Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Feature structure distillation with Centered Kernel Alignment in BERT transferring

Hee-Jun Jung [a], Doyeon Kim [a], Seung-Hoon Na [b], Kangil Kim [a,*]

[a] AI graduate school of Gwangju Institute of Science and Technology (GIST), 123, Cheomdangwagi-ro, Buk-gu, Gwangju, 61005, South Korea
[b] Department of Computer Science and Engineering, Jeonbook National University, 567, Baekje-daero, Deokjin-gu, Jeonju-si, Jeollabuk-do, 54896, South Korea

ARTICLE INFO

Keywords:
Knowledge distillation
BERT
Centered Kernel Alignment
Natural language processing

ABSTRACT

Knowledge distillation is an approach to transfer information on representations from a teacher to a student by reducing their difference. A challenge of this approach is to reduce the flexibility of the student's representations inducing inaccurate learning of the teacher's knowledge. To resolve the problems, we propose a novel method *feature structure distillation* that elaborates information on structures of features into three types for transferring, and implements them based on Centered Kernel Analysis. In particular, the global local-inter structure is proposed to transfer the structure beyond the mini-batch. In detail, the method first divides the feature information into three structures: intra-feature, local inter-feature, and global inter-feature structures to subdivide the structure and transfer the diversity of the structure. Then, we adopt CKA which shows a more accurate similarity metric compared to other metrics between two different models or representations on different spaces. In particular, a memory-augmented transfer method with clustering is implemented for the global structures. The methods are empirically analyzed on the nine tasks for language understanding of the GLUE dataset with Bidirectional Encoder Representations from Transformers (BERT), which is a representative neural language model. In the results, the proposed methods effectively transfer the three types of structures and improves performance compared to state-of-the-art distillation methods: (*i.e.*) ours achieve 66.61% accuracy compared to the baseline (65.55%) in the RTE dataset. Indeed, the code for the methods is available at https://github.com/maroo-sky/FSD.

1. Introduction

In current deep learning models, knowledge distillation (KD) is a common approach to transfer information of features of a larger model to a smaller student model (Gou, Yu, Maybank, & Tao, 2021). his approach reduces the difference in prediction confidence between two models. Confidence is usually represented as a probability vector. Moreover, distillation can be applied to various vector distributions to transfer more feature information. For example, distribution on an intermediate layer (Sun, Cheng, Gan, & Liu, 2019; Wang et al., 2020; Zhao, Xing, Wang, Song, & Xiao, 2023) or a final fully connected layer (Jiao et al., 2020) have been directly compared with a single level layer-wise method (*i.e.*) sentence by sentence level. A problem with the direct fitting of a vector is its huge constraint in geometric space, even in the same setting of neural networks. One of the solutions is to transfer more rich information, and pairwise relations between features (Li et al., 2020; Park, Kim, Lu, & Cho, 2019; Peng et al., 2019) have also been used to reduce the constraint. Pairwise relations methods solve the geometric constraint in vector-level, but still remain the constraint in structure-level. This constraint may cause ambiguity to the guide for a student to learn the teacher's knowledge.

Transferring more rich information on representations' connectivity is a possible solution. Centered Kernel Alignment (CKA) (Cortes, Mohri, & Rostamizadeh, 2012) is a suitable metric for this approach as it assigns a similarity value to feature structure. Furthermore, its score is higher and more consistent on potentially similar representations trained on different architectures and layers than other similarity metrics (*i.e.*) cosine similarity, and canonical correlation analysis (Kornblith, Norouzi, Lee, & Hinton, 2019). This property is expected to help distillation focus on more informative feature distribution. Implementations of this approach have been reported in a few recent computer vision tasks, but widely used BERT model in natural language processing is not sufficiently studied yet.

In this work, we propose *feature structure distillation* (FSD) method to adapt CKA to KD between a teacher and a student model for transferring rich information. The proposed methods transfer rich information categorized into three types of structures on the feature representations

[*] Corresponding author.
E-mail addresses: jungheejun93@gm.gist.ac.kr (H.-J. Jung), marri1818@gm.gist.ac.kr (D. Kim), nash@jbnu.ac.kr (S.-H. Na), kikim01@gist.ac.kr (K. Kim).

https://doi.org/10.1016/j.eswa.2023.120980
Received 28 February 2023; Received in revised form 10 June 2023; Accepted 9 July 2023
Available online 15 July 2023

to subdivide the level of teacher's knowledge: (1) intra-feature, (2) local inter-feature, and (3) global inter-feature structures. A separate distillation loss is introduced for each structure defined on the feature distribution generated from the penultimate layer. To obtain the global inter-feature structures over the full batch of training samples, we newly add a memory architecture that is induced via clustering.

We present experiments on the General Language Understanding Evaluation (GLUE) (Wang et al., 2019) benchmark with BERT distilled by FSD methods. In the results, FSD methods show possibility that these methods outperform other state-of-the-art KD methods and even teacher models in some tasks. Far from many previous works (Jiao et al., 2020; Li et al., 2020; Park et al., 2019; Park, Kim, & Yang, 2021; Peng et al., 2019; Sun et al., 2019, 2020; Wang et al., 2020) which mainly focus on model performance, we provide the results of the restoration rate of the teacher's prediction and the similarity change of geometric structures for deeper understanding of the structure distillation. Our method shows better performance than baselines in GLUE tasks, the necessity of global structure for stability, and the student's feature structure is close to the teacher's compared to baselines. Our key contributions are summarized as follows:

- We adapt CKA to KD for more informative transfer of structures in BERT.
- We categorize three feature structures (intra-feature, local inter-feature, and global inter-feature structure).
- We propose their distillation methods, especially memory augmentation with clustering for global structures.
- We empirically analyze restoration rate, patterns of transferring feature structures, and task-specific properties.
- We validate practical usefulness over a wide range of language understanding tasks (GLUE benchmark).

We describe related work in Section 2, our methodology in Section 3 for three feature structures, and settings of experimental environments in Section 4. Then we represent the results on each experiment in Section 5, the quantitative and qualitative analysis in Section 6 and summarize our contribution in Section 7.

Terms and Notations

Feature	A representation vector
Relation	A pair-wise relation of two features
Feature structure	A set of relations
Feature distribution	A set of features
a	A scalar (integer or real)
$\mathbf{a}$	A vector
$\mathbf{A}$	A matrix
$\mathbf{I}_n$	Identity matrix with n rows and n columns
$\mathbf{J}_n$	All-ones matrix with n rows and n columns
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^{m\times n}$	m by n shape of matrix
$\mathbb{R}^{m\times n\times k}$	m by n by k shape of 3rd-order tensor
$\{0,1,\dots,n\}$	The set of all integers between 0 and n
$\mathcal{D}_{KL}(P\|\|Q)$	Kullback–Leibler divergence of P and Q
$\|\|\mathbf{x}\|\|_2$	L2 norm of $\mathbf{x}$
FSD	Proposed distillation for all feature structure types
FSD$_I$	FSD for only local intra-feature structure
FSD$_L$	FSD for only local inter-feature structure
FSD$_G$	FSD for only global inter-feature structure
FSD$_{IL}$	Integration of FSD$_I$ and FSD$_L$

2. Related work

2.1. Analysis of similarity of representation

The similarity between representations of deep networks has been measured by various methods. Canonical Correlation Analysis (CCA) (Hotelling, 1992) estimates the association between two variables and identifies a linear relationship with weight to maximize correlation. CCA is sensitive to perturbation when the condition number of representations is large (Golub & Zha, 1995). To reduce the sensitivity of perturbation, Singular Value Canonical correlation Analysis (SVCCA) (Raghu, Gilmer, Yosinski, & Sohl-Dickstein, 2017) applies singular value decomposition to use more important principal components, and Projection Weighted CCA (PWCCA) (Morcos, Raghu, & Bengio, 2018) assigns higher weights to more important canonical correlations. These methods aimed to assign the same relation of flexibly located representations in different models, but the consistency of their methods is insufficient (Kornblith et al., 2019). CKA is an alternative for enhancing the invariance to orthogonal transformation and isotropic scaling, which is expected to enhance the consistency (Kornblith et al., 2019). The metric improved the performance of alignment-based algorithms (Cortes et al., 2012), measuring the similarity between kernels or kernel matrices. Furthermore, CKA outperforms CCA, SVCCA, and PWCCA on the test of identifying corresponding layers (Kornblith et al., 2019). Even though CKA has shown higher and more consistent results on two different models, which trained with the same dataset but different architecture, than other similarity metrics, it is not utilized as a similarity metric for KD.

2.2. Knowledge distillation for BERT

KD (Hinton, Vinyals, & Dean, 2015) is a method to transfer dark knowledge of a large teacher model to a smaller student model while preserving the training accuracy, and this method is applied to DistilBERT (Sanh, Debut, Chaumond, & Wolf, 2019). An extension of KD is to directly reduce distance between representations. For example, TinyBERT (Jiao et al., 2020) delivers word embedding, self-attention head, and representations on selected intermediate layers. MobileBERT (Sun et al., 2020) moves representations on all layers to a student of the same number of layers, and MiniLM (Wang et al., 2020) uses relations between the values in the self-attention and the attention distribution that is computed from the scaled dot products of the queries and keys (Vaswani et al., 2017). DistilBERT and TinyBERT models perform distillation on the pre-training and fine-tuning stages but MobileBERT and MiniLM models operate distillation only on the pre-training stage. To reduce the interference of other factors in the analysis, we conduct distillation on the fine-tuning stage as *patient knowledge distillation* (PKD) (Sun et al., 2019). The method implements teacher representations from multiple intermediate layers normalized to the student layers in downstream tasks, enabling transfer between neural networks of different numbers of layers. These methods penalize the difference between teacher and student features, then force them closer in the same vector space. However, CKA allows of using different vector spaces and dimensions.

