\pi^{n/2}} \frac{2^{-(n-k)/2}}{\Gamma((n-k)/2 + 1)} \cdot \frac{2^{k/2}}{\Gamma(k/2 + 1)} \cdot (1 - d^{-(n-k)}) \cdot d^{2k}
\]

\[
\leq \frac{2}{k(n-k)} \cdot B(k/2, (n-k)/2) \cdot d^{2k} (1 - d^{-(n-k)})
\]

\[\text{(2)}\]

where $B(\cdot, \cdot)$ is the beta function. The result show that the probability of a singular vector residing in the sub-area $S_\phi$ will converge to 0 exponentially with the growth of $k$. As such, when $k$, the number of smaller singular vectors, is large, the distance between the embedding space and the subspace spanned by the larger singular vectors is bounded by $C$, the smallest value in the larger singular value group.

**B MODEL CONFIGURATIONS AND TRAINING DETAILS**

**Unsupervised Setting** In the unsupervised setting on the STS task, we use the datasets processed by [Huang et al., 2021] and follow their evaluation pipeline by replacing their Whitening function with our SoftDecay function in their released code. We do not use any dataset to train the transformation function, instead, we choose a fixed $\alpha$ empirically ($\alpha$ is the hyper-parameter in Eq.(3)). As we did not see significant changes across different $\alpha$, we set $\alpha$ to $-0.6$ for all the datasets and PTLMs. For metrics calculation, we use $t = 0.5$ in RBF$_{dis}$ and we choose the nearest 12 points to reconstruct the query point in LDS.

**Supervised Setting** We apply SoftDecay to the output of the last layer of a PTLM provide by huggingface, before layer normalisation. We use the default parameters configured in BERT-base-uncased [ALBERT-base, RoBERTa-base, and DistilBERT-base-uncased] as the baselines. For hyper-parameter setting, we search the initial alpha for different datasets from $[-0.2, -0.5, -0.8]$, and set different learning rates from $[2e - 3, 2e - 5]$ for the transformation layer and the pretrained models.

**C ADDITIONAL RESULTS ON SEMANTIC TEXTUAL SIMILARITY DATASET**

In this section, we first examine the potential reasons of improvement by comparing the learnt representations from baselines models (i.e., vanilla PLTs and WhiteningBERT) and our proposed SoftDecay through quantitative evaluation results and the visualisation results (See in §5.1 and §5.2). We then discuss a comparison between SoftDecay and a representative contrastive learning method, SimCSE [Gao et al., 2021], which also aims to...
Table 1: Uniformity metrics (EVs, TokenUni, RBF\textsubscript{dis}) evaluates the isotropy in transformed feature space comparing to the vanilla PTLMs features. Smaller values means the features are better uniformly distributed. It can be seen that SoftDecay can greatly improve the uniformity.

### C.1 FEATURE EVALUATION RESULTS ON STS DATASETS

We show in Table 1 and Figure 1 both the uniformity and local neighborhood preservation evaluation results of different methods over the seven STS datasets. The lower scores returned by SoftDecay in Table 1 in comparison to the base PTLMs verify its capability of alleviating anisotropic feature space derived from BERT. In Figure 1, SoftDecay preserves the local neighbourhood structure better among all the datasets, which explains its performance superiority comparing with Whitening which ignores the original local manifold structure.

### C.2 VISUALISATION OF FEATURES IN STS DATASETS

We show the representations of sentence pairs generated from BERT, with Whitening and with SoftDecay via tSNE for the rest five STS datasets in Figure 2. In STSB, STS13 and STS16, the representation mapping results in Whitening are not unit Gaussian due to some abnormal data point. Our proposed method SoftDecay gives better uniformity score than vanilla BERT and better \textit{LSDS} than WhiteningBERT, as have been shown in Figure 1 and Table 1.

### C.3 COMPARISON WITH CONTRASTIVE LEARNING ON STS

The objective of contrastive learning methods is to align semantically-related positive data pairs and make the learned representations evenly distributed in the resulting embedding space [Wang and Isola, 2020]. The latter property naturally addresses the token uniformity issue. Therefore, we further compare Softdecay with a representative contrastive learning method, SimCSE [Gao et al., 2021], on STS. As SimCSE needs to be trained on datasets to fine-tune its parameters, we conduct experiments using SimCSE fol-
Figure 2: The tSNE visualisation of representations of sentence pairs in datasets SICKR, STSB, STS12-16 (except STS15) in different columns. These representations from top to bottom are derived from vanilla BERT, BERT+whitening and BERT+SoftDecay. For each sentence pair, the two sentences are denoted by different colors, e.g., black and red in BERT. We can see clear clusters in BERT and BERT+SoftDecay for STS-B, STS-12 and STS-14 datasets.

