How to Distill your BERT: An Empirical Study on the Impact of Weight Initialisation and Distillation Objectives

Xinpeng Wang* **Leonie Weissweiler*◇** **Hinrich Schütze*◇** **Barbara Plank*◇**
*Center for Information and Language Processing (CIS), LMU Munich, Germany
◇Munich Center for Machine Learning (MCML), Munich, Germany
{xinpeng, weissweiler, bplank}@cis.lmu.de

Abstract

Recently, various intermediate layer distillation (ILD) objectives have been shown to improve compression of BERT models via Knowledge Distillation (KD). However, a comprehensive evaluation of the objectives in both task-specific and task-agnostic settings is lacking. To the best of our knowledge, this is the first work comprehensively evaluating distillation objectives in both settings. We show that attention transfer gives the best performance overall. We also study the impact of layer choice when initializing the student from the teacher layers, finding a significant impact on the performance in task-specific distillation. For vanilla KD and hidden states transfer, initialisation with lower layers of the teacher gives a considerable improvement over higher layers, especially on the task of QNLI (up to an absolute percentage change of 17.8 in accuracy). Attention transfer behaves consistently under different initialisation settings. We release our code as an efficient transformer-based model distillation framework for further studies.[1]

1 Introduction

Large-scale pre-trained language models (PLMs) have brought revolutionary advancements to natural language processing, such as BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), ELECTRA (Clark et al., 2020) and GPT-3 (Brown et al., 2020). However, the enormous size of these models has led to difficulties in deploying them in resource-constrained environments. Therefore significant interest has emerged in developing methods for reducing their size.

Knowledge Distillation (KD) (Hinton et al., 2015) transfers the knowledge embedded in one model to another, which can be used for cross-lingual transfer, cross-modal transfer, and model compression. KD heavily depends on the distillation objective, which determines how knowledge is transferred. Many works have tried to design different distillation objectives for Transformer-based (Vaswani et al., 2017) model compression and successfully distilled PLMs into smaller models, either task-specifically (Sun et al., 2019a; Jiao et al., 2020) or task-agnostically—which differ in whether KD is performed at the pre-training stage or during task finetuning (Sanh et al., 2019; Sun et al., 2020b; Wang et al., 2020; Wang et al., 2021).

Despite their impressive results, determining the best distillation objective is difficult due to their diverse comparison setups, such as data preprocessing, student model initialization, layer mapping strategies, task-specific/agnostic settings, and others. This breadth of choices and lack of code has led to comparison on unequal grounds and contradictory findings.[2] This shows a substantial need to reproduce and evaluate distillation objectives within the same setting. Motivated by this gap, we conduct experiments on the most common distillation objectives and their combinations in task-specific and task-agnostic settings. From our empirical evaluation, we show: (1) attention transfer performs consistently well in various initialisation settings, (2) initialisation with lower layers of the teacher gives a considerable improvement over higher layers in task-specific distillation.

In summary, our **contributions** are:

- We perform an evaluation of the effectiveness of different distillation objectives and the layer choice for initializing the student from the teacher layer.
- We make our code available as an efficient distillation framework.
- We provide practical guidance in terms of teacher layer choice for initialisation, distillation objectives and training parameters.

[1] https://github.com/mainlp/How-to-distill-your-BERT

[2] For example, both Jiao et al. (2020) and Wang et al. (2020) claimed to be the better method in their setting. See section 5 for detail.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 1843–1852
July 9-14, 2023 ©2023 Association for Computational Linguistics

2 Related Work

Task-specific Distillation Sun et al. (2019b) task-specifically compressed BERT by learning from the every k-th layer of the teacher. To avoid leaving out some of the teacher layers, many follow-up works (Wu et al., 2020, Passban et al., 2021, Wu et al., 2021) designed new layer mapping strategies to fuse the teacher layers. Jiao et al. (2020) used data augmentation to further improve the performance. Initialising the student model with pre-trained weights is crucial for performance since the student learns from the teacher only shortly in downstream tasks. Common choices for initialization are: (1) task-agnostically distilling models first, (2) using publicly available distilled models, or (3) initializing with teacher layers. As part of this study, we examine how to maximize the benefits of initializing from teacher layers.

Task-agnostic Distillation In the field of task-agnostic distillation, one line of work is to compress the teacher model into a student model with the same depth but narrower blocks (Sun et al., 2020b, Zhang et al., 2022). Another line of work is to distill the teacher into a student with fewer layers (Sanh et al., 2019, Jiao et al., 2020, Wang et al., 2020, Wang et al., 2021), which is our focus.