2.3. Transferring rich information

Transferring rich information as differences or relation has been introduced in a few vision tasks. Correlation Congruence for Knowledge Distillation (CCKD) (Peng et al., 2019) transfers a correlation matrix between representations generated from kernels, Relational Knowledge Distillation (RKD) (Park et al., 2019) evaluates the difference in Euclidean distance or cosine similarity, and Local Correlation Consistency for Knowledge Distillation (LKD) (Li et al., 2020) additionally uses the difference of angles, each of which is determined by three representations. Indeed, Contextual Knowledge Distillation CKD (Park et al., 2021) proposed layer transforming relation as well as word relation-based contextual knowledge distillation with same manner of RKD to evaluate the difference of structure. It extends transferring teacher structure from word level (within same layer) to layer level (over layers). In addition, Similarity-Preserving Knowledge Distillation (Tung

& Mori, 2019) transfers pairwise similarity with outer products of mini-batch samples. Another approach (Liu et al., 2019) conveys teacher knowledge with Instance Relationship Graph (IRG) to student for reducing the distance of vertex by vertex and edge by edge of IRG. Differently, our methods cover a wider range of feature structures as global structures and effective intra-feature structures are specifically designed for transformers. To measure the difference between structures, we adopt CKA, which effectively maintains the implicit relations between representations (Kornblith et al., 2019). CKA has been adopted for KD in convolutional networks (Wu et al., 2020) and has shown successful performance improvement, but the extension to global structure and a deeper analysis in the transformer networks have rarely been discussed.

3. Method

In this section, we clarify the three feature structures to subdivide the structure level as a token, sentence, and global level. Also, we utilize the CKA metric to implement KD on each structure level for rich information.

3.1. Base knowledge distillation

We set two compatible base settings for KD to transfer a feature distribution introduced in previous works (Hinton et al., 2015; Sun et al., 2019). In the former setting, given N training samples for training a student model S, a fine-tuned teacher model T transfers a feature distribution on the final layer by training S with the similarity loss as

$$\mathcal{L}_{KLD} = \sum_{i=1}^{N} \mathcal{D}_{KL}\left(\frac{f_T(\mathbf{x}_i)}{\tau} \parallel \frac{f_S(\mathbf{x}_i)}{\tau}\right), \tag{1}$$

$f(\cdot)$ is a neural network to generate a probability vector through softmax function and layers. An input sample for the network is composed of vector $\mathbf{x}_i$ and its ground-truth $\mathbf{y}_i$, where i is the sample index in the training data. The temperature τ is to control relaxation. To distill teacher knowledge, the model is trained by a task-specific cross-entropy loss $\mathcal{L}_{CE}$ with $\mathcal{L}_{KLD}$ as

$$\mathcal{L}_{VKD} = \alpha\mathcal{L}_{CE} + (1-\alpha)\mathcal{L}_{KLD}, \tag{2}$$

where α is the interpolation rate between two loss functions, which needs to be tuned empirically. This approach to transfer a feature with the label is called *vanilla knowledge distillation* (VKD) in this paper. The latter setting follows the Eq. (2), but the similarity loss is replaced with the distance between hidden vectors generated from the several intermediate layers and penultimate layers $f'(\cdot)$ of T and S, which is the method of PKD (Sun et al., 2019). As the setting of both methods, we initialize the parameters of S from the pre-trained T and then perform distillation on the fine-tuning stage.

In this paper, the proposed FSD method is used as distillation loss functions in training the student model as

$$\mathcal{L} = \mathcal{L}_{VKD} + \beta\mathcal{L}_{FSD}, \tag{3}$$

where β is a hyper-parameter to control the interpolation rate between VKD loss and the proposed distillation loss.

3.2. Feature structure distillation with CKA

3.2.1. Overview

In this paper, we propose FSD method to transfer more rich information on representations by comparing feature structures relations rather than relation. *Feature structure* is split into three groups with respect to their locality: intra-feature, local inter-feature, and global inter-feature structures. Fig. 1 shows the distinction of the structures. In addition, feature structures are conveyed only on the penultimate layer to compare baselines (PKD and RKD).

3.2.2. Similarity between feature structures

We adapt CKA for evaluating similarity between feature structures in order to use its robustness to the constraint of feature distribution and consequently to reduce ambiguity of distillation.

$$CKA(\mathbf{E}^1, \mathbf{E}^2) = \frac{HSIC(\mathbf{K}, \mathbf{L})}{\sqrt{HSIC(\mathbf{K}, \mathbf{K})HSIC(\mathbf{L}, \mathbf{L})}}, \tag{4}$$

where $\mathbf{E}^1$ and $\mathbf{E}^2$ are features; the function $HSIC$ is the Hilbert–Schmidt Independence Criterion for determining Independence of two sets of variables (Gretton, Bousquet, Smola, & Schölkopf, 2005); $\mathbf{K} = \mathbf{E}^{1^{\mathrm{T}}}\mathbf{E}^1$, $\mathbf{L} = \mathbf{E}^{2^{\mathrm{T}}}\mathbf{E}^2$. The function HSIC is the Hilbert–Schmidt Independence Criterion (Gretton et al., 2005) defined as

$$HSIC(\mathbf{K}, \mathbf{L}) = \frac{1}{(\mathbf{N}-1)^2} tr(\mathbf{KCLC}), \tag{5}$$

where tr is a trace in a matrix, $\mathbf{C}$ is a centering matrix $\mathbf{C}_n = \mathbf{I}_n - \frac{1}{n}\mathbf{J}_n$.

In each proposed method, we use different $\mathbf{E}^1$ and $\mathbf{E}^2$, but they are all based on the hidden vectors generated from the penultimate layer of teacher, and student, notated as $\mathbf{H}^T$ for the teacher and $\mathbf{H}^S$ for the student in the shape of $\mathbb{R}^{|B|\times|W\|D|}$. The constants $|B|$, $|W|$, and $|D|$ are the number of samples in a mini-batch, the maximum sequence length, and the hidden state dimension, respectively.

3.2.3. Intra-Feature Structure Distillation (FSD_I)

intra-feature structure implies the set of difference values between segments of the hidden vector from the penultimate layer from a single input sample. In the transformer networks, the unit for segmentation is a token but its structure is not considered to transfer. To obtain the difference of token-level structures between the teacher and student model, we split the hidden vector $\mathbf{H}_i^S \in \mathbb{R}^{|W\|D|}$ into token-level feature vectors $\mathbf{H}_{i,j}^S \in \mathbb{R}^{|D|}$ for the jth token from ith input sample and this is equally applied to the teacher model. Then, we reshape teacher and student penultimate layer representation $\mathbf{H}^T$ and $\mathbf{H}^S$ into $\mathbf{H}_{re}^T$ and $\mathbf{H}_{re}^S$ in shape of $\mathbb{R}^{|B|\times|W|\times|D|}$. The loss function implying the difference is defined as

$$\mathcal{L}_I = -\frac{1}{|B|}\sum_{i=1}^{|B|} \log CKA|(\mathbf{H}_{re}^S, \mathbf{H}_{re}^T)|. \tag{6}$$

3.2.4. Local inter-Feature Structure Distillation (FSD_L)

Local inter-feature structure implies the set of difference between hidden vectors generated at the penultimate layer from the samples of a mini-batch. Compared to the intra-feature structure method, it deals with the structure between samples rather than internal units from a single sample. To reduce the constraints of representation, we transfer the sentence-level structure with CKA instead Euclidean distance or cosine similarity. The distillation loss adapting CKA for comparing the local inter-feature structures is

$$\mathcal{L}_L = -\log|CKA(\mathbf{H}^S, \mathbf{H}^T)|. \tag{7}$$

3.2.5. Global inter-Feature Structure Distillation via memory augmentation (FSD_G)

The local approach is easy to implement but transfers only the partial inter-feature structures because it evaluates the structures only for samples in the same mini-batch. To transfer all inter-feature structures, a pairwise difference for two samples has to be evaluated for $O(n^2)$ relations, but the local method considers $O(\frac{n}{b}b^2) = O(n)$ where n is the total number of samples in the full batch, and b is a given constant number of samples in a mini-batch. As shown in Fig. 2, relations in mini-batches could not cover all relations in full batches. Thus, the coverage of FSD$_L$ rapidly decreases by the input samples size. This decrease implies that the transferred feature structure is easily insufficient for a large dataset. Moreover, the large scale makes transferring all structures computationally inefficient and infeasible in parallel programming with limited hardware.