Following its original setup: (1) **Unsupervised.** Train the model on sampled 1 million sentences from English Wikipedia[^1] and pass the same sentence twice to a pre-trained encoder with standard dropout to generate two different sentence embeddings as positive pairs. Other sentences in the same mini-batch are taken as negative pairs; (2) **Supervised.** Train the model on natural language inference datasets, MNLI and SNLI[^2] and use the annotated entailment and contradictory pairs as positive and negative sentence pairs, respectively. The results are shown in Table 2. It can be observed that SoftDecay outperforms SimCSE in general, especially under the supervised setting. The end goal of our approach (via increasing the weights of small singular values in the output embedding space) is similar to SimCSE (via random dropout masks) under the unsupervised setting, as both aim to learn an isotropic embedding distribution. However, in the supervised SimCSE, its contrastive loss is calculated on a subset of training pairs, as such, it is relatively difficult to achieve the universal isotropy, which is not the case in our approach.

**D ADDITIONAL RESULTS ON GLUE DATASETS**

In this section, we first show the results of comparing SoftDecay with another method, which applies regularisation during training to alleviate the anisotropy issue. Then, we display the Cumulative distribution function (CDF) of singular value distributions before and after applying SoftDecay.

**D.1 COMPARING WITH ANOTHER SINGULAR VALUE TRANSFORMATION FUNCTION**

In addition to Sentence-BERT (S-BERT for short) [Reimers and Gurevych, 2019] and BERT-CT [Carlsson et al., 2021], we also compare with another method which applies regularisation on the output embedding matrix with an exponentially decayed singular value prior distribution during training.

[^1]: Download link for Sampled English Wikipedia dataset
[^2]: Download link for the combined NLI dataset
<table>
<thead>
<tr>
<th>Model</th>
<th>STSB</th>
<th>STS-12</th>
<th>STS-13</th>
<th>STS-14</th>
<th>STS-15</th>
<th>STS-16</th>
<th>SICK-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trained on wiki-text (unsupervised)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimCSE [Goyal et al., 2020]</td>
<td>74.48</td>
<td>66.01</td>
<td>81.48</td>
<td>71.77</td>
<td>77.55</td>
<td>76.53</td>
<td>69.36</td>
</tr>
<tr>
<td>SoftDecay</td>
<td>75.81</td>
<td>63.25</td>
<td>78.67</td>
<td>70.41</td>
<td>79.37</td>
<td>77.69</td>
<td>71.15</td>
</tr>
<tr>
<td>Trained on MNLI and SNLI dataset (supervised)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimCSE [Goyal et al., 2020]</td>
<td>82.26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SoftDecay</td>
<td>83.51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Comparison with contrastive learning method, SimCSE. Our methods demonstrate overall better results under the supervised setting.

**D.2 SINGULAR VALUE DISTRIBUTION**

**The effects of dataset size on NLI dataset** We highlight the different singular value distribution in QNLI and RTE, two datasets for language inference task (See in Figure 3).

**BERT-Based Model Results** For BERT-based model, we show the CDF of singular values on all the evaluated datasets in Figure 4. We observe that by applying SoftDecay (bottom row of Figure 4), the CDF of singular values in the last layer becomes more flattened compared to that in vanilla BERT (top row of Figure 4).

**ALBERT-Based and DistilBERT-based Model Results** We also show the results for ALBERT (Figure 5 and Figure 6) and DistilBERT (Figure 7 and Figure 8). By comparing with the vanilla PTLMs (the top row of each figure), we notice that the application of SoftDecay has a larger impact on ALBERT compared to DistilBERT, especially on the CoLA dataset. For DistilBERT, its feature space becomes anisotropic gradually as layers go deeper.
Figure 3: The CDF of singular value in QNLI (left) and RTE (right) dataset derived from vanilla BERT. For the same percentage 0.8, the larger dataset QNLI dataset has smaller $\Delta L_i$ among all the layers, refers to a more serious token uniformity issue.

Figure 4: Cumulative distribution function (CDF) of singular value distributions. The upper ones are from vanilla BERT, bottom ones are from BERT+SoftDecay. From left to right, the evaluation datasets are SST-2, MRPC, QNLI and CoLA. Different curves represent distributions derived from different model layers. The x-axis represents the normalised singular values sorted in an ascending order. SoftDecay adjusts the anisotropy of the feature space with the effect more noticeable in MRPC and less obvious in QNLI.
Figure 5: CDF of SST-2, MRPC and QNLI datasets. The upper row results are from the vanilla ALBERT, the bottom ones are from ALBERT+SoftDecay.

Figure 6: CDF of CoLA and RTE datasets. The upper row results are from the vanilla ALBERT, the bottom ones are from ALBERT+SoftDecay.
Figure 7: CDF of SST-2, MRPC and QNLI datasets. The upper row results are from the vanilla DistilBERT, the bottom ones are from DistilBERT+SoftDecay.

Figure 8: CDF of CoLA and RTE datasets. The upper row results are from the vanilla DistilBERT, the bottom ones are from DistilBERT+SoftDecay.
References