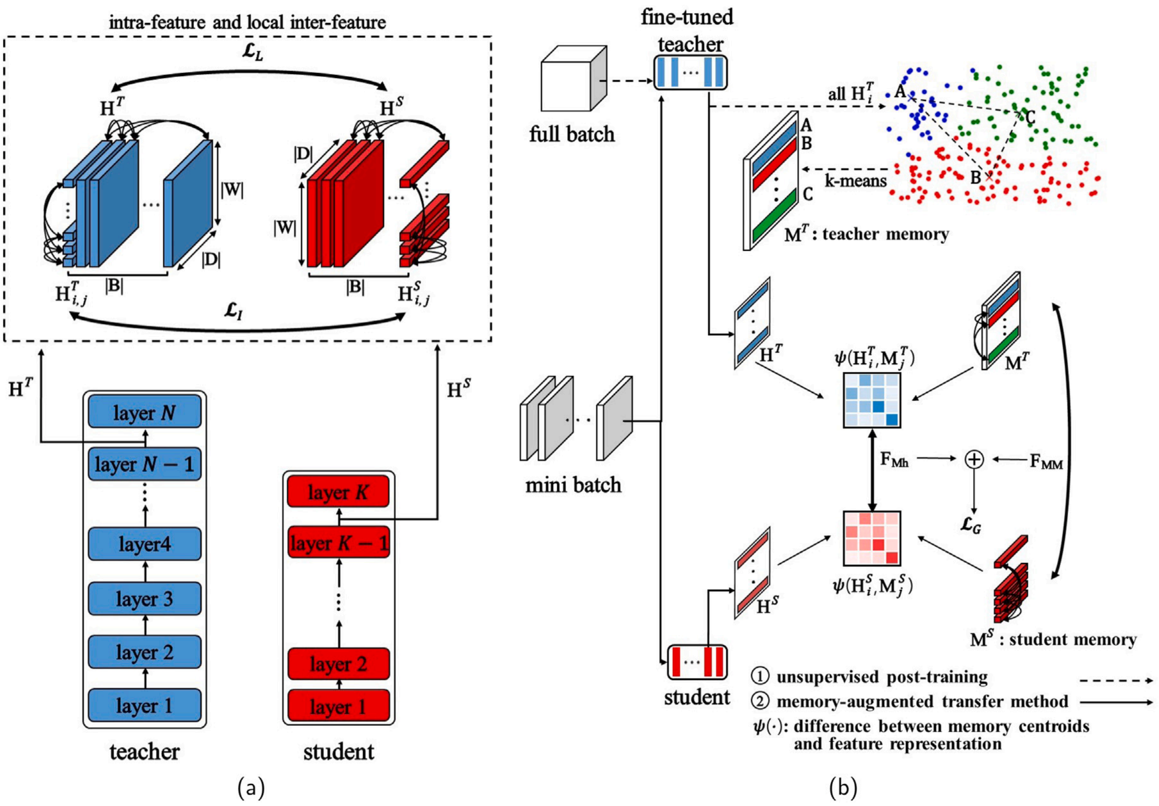

Fig. 1. Overview of the proposed loss functions of distillation methods for three feature structures: $\mathcal{L}_I$ and $\mathcal{L}_L$ in Fig. 1(a) uses CKA between teacher and student representations on the penultimate layer. In Fig. 1(b), the first stage is an unsupervised post-training and the second stage is a memory-augmented transfer method. In the first stage, $\mathbf{M}^T$ is first trained to memorize the centroids of representations on the penultimate layer of a teacher model. Then, its stored centroids are transferred to a student model by $\mathcal{L}_G$ in the second stage.

Algorithm 1 FSD Loss

```
Require: $i$-th iteration mini-batch $\mathbf{X}_i \in \mathcal{X}$ (dataset) $\mathbf{X}_i \in \mathbb{R}^{|B|\times|W|\times|D|}$,
    hyper-parameters $\beta, \gamma_m, \gamma_i, \gamma_l, \gamma_g$, and learning rate $\eta$. $M$ is a memory,
    $f^H(\cdot)$ is output of penultimate layer. $S$ and $T$ is student and teacher
    respectively.
Ensure: Loss $\mathcal{L}_{FSD}$

    # We freeze the teacher memory $M^T$ during the training.

    reshape$(\cdot)$: $\mathbb{R}^{|B|\times|W|\times|D|} \rightarrow \mathbb{R}^{|B|\times|W||D|}$
    $\mathbf{H}^S_{re}, \mathbf{H}^T_{re} \leftarrow f^H_S(\mathbf{X}_i), f^H_T(\mathbf{X}_i)$
    $\mathcal{L}_I \leftarrow -\frac{1}{|B|}\sum_{i=1}^{|B|} \log CKA|(\mathbf{H}^S_{re}, \mathbf{H}^T_{re})|$        {Eq. (6)}
    $\mathbf{H}^S, \mathbf{H}^T \leftarrow$ reshape$(\mathbf{H}^S_{re})$, reshape$(\mathbf{H}^T_{re})$
    $\mathcal{L}_L \leftarrow -\log|CKA(\mathbf{H}^S, \mathbf{H}^T)|$        {Eq. (7)}
    $\mathbf{F}_{\mathbf{MM}} \leftarrow -\log|CKA(\mathbf{M}^T, \mathbf{M}^S)|$        {Eq. (8)}
    $\mathbf{F}_{\mathbf{Mh}} \leftarrow \frac{\sum_{i,j}||\psi(\mathbf{H}^T_i,\mathbf{M}^T_j)-\psi(\mathbf{H}^S_i,\mathbf{M}^S_j)||_2^2}{|B||C|}$        {Eq. (9)}
    $\mathcal{L}_G \leftarrow \gamma_m \mathbf{F}^E_{\mathbf{Mh}} + (1-\gamma_m)\mathbf{F}^C_{\mathbf{Mh}} + \mathbf{F}_{\mathbf{MM}}$        {Eq. (10)}
    $\mathcal{L}_{FSD} \leftarrow \gamma_i \mathcal{L}_I + \gamma_l \mathcal{L}_L + \gamma_g \mathcal{L}_G$
    return $\mathcal{L}_{FSD}$
```

To overcome this problem, we propose an approach to effectively transfer the inter-feature structures between samples in different mini-batches, called *global inter-feature* structures. The key idea is to store centroids of full batch into a memory and then transfer the structures between centroids in a teacher and a student. As locating features of the student to its centroids, the method transfers the global inter-feature structures built by samples near the involved centroids. This method has two sequential stages: (1) *unsupervised post-training* and (2) *memory-augmented transfer method*. In the first stage, we use the fine-tuned teacher to generate representations of all samples from the penultimate layer in the full batch. After randomly initialized element vectors in memory, each vector is updated to minimize the Euclidean distance to its corresponding k-nearest neighbors, thereby performing *k-means clustering*. The second stage is the memory-augmented transfer method to reduce the difference of inter-feature structures over the centroids in the teacher memory $\mathbf{M}^T$ and student memory $\mathbf{M}^S \in \mathbb{R}^{(|C|\times|W||D|)}$, where the number of elements of the memory $|C| \in \{100, 300\}$. The difference $\mathbf{F}_{\mathbf{MM}}$ is defined as following equation:

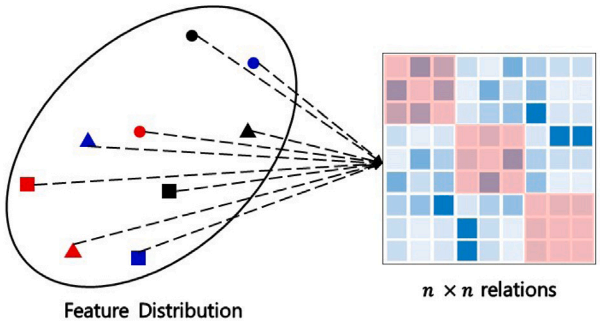

Fig. 2. The difference of covered relations between a local structure and a global structure is shown. The blue blocks of the right-side matrix are relations of the global structure in the full batch. The red blocks are found in mini-batches (local structure). Variously colored and shaped markers indicate features and the matrix shows relations over the features.

$$\mathbf{F}_{\mathbf{MM}} = -\log|CKA(\mathbf{M}^T, \mathbf{M}^S)|. \tag{8}$$

The transferred structures on the centroids are propagated to features of the student by learning the teacher's feature-to-centroid difference. For transferring centroid difference, we define $\mathbf{F}_{\mathbf{Mh}}$ as

$$\mathbf{F}_{\mathbf{Mh}} = \frac{\sum_{i,j} \| \psi(\mathbf{H}_i^T, \mathbf{M}_j^T) - \psi(\mathbf{H}_i^S, \mathbf{M}_j^S) \|_2^2}{|B \| C|}, \tag{9}$$

where ψ is a function for evaluating distance. $\mathbf{F}^E_{\mathbf{Mh}}$ and $\mathbf{F}^C_{\mathbf{Mh}}$ specify ψ to Euclidean distance and cosine similarity, respectively. The final distillation loss for transferring the global inter-feature structure is defined as follows:

$$\mathcal{L}_G = \gamma_m \mathbf{F}^E_{\mathbf{Mh}} + (1 - \gamma_m)\mathbf{F}^C_{\mathbf{Mh}} + \mathbf{F}_{\mathbf{MM}}, \tag{10}$$

where γ_m is a hyper-parameter for interpolation.

We used Euclidean and cosine similarity as Park et al. (2019, 2021) instead of CKA for $\mathbf{F}_{\mathbf{Mh}}$ to allow this method to set a flexible memory size because CKA cannot be directly applied when the batch size of $\mathbf{H}$ is different to the number of centroids in the memory.

3.2.6. Integration

The intra-feature, local inter-feature, and global inter-feature structures represent different types of structures. Thus, their integration is a natural extension for transferring more rich feature structure. The integration is simply implemented as the sum of all three loss functions as:

$$\mathcal{L}_{FSD} = \gamma_i \mathcal{L}_I + \gamma_l \mathcal{L}_L + \gamma_g \mathcal{L}_G, \tag{11}$$

where the hyper-parameter γ_i, γ_l, and γ_g are interpolation rates. This integrated loss is used for training a student model as described in the following Algorithm 1.

4. Experiments

In this section, we describe the experimental settings and datasets. We empirically analyze FSD method applied to BERT for well-known language understanding tasks. Beyond usual performance and computational efficiency evaluation to evaluate practical impact, we focus on understanding of how teacher knowledge is effectively transferred, because its impact to performance is promising as shown in Xu et al. (2021) and the final performance is interfered by other effects of distillation as generalization (Yuan, Tay, Li, Wang, & Feng, 2020). The analysis has three parts (1) quantitative analysis, (2) qualitative analysis, (3) and additional discussion. In the quantitative analysis, we evaluate the practical impact of our method compared with state-of-the-art, the impact of each feature structure level, and the impact of memory stability. In the qualitative analysis, we analyze how much student reflects and is close to teacher's structure. In the last, we discuss geometry property, and model and time complexity. The performance is for evaluating practical impact of our methods in comparison with state-of-the-art. The other three parts are to evaluate the effectiveness of transferring teacher knowledge on feature distributions in various perspectives.

4.1. Datasets

The General Language Understanding Evaluation benchmark (GLUE)[1] is presented in Table 1

Table 1
Evaluation Metrics and the number of dataset of GLUE benchmark. |**train**| and |**dev**| are the size of training and development dataset, and Corr is a correlation.

	Corpus	|train|	|dev|	Metrics
Single-sentence tasks	CoLA	8.5k	1k	Matthews Corr
	SST-2	67k	872	Accuracy
Similarity and paraphrase tasks	QQP	364k	40k	Accuracy/F1
	MRPC	3.7k	408	Accuracy/F1
	STS-B	7k	1.5k	Pearson Corr Spearman Corr
Inference tasks	MNLI	393k	20k	Accuracy
	RTE	2.5k	276	Accuracy
	QNLI	105k	5.5k	Accuracy
	WNLI	634	71	Accuracy

GLUE the General Language Understanding Evaluation (Wang et al., 2019) consists of nine English sentence-understanding tasks. Single-sentence tasks include the Corpus of Linguistic Acceptability (CoLA) (Warstadt, Singh, & Bowman, 2019), Stanford Sentiment Treebank (Socher et al., 2013). In the similarity and paraphrase tasks, Microsoft Research Paraphrase Corpus (MRPC) (Dolan & Brockett, 2005), Quora Question Pairs (QQP),[2] and Semantic Textual Similarity Benchmark (STS-B) (Cer, Diab, Agirre, Lopez-Gazpio, & Specia, 2017) are included. In the last, inference tasks include the Multi-Genre Natural Language Inference Corpus (MNLI) (Williams, Nangia, & Bowman, 2018), Stanford Question Answering Dataset (QNLI) (Rajpurkar, Zhang, Lopyrev, & Liang, 2016), Recognizing Textual Entailment (RTE) (Bentivogli, Magnini, Dagan, Dang, & Giampiccolo, 2009; Dagan, Glickman, & Magnini, 2006; Giampiccolo, Magnini, Dagan, & Dolan, 2007), and Winograd Schema Challenge (WNLI) (Levesque, Davis, & Morgenstern, 2012).

4.2. Distillation settings

4.2.1. Environment setup

We conduct knowledge distillation with GLUE on a single RTX-2080-Ti and RTX-8000 GPU with 32 batches, 128 max sequence length, and 768 dimensions. WNLI, MPRC, and SST-2 are implemented on single RTX-2080-Ti, and RTE, STS-B, CoLA, and QNLI are implemented on a single RTX-8000 GPU. FSD of QQP and MNLI are implemented on a single RTX-8000 because of the memory size, and other methods of QQP and MNLI are operated on a single RTX-2080-Ti GPU. Each task performance is slightly different depend on GPU device and number of device.

4.2.2. BERT-base preparation

We set a 12-layer transformer encoder with 768 hidden nodes and 12 attention heads as a teacher model. We conduct fine-tuning with the uncased version of pre-trained BERT-base[3] on nine GLUE tasks independently. The maximum sequence length is 128 which is referred in Sun et al. (2019). The number of train epochs is 3. The training batch size is 32. The learning rate is 2e–5 except for STS-B and WNLI tasks, which are set in 5e–5 to slightly improve teacher performance. We note that fine-tuning of BERT-base can be more improved by adding other methods irrelevant to knowledge distillation. In this paper, our primary goal is not to solve language understanding tasks by whatever means possible, but to prove the impact of more accurate transferring of teacher's knowledge in a practical environment. Therefore, fair comparative group is the state-of-the-art transferring methods rather than the state-of-the-art language understanding model.

4.2.3. Baseline method settings

We reproduce VKD, PKD, MiniLM, and RKD but MiniLM (Wang et al., 2020) performs KD on the pre-training stage. For consistency with VKD, PKD, and FSD, we apply MiniLM method on the fine-tuning stage. Previous work (Sun et al., 2019) uses 6-layers of BERT model (BERT$_6$) as a student and we implement distillation experiments

[1] https://gluebenchmark.com/tasks.

[2] https://data.quora.com/First-Quora-Dataset-Release- Question-Pairs.

[3] https://s3.amazonaws.com/models.huggingface.co/ bert/bertbase-uncased-pytorch model.bin.

Table 2

Performance on small tasks of GLUE benchmark. Mean and standard deviation of their evaluation metrics on six runs on their development (dev.)sets. [*p*-value]s of t-test are calculated for comparison of the proposed and the best baseline methods. The numbers imply accuracy in WNLI and RTE, Pearson/Spearman correlation in STS-B, Matthew's correlation in CoLA, and F1/accuracy in MRPC task. A bold number is the best result among distillation methods in each task.

	Method	WNLI	RTE	STS-B	CoLA	MRPC
Teacher	HF[a]	56.34	67.15	93.95/83.70	49.23	89.47/85.29
	BERT-base	56.34	66.79	88.78/88.48	55.47	86.00/81.10
Baseline	VKD	54.37(±0.77)	65.40(±1.56)	88.16(±0.17)/87.83(±0.16)	41.46(±0.89)	86.24(±0.52)/80.75(±0.65)
	PKD	54.37(±2.36)	63.90(±0.79)	88.45(±0.10)/88.08(±0.06)	41.87(±1.13)	86.37(±0.60)/80.98(±0.81)
	MiniLM[b]	48.45(±6.19)	61.49(±0.59)	87.82(±0.08)/87.49(±0.09)	33.34(±0.83)	84.57(±1.36)/78.13(±2.02)
	RKD	51.83(±5.84)	65.22(±0.90)	88.43(±0.17)/88.12(±0.14)	**43.07(±1.49)**	86.87(±0.47)/81.84(±0.35)
Proposed	FSD	**55.49(±1.61)** [0.115]	**66.61(±1.01)** [0.002]	**88.69(±0.10)/88.33(±0.09)** [0.004]/[0.013]	43.03(±1.43) [0.521]	**87.10(±0.24)/82.20(±0.33)** [0.115]/[0.060]

[a] Denotes that the results are taken from the huggingface BERT$_{BERT}$ (https://huggingface.co/transformers/v2.6.0/examples.html).

[b] Denotes reproduced model for consistency with VKD, PKD, and FSD methods.

Table 3

Performance on large tasks of GLUE benchmark. Mean and standard deviation of their evaluation metrics on six runs on their development (dev.)sets. [*p*-value]s of t-test are calculated for comparison of the proposed and the best baseline methods. The numbers are measured by accuracy in SST-2, QNLI, and MNLIs, and accuracy/F1 in QQP task. A bold number is defined by the same manner in Table 2.

	Method	SST-2	QNLI	QQP	MNLI/MNLI-mm
Teacher	HF*	91.97	87.46	88.40/88.31	90.61/81.08
	BERT-base	92.09	91.60	91.07/88.06	84.70/84.65
Baseline	VKD	90.92(±0.62)	88.46(±0.47)	91.03(±0.08)/87.96(±0.10)	82.20(±0.19)/82.85(±0.12)
	PKD	90.77(±0.41)	88.57(±0.17)	91.00(±0.11)/87.92(±0.14)	82.27(±0.10)/82.57(±0.21)
	MiniLM†	90.23(±0.39)	**89.48(±0.19)**	90.53(±0.02)/87.21(±0.02)	82.23(±0.09)/82.58(±0.13)
	RKD	91.02(±0.19)	88.91(±0.31)	**91.21(±0.06)**/ 88.13(±0.08)	82.39(±0.19)/82.92(±0.13)
Proposed	FSD	**91.04(±0.28)** [0.348]	88.97(±0.25) [0.995]	91.19(±0.05)/**88.14**(±0.08) [0.982]/[0.402]	**82.42**(±0.13)/**83.00(±0.20)** [0.386]/[0.242]

with the same student architecture. We utilize parameters from 1st to 6th layer of pre-trained BERT$_{BASE}$ to initialize BERT$_6$. Fine-tuning for VKD, we conduct each task with α from $\{0.2, 0.5, 0.7\}$, temperature τ from $\{5, 10, 20\}$, and learning rate from {1e–5, 2e–5, 5e–5} to search for the best model. Additionally, we set angle and distance loss hyper-parameters introduced in To reproduce RKD (Park et al., 2019), we set hyper-parameters for its angle and distance loss from $\{200.0, 2000.0, 20000.0\}$, and $\{100.0, 1000.0, 10000.0\}$.

4.3. FSD settings

4.3.1. FSD method setting

We fix the best cases of α, τ and epochs on each downstream task as the result of grid search to reduce the cost of tuning hyper-parameters in FSD. We conduct additional hyper-parameters β set 1 in FSD (w/o G) and FSD methods to reduce hyper-parameter space and set β from $\{3, 4, 5, 6\}$ except STS-B, which is set from $\{3, 4, 5, 6, 10\}$ and learning rate from {3e–5, 4e–5, 5e–5, 6e–5}. Besides, by applying FSD (w/o G) method, we fix the best case of β on methods for the intra-feature and local inter-feature structure methods and set γ from $\{0.1, 0.2, \ldots, 0.9\}$.

4.3.2. FSD (w/o IL) settings

Implementing unsupervised post-training for the global inter-feature structures, we set different memory sizes depending on a given dataset size. In a large task such as QQP and MNLI, 300 memory entries ($|C| = 300$) are used to store centroids. The other smaller tasks used 100 entries ($|C| = 100$). The number of epochs for clustering is set by 3 for a teacher memory, which shows sufficient convergence in preliminary tests. After training the teacher memory, the structures in the memory are transferred to a randomly initialized student memory by distillation loss. Hyper-parameters for the distillation are separately set for each downstream task by greedy and grid search in terms of performance.

First, we set γ_m and β from {0, 1e–7, 1e–6, 1e–5, 1e–4, 1e–3,1e–2, 0.1, 1.0} find the best case, then set again γ_m from {6e–7, 7e–7, 8e–7, 9e–7, 2e–6, 3e–6, 4e–6, 5e–6} in the WNLI, RTE, and MNLI, and set from {6e–5, 7e–5, 8e–5, 9e–5 2e–4, 3e–4, 4e–4, 5e–4} in the SST-2, and from {6e–3, 7e–3, 8e–3, 9e–3 2e–2, 3e–2, 4e–2, 5e–2} in the QQP and set from {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} in the STS-B, CoLA, MRPC, and QNLI. Also, we set β again from {6e–8, 7e–8, 8e–8, 9e–8, 2e–7, 3e–7, 4e–7, 5e–7} in the STS-B, SST-2, and QQP, and set {6e–7, 7e–7, 8e–7, 9e–7, 2e–6, 3e–6, 4e–6, 5e–6} in the CoLA, MRPC, QNLI, and MNLI, and set from {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} in the WNLI, and RTE. The α, τ and epoch are fixed as the best case in VKD to reduce hyper-parameter optimization cost.

4.3.3. FSD loss hyper-parameters settings

We fix γ_m and set γ_g from {1e–7, 1e–6, 1e–5, 1e–4, 1e–3, 1e–2, 0.1, 1.0}. Also, we set γ_i from {0.1, 1.0, β of FSD (w/o LG)} and set γ_l from {0.1, 1.0, β of FSD (w/o IG)}.

5. Results

5.1. Quantitative analysis

5.1.1. Model performance

Table 2 presents the results on the GLUE dev. sets in small-sized tasks whose samples are less than 10,000. Table 3 shows the other larger-sized tasks and we estimate each task metric as shown in Table 1. The FSD methods show higher performance in the seven WNLI, RTE, MRPC, SST-2, STS-B, QQP, and MNLI (match and mismatch) tasks than the baseline methods, but CoLA, and QNLI show lower performance than the baseline. Compared to the teacher model, the proposed methods show slightly higher performance than BERT$_{BASE}$(T), by 1.0% on the MRPC task.

5.1.2. Ablation study

We implement an ablation study for the necessity of integration method for KD. We run a single feature structure method and a local-level structure method and we follow the same experimental setting as shown in Section 4.

As shown in Table 4, considering all structure results are higher than other methods. When we run a single structure method (w/o LG,

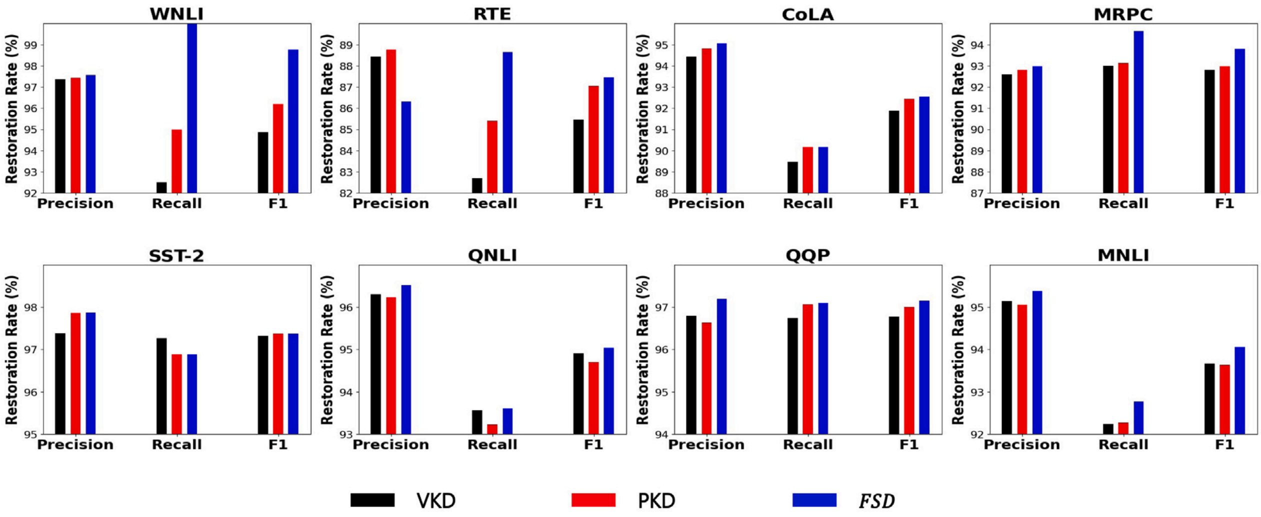

Fig. 3. Restoration rate of teachers' predictions in students on the GLUE tasks. MNLI used the matched subset.

Table 4
Ablation study on GLUE benchmark. The performance reduction rates over FSD ($\frac{\text{reduced performance}}{\text{FSD performance}}$) are shown. The results are averaged over six runs with different random seeds. **Avg.** is a average of results. The numbers follow same manner on Tables 2 and 3.

	Method	FSD (w/o LG)	FSD (w/o IG)	FSD (w/o IL)	FSD (w/o G)
Tasks	WNLI	−6.60%	−1.02%	−3.55%	−2.03%
	RTE	−0.72%	−0.45%	−0.18%	−0.63%
	CoLA	−2.15%	−8.14%	−1.60%	−2.50%
	SST-2	−0.44%	−0.57%	−0.31%	−0.34%
	QNLI	−0.14%	−0.56%	−0.51%	−0.31%
	STS-B	−0.04%/−0.02%	−0.03%/−0.08%	−0.21%/−0.22%	−0.02%/−0.03%
	MRPC	−0.18%/−0.40%	−0.10%/−0.24%	−1.24%/−2.16%	−0.08%/−0.24%
	QQP	−0.03%/−0.06%	−0.00%/+0.01%	−0.20%/−0.23%	−0.07%/−0.10%
	MNLI	−0.09%/−0.18%	−0.08%/−0.47%	−0.46%/−0.27%	+0.36%/−0.36%
	Avg.	−0.78%	−0.90%	−0.86%	−0.48%

Table 5
Each value represents the standard deviation of model performance of 18 runs composed 6 runs for three mini-batch sizes in $\{8, 16, 32\}$.

Method	MRPC	SST-2
FSD (w/o IG)	0.66/0.44	0.68
FSD (w/o IL)	**0.47/0.38**	**0.62**

w/o IG, and w/o IL), and local-level method FSD (w/o LG) shows less decline and (w/o IG) shows worst case among three cases. In addition, FSD (w/o G) case shows better than single structure distillation methods.

5.1.3. Stability of global structure for transferring

We estimate of standard deviation in different mini-batch cases to show that utilized memory method (FSD (w/o IL)) is less affected by mini-batch size. We select MRPC and SST-2 tasks because each task is in small and large dataset of the GLUE and both tasks are stably improved model performance. As shown in Table 5, two tasks standard deviation is the lowest in FSD (w/o IL).

5.2. Qualitative analysis

5.2.1. Restoration rate of teacher prediction

Fig. 3 shows the restoration rates of teacher prediction. As all tasks are binary classification except STS-B, we plot precision, recall, and F1-score for each method in each task, over teacher prediction result as the ground-truth. FSD method generally shows higher restoration rates than baseline.

5.2.2. Task-specific structural similarity between teacher and student

We evaluate CKA similarity heat maps of teacher by teacher (T-T), teacher by no-Distillation-Student (T-noDS), and teacher by student with methods for each task to investigate structural properties on CKA perspective. Fig. 4 shows CKA heat maps and Table 6 shows the average of similarities on the diagonal lines and the average over all tasks. CKA similarity evaluation is conducted on mini-batches. We split a test set to build a mini-batch pool for each task. Then, teacher and another model separately select their mini-batch and generates corresponding feature for evaluating CKA similarity. This value is shown in a pixel and we repeated it for all mini-batch pairs. The size of mini-batches differs by tasks because of different test set size.

Fig. 4, the lightness of each diagonal line implies the accuracy of transferring teacher knowledge captured by CKA. Its maximum value is 1.0 obtained in any T-T cases. More accurate numerical comparison in Table 6 shows that FSD shows significantly better average diagonal values over all tasks than other methods.

5.2.3. Patterns of transferring structures

To analyze the patterns of transferred structures, we evaluate *relation difference* (RD) for inter-feature structure (RD_{inter}) as

$$RD_{inter} = \sum_{i,j} \frac{|\psi(\mathbf{H}_i^T, \mathbf{H}_j^T) - \psi(\mathbf{H}_i^S, \mathbf{H}_j^S)|}{|B|^2}, \tag{12}$$

where $i, j \in \{1, 2, \dots, |B|\}$, and for intra-feature structure (RD_{intra}) as

$$RD_{intra} = \sum_{i,j,k} \frac{|\psi(\mathbf{H}_{i,j}^T, \mathbf{H}_{i,k}^T) - \psi(\mathbf{H}_{i,j}^S, \mathbf{H}_{i,k}^S)|}{|W|^2 \cdot |B|}, \tag{13}$$

where $i \in \{1,2, \dots, |B|\}$ and $j, k \in \{1,2, \dots, |W|\}$. This metric implies the average difference of the relation unit building intra-feature structures and inter-feature structures between the teacher and students.

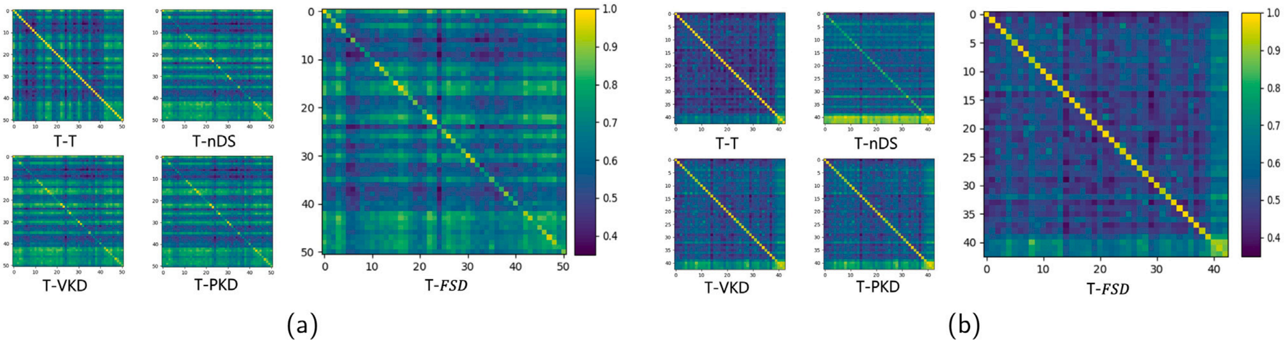

Fig. 4. CKA heat maps of GLUE benchmark. CoLA data set result is representative on GLUE benchmark, x-axis, y-axis are indices of mini-batches on the same data set. Each pixel shows the CKA similarity between teacher's and other model's representations generated from their mini-batches. In the diagonal lines, the same samples are used for the generation. Right side bar shows the normalized range of CKA similarities observed in each task. (a) is a CoLA case and (b) is a SST-2 case. The heat maps of all the other tasks are in Appendix.

Table 6
Each value is the average of diagonal values for each task and each teacher-student pair.

CKA heat map diagonals											
	Method	WNLI	RTE	STS-B	CoLA	MRPC	SST-2	QNLI	QQP	MNLI	Avg
No KD	T-T	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	T-nDS	0.998	0.985	0.954	0.803	0.981	0.794	0.972	0.909	0.947	0.927
Baseline	T-VKD	0.997	0.984	0.955	0.811	0.979	0.910	0.965	0.889	0.919	0.934
	T-PKD	0.997	0.980	0.956	0.813	0.979	0.900	0.961	0.903	0.944	0.937
Proposed	T-FSD	0.999	0.990	0.981	0.835	0.990	0.942	0.985	0.958	0.942	**0.958**

Table 7
Each value is the average of ranks of four *relation difference* results of each task. The lower is the better and the final difference values after training are used for the ranking. (Avg: the average of the average ranks over all tasks).

Average rank of relation difference table											
	Method	WNLI	RTE	STS-B	CoLA	MRPC	SST-2	QNLI	QQP	MNLI	Avg
Baseline	VKD	3.00	2.50	3.00	3.00	3.25	3.50	2.75	**2.50**	2.50	2.89
	PKD	2.50	2.50	2.50	2.75	2.50	3.25	2.75	**2.50**	2.75	2.64
	RKD	2.50	3.00	3.00	3.00	2.50	**1.50**	**2.00**	**2.50**	**2.25**	2.44
Proposed	FSD	**2.00**	**2.00**	**1.50**	**1.25**	**2.25**	1.75	2.50	**2.50**	2.50	**2.03**

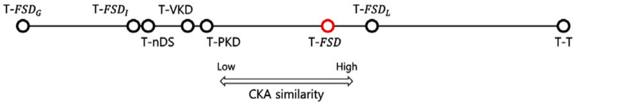

Fig. 5. Each point represents the average of diagonal values on CKA heat map. The average of T-FSD$_G$ is 0.905 and T-T case is 1.000. We plot each point to reflect difference.

Table 8
The number of model parameters (#Param), and training and inference time ratio compare to VKD, *i.e.*) $t_{\text{methods}}/t_{\text{VKD}}$, where t is the training or inference time.

Method	#Param	Training ↑	Inference ↑
VKD	6.70M	1.00 ×	1.00 ×
PKD	6.70M	0.96 ×	1.00 ×
RKD	6.70M	0.77 ×	1.00 ×
FSD	7.68M	0.92 ×	0.88 ×

RD^E and RD^C are defined by the same manner of $\mathbf{F}^E_{\mathbf{Mh}}$ and $\mathbf{F}^C_{\mathbf{Mh}}$ to specify ψ. Then, we evaluate the rank of the last iteration RD values over each method on each task. Baseline models and primitive FSD methods are tested for clear analysis of the impact of structure types.

RD values in training are illustrated in Fig. 6, and their RD ranks are shown on the Table 7. The lower RD, the better rank close to one. In the WNLI graph, most proposed methods were more effective to reduce RD^E than the baselines, while RD^C shows inconsistent superiority. The pattern is similarly observed in the other GLUE tasks. In Table 7, the proposed FSD method shows the best ranks on most tasks. FSD shows the best average rank on the GLUE task, where FSD is the first rank in the WNLI, RTE, STS-B, CoLA, MRPC, and QQP and the second-best rank in the rest of GLUE tasks.

5.2.4. Model and time complexity

As shown in Table 8 we conduct model and time complexity for training and inference. Even though our proposed method model size is about 1.15 times larger than RKD, the FSD is much faster than RKD during training because the observed structure size is $O(n^2)$ and $O(n^3)$ in FSD and RKD, respectively.

6. Analysis and discussion

6.1. Our model

To estimate p-value, we set null hypothesis ($\mathcal{H}_0$): $\mu_s = \mu_{\text{FSD}}$, where μ_s is a mean of the best case of baseline samples, and μ_{FSD} is a mean of FSD results. We estimate right-tail p-value if $\mu_s \leq \mu_{FSD}$, else left-tail p-value. By the p-value results, FSD shows statistically significant on RTE, STS-B, and MRPC. In the results, the FSD method effectively and stably improves the test accuracy of the benchmark, especially on small datasets. Furthermore, the methods can generalize the student model to show better performance than the teacher models in particular tasks (MRPC and QQP).

Furthermore, Table 4 shows the necessity of an integrated method. These results imply that integrated methods are better than single

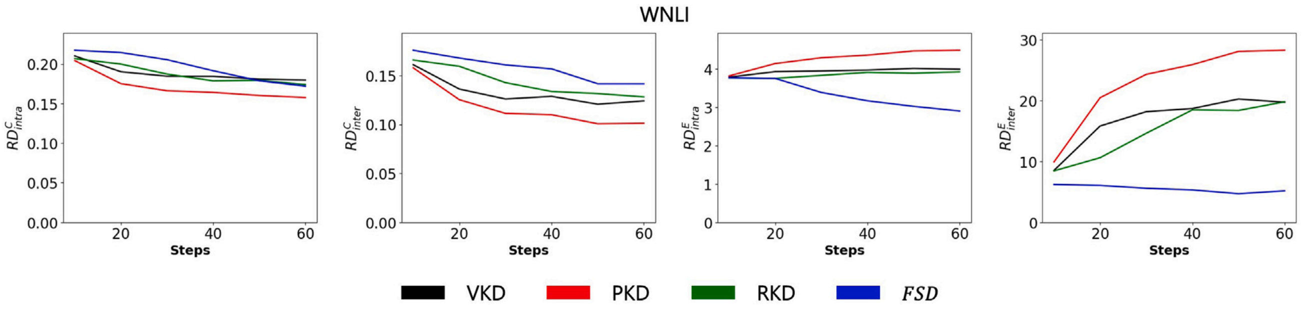

Fig. 6. The results of relation difference. Each task shows the results of using different four difference metrics: RD^C_{intra}, second is RD^C_{inter}, third is RD^E_{intra}, and RD^E_{inter}. Fisrt row, WNLI, results are representative. The results of all the other tasks are in Appendix.

structure distillation, and in particular, applied for all feature structure transfers teacher's knowledge is more effective than others. In particular, the gap between FSD and other methods on small datasets such WNLI, and CoLA is larger than other tasks. These results show that the proposed method is more effective on small datasets.

In the last, the global inter-feature structure preserves teacher feature structure information in memory and stably maintains model performance regardless of mini-batch size. It implies that the global inter-feature structure covers the drawback of the local-level method as shown in Fig. 2.

6.2. *Qualitative analysis*

6.2.1. *KD quality on restoration rate*

In KD, correctly recovering a teacher's prediction affects the quality of transferring. In the restoration rate results analysis, the higher restoration rate implies student imitates teacher results accurately, therefore the proposed method is effective to emulate teachers. Moreover, FSD shows better mimicry of teacher results than the other proposed methods. As shown in the WNLI, RTE, CoLA, and MRPC results, FSD method significantly shows more effective on small datasets than larger datasets.

6.2.2. *Impact of CKA to model performance*

In the comparison of heat map patterns as shown in Fig. 4, the FSD is similar to the T-T heat map, called teacher group. In contrast, VKD, and PKD are more closer to noDS, called noDS group. In the results, the different range of heat maps and patterns show the implicit difference of structures between tasks. Depending on the complexity of transferring the teacher's structures and conflict with students' structures, the performance is heavily affected. The clear distinction of heat map patterns between the teacher and noDS groups implies that the transferred structures largely differ by their types.

As shown in Fig. 5, VKD, PKD, FSD$_I$, and FSD$_G$ are close to nDS, and FSD$_L$ and FSD are located more closer to ideal case (T-T). It shows that VKD and PKD less reflect teacher's knowledge, compare to FSD$_L$ and FSD. In addition, FSD is interpolated by FSD$_I$, FSD$_L$, and FSD$_G$.

As shown in Table 7, the average of FSD RD rank is the highest, but RD structure graph pattern shown in Fig. 6 is not consistent in all GLUE task. On the other hand, CKA heat map diagonals in Fig. 4 are consistent regardless of tasks. As referred on the Results section, CKA heat map could be a clue to explain the best case of method (FSD). Therefore, CKA analysis is more related metric than Euclidean distance and cosine similarity to explain KD model performance.

6.2.3. *Transferred structure on traditional perspective*

The RD results of WNLI task show that the proposed method preserves structures on RD^E but unstably transfers the structures on RD^C because of the CKA property to preserve Euclidean distance and dot product.

The higher ranks of FSD method in most tasks implies that CKA similarity is effective to preserve structures on RD metrics. RKD, the best among baseline, is still significantly worse than FSD even if it spends larger computational cost for evaluating $O(n^3)$ relations than $O(n^2)$ of FSD method. In sum, FSD using CKA similarity effectively transfers teachers' potential structures defined by various difference metrics with relatively good computational efficiency.

7. Conclusion

In this paper, we addressed transferring rich information of feature representations via knowledge distillation in BERT. To represent the features, we proposed three levels of feature structures defined by CKA similarity: intra-feature, local inter-feature, and global inter-feature structures. To transfer them, we implemented **feature structure distillation** methods separately for the structures, especially for the global structures using memory-augmented transferring with clustering. We find that transferring all the structures induces more similar student representations to its teacher and consistently improves performance in the GLUE language understanding tasks. This work can be extended to more downstream applications using pre-trained BERT for more fine transferring.

CRediT authorship contribution statement

Hee-Jun Jung: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision. **Doyeon Kim:** Software. **Seung-Hoon Na:** Conceptualization, Writing – review & editing. **Kangil Kim:** Conceptualization, Methodology, Resources, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

There are no financial interests/personal relationships which may be considered as potential competing interests.

Data availability

I have shared the link to my code at github (please check abstract).

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2022R1A 2C2012054).

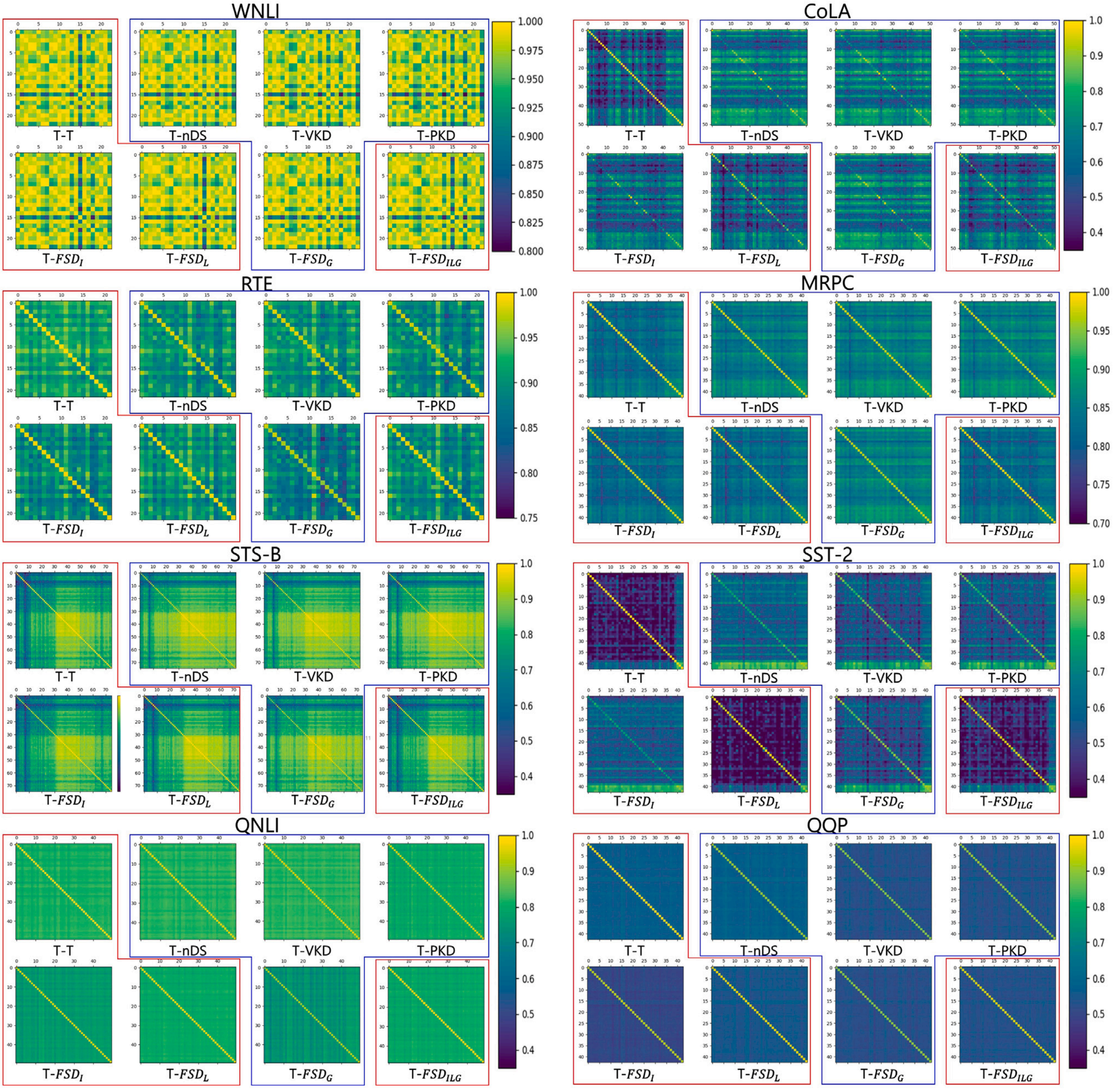

Fig. 7. CKA heatmap of GLUE benchmark. Red box represents teacher closed methods and blue represents no-Distillation-Student closed methods.

Appendix

A.1. CKA heat maps of all GLUE benchmark results

This material is to show the full results of Fig. 4 in the manuscript.

As shown in Fig. 7, overall diagonal lines are lighter in WNLI, RTE, STS-B, MRPC, QNLI, and QQP than CoLA, SST-2, and MNLI, which shows the task-specific difference between teacher and student knowledge. Normally, patterns are divided into teacher closed (FSD_I, FSD_L, and FSD) and no-Distillation-Student closed models (VKD, PKD, and FSD_G), and it is clearly shown on CoLA, RTE, MRPC, and SST-2. These results still show that the proposed method is more effective on small datasets.

A.2. Relation difference of all GLUE benchmark result

This material is to show the full results of Fig. 6 in the manuscript. In Fig. 8, similar patterns are shown in overall tasks. In some cases as COLA, SST-2, and STS-B, Euclidean distance also largely decreases in FSD compared to baselines.

The distinctive patterns in FSD implies that the knowledge transferred by it is all different, which explains why the integration shows the best performance.

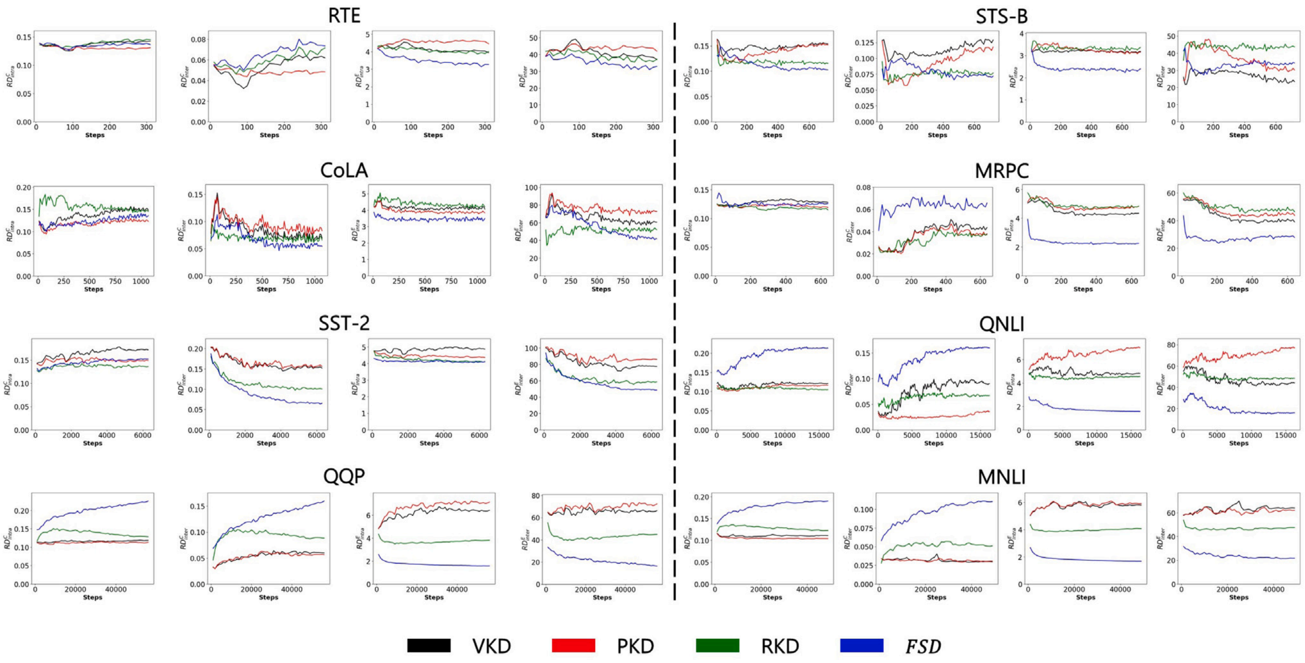

Fig. 8. The results of relation difference. Each task shows the results of using different four difference metrics: RD^C_{intra}, second is RD^C_{inter}, third is RD^E_{intra}, and RD^E_{inter}.

References

Bentivogli, L., Magnini, B., Dagan, I., Dang, H. T., & Giampiccolo, D. (2009). The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the second text analysis conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST, URL: https://tac.nist.gov/publications/2009/additional.papers/RTE5_overview.proceedings.pdf.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I. n., & Specia, L. (2017). SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)* (pp. 1–14). Vancouver, Canada: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/S17-2001.

Cortes, C., Mohri, M., & Rostamizadeh, A. (2012). Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, *13*(1), 795–828.

Dagan, I., Glickman, O., & Magnini, B. (2006). The PASCAL recognising textual entailment challenge. In *Machine learning challenges. Evaluating predictive uncertainty, visual object classification, and recognising tectual entailment* (pp. 177–190). Berlin, Heidelberg: Springer Berlin Heidelberg.

Dolan, W. B., & Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the third international workshop on paraphrasing (IWP2005)*. URL: https://aclanthology.org/I05-5002.

Giampiccolo, D., Magnini, B., Dagan, I., & Dolan, B. (2007). The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing* (pp. 1–9). USA: Association for Computational Linguistics.

Golub, G. H., & Zha, H. (1995). The canonical correlations of matrix pairs and their numerical computation. In *Linear algebra for signal processing* (pp. 27–49). New York, NY: Springer New York.

Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, *129*(6), 1789–1819. http://dx.doi.org/10.1007/s11263-021-01453-z.

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic learning theory* (pp. 63–77). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. URL: http://arxiv.org/abs/1503.02531, cite arxiv:1503.02531 Comment: NIPS 2014 Deep Learning Workshop.

Hotelling, H. (1992). Relations between two sets of variates. In S. Kotz, & N. L. Johnson (Eds.), *Breakthroughs in statistics: Methodology and distribution* (pp. 162–190). New York, NY: Springer New York, http://dx.doi.org/10.1007/978-1-4612-4380-9_14.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., et al. (2020). TinyBERT: distilling BERT for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020* (pp. 4163–4174). http://dx.doi.org/10.18653/v1/2020.findings-emnlp.372.

Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019). Similarity of neural network representations revisited. In K. Chaudhuri, & R. Salakhutdinov (Eds.), *Proceedings of machine learning research*: *vol. 97*, *Proceedings of the 36th international conference on machine learning* (pp. 3519–3529). PMLR.

Levesque, H. J., Davis, E., & Morgenstern, L. (2012). The winograd schema challenge. In *Proceedings of the thirteenth international conference on principles of knowledge representation and reasoning* KR '12, (pp. 552–561). AAAI Press.

Li, X., Wu, J., Fang, H., Liao, Y., Wang, F., & Qian, C. (2020). Local correlation consistency for knowledge distillation. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), *Computer vision – ECCV 2020* (pp. 18–33). Cham: Springer International Publishing.

Liu, Y., Cao, J., Li, B., Yuan, C., Hu, W., Li, Y., et al. (2019). Knowledge distillation via instance relationship graph. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 7089–7097). http://dx.doi.org/10.1109/CVPR.2019.00726.

Morcos, A., Raghu, M., & Bengio, S. (2018). Insights on representational similarity in neural networks with canonical correlation. In *Advances in neural information processing systems, vol. 31* (pp. 5727–5736). Curran Associates, Inc..

Park, W., Kim, D., Lu, Y., & Cho, M. (2019). Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Park, G., Kim, G., & Yang, E. (2021). Distilling linguistic context for language model compression. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 364–378). http://dx.doi.org/10.18653/v1/2021.emnlp-main.30.

Peng, B., Jin, X., Li, D., Zhou, S., Wu, Y., Liu, J., et al. (2019). Correlation congruence for knowledge distillation. In *2019 IEEE/CVF international conference on computer vision (ICCV)* (pp. 5006–5015). http://dx.doi.org/10.1109/ICCV.2019.00511.

Raghu, M., Gilmer, J., Yosinski, J., & Sohl-Dickstein, J. (2017). Svcca: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in neural information processing systems, vol. 30* (pp. 6076–6085). Curran Associates, Inc..

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 2383–2392). Austin, Texas: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D16-1264.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *5th Workshop on energy efficient machine learning and cognitive computing @ NeurIPS 2019*. arXiv:1910.01108, URL: http://arxiv.org/abs/1910.01108.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., et al. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631–1642). Seattle, Washington, USA: Association for Computational Linguistics.

Sun, S., Cheng, Y., Gan, Z., & Liu, J. (2019). Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)* (pp. 4323–4332). http://dx.doi.org/10.18653/v1/D19-1441.

Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., & Zhou, D. (2020). MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 2158–2170). http://dx.doi.org/10.18653/v1/2020.acl-main.195.

Tung, F., & Mori, G. (2019). Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems, vol. 30* (pp. 5998–6008). Curran Associates, Inc..

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). GLUE: a multi-task benchmark and analysis platform for natural language understanding. In *International conference on learning representations*. URL: https://openreview.net/forum?id=rJ4km2R5t7.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in neural information processing systems, vol. 33* (pp. 5776–5788). Curran Associates, Inc..

Warstadt, A., Singh, A., & Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, *7*, 625–641. http://dx.doi.org/10.1162/tacl_a_00290.

Williams, A., Nangia, N., & Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human language technologies, volume 1 (Long Papers)* (pp. 1112–1122). New Orleans, Louisiana: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N18-1101.

Wu, J., Yu, S., Chen, W., Ma, K., Fu, R., Liu, H., et al. (2020). Leveraging undiagnosed data for glaucoma classification with teacher-student learning. In *Medical image computing and computer assisted intervention – MICCAI 2020* (pp. 731–740). Cham: Springer International Publishing.

Xu, C., Zhou, W., Ge, T., Xu, K., McAuley, J., & Wei, F. (2021). Beyond preserved accuracy: Evaluating loyalty and robustness of BERT compression. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 10653–10659). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.emnlp-main.832.

Yuan, L., Tay, F. E. H., Li, G., Wang, T., & Feng, J. (2020). Revisiting knowledge distillation via label smoothing regularization. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 3902–3910).

Zhao, B., Xing, H., Wang, X., Song, F., & Xiao, Z. (2023). Rethinking attention mechanism in time series classification. *Information Sciences*, *627*, 97–114. http://dx.doi.org/10.1016/j.ins.2023.01.093, URL: https://www.sciencedirect.com/science/article/pii/S0020025523000968.