Comparative Studies Li et al. (2021) conducted out-of-domain and adversarial evaluation on three KD methods, which used hidden state transfer or data augmentation. Lu et al. (2022) is closely related to our work, where they also evaluated knowledge types and initialisation schemes. However, they did not consider layer choice when initialising from the teacher, and the evaluation was only for task-specific settings. Hence, our work complements theirs.

3 Distillation Objectives

Prediction Layer Transfer Prediction layer transfer minimizes the soft cross-entropy between the logits from the teacher and the student: $\mathcal{L}_{\text{pred}} = \text{CE}\left(\boldsymbol{z}^T/t, \boldsymbol{z}^S/t\right)$, with $\boldsymbol{z}^T$ and $\boldsymbol{z}^S$ the logits from the teacher/student and t is the temperature value.

Following the vanilla KD approach (Hinton et al., 2015), the final training loss is a combination of $\mathcal{L}_{\text{pred}}$ and supervision loss $\mathcal{L}_{\text{ce}}$ (masked language modelling loss $\mathcal{L}_{\text{mlm}}$ in the pertaining stage). We denote this objective as **vanilla KD**.

Hidden States Transfer Hidden states transfer penalizes the distance between the hidden states of specific layers from the teacher and the student. Common choices for the representation are the embedding of the `[CLS]` token (Sun et al., 2019b) and the whole sequence embedding (Jiao et al., 2020). We use Mean-Squared-Error (MSE) to measure the distance between the student and teacher embedding, which can be formulated as $\mathcal{L}_{\text{hid}} = \text{MSE}\left(\boldsymbol{h}^S\boldsymbol{W}_h, \boldsymbol{h}^T\right)$, where $\boldsymbol{h}^S \in \mathbb{R}^d$ and $\boldsymbol{h}^T \in \mathbb{R}^{d'}$ are the `[CLS]` token embedding of specific student and teacher layer, d and d' are the hidden dimensions. The matrix $\boldsymbol{W}_h \in \mathbb{R}^{d\times d'}$ is a learnable transformation. We denote this objective as **Hid-CLS**. In the case of transferring the sequence embedding, one can replace the token embeddings with sequence embeddings $\boldsymbol{H}^S \in \mathbb{R}^{l\times d}$ and $\boldsymbol{H}^T \in \mathbb{R}^{l\times d'}$, where l is the sequence length. The objective that transfers the sequence embedding with MSE loss is denoted as **Hid-Seq**.

We also evaluated a contrastive representation learning method which transfers the hidden state representation from the teacher to the student with a contrastive objective (Sun et al., 2020a). We inherited their code for implementation and refer our readers to the original paper for details. We denote this objective as **Hid-CLS-Contrast**.

Attention and Value Transfer The attention mechanism has been found to capture rich linguistic knowledge (Clark et al., 2019), and attention map transfer is widely used in transformer model distillation. To measure the similarity between the multi-head attention block of the teacher and the student, MSE and Kullback-Leibler divergence are the two standard loss functions. The objective using MSE is formulated as $\mathcal{L}_{\text{att}} = \frac{1}{h}\sum_{i=1}^{h}\text{MSE}(\boldsymbol{A}_i^S, \boldsymbol{A}_i^T)$, where h is the number of attention heads, matrices $\boldsymbol{A}_i \in \mathbb{R}^{l\times l}$ refers to the i-th attention head (before the softmax operation) in the multi-head attention block. We denote this objective as **Att-MSE**.

Since the attention after the softmax function is a distribution over the sequence, we can also use the KL-divergence to measure the distance: $\mathcal{L}_{\text{att}} = \frac{1}{TH}\sum_{t=1}^{T}\sum_{h=1}^{H} D_{KL}(a_{t,h}^T \| a_{t,h}^S)$, where T is the sequence length and H is the number of attention heads. We will denote this objective as **Att-KL**. In addition to attention transfer, value-relation transfer was proposed by Wang et al. (2020), to which we refer our readers for details. Value-relation transfer objective will be denoted as **Val-KL**.

Objectives	QNLI Acc	SST-2 Acc	MNLI Acc	MRPC F1	QQP Acc	RTE Acc	CoLA Mcc	Avg
Vanilla KD	$66.5_{\pm 1.49}$	$84.7_{\pm 0.16}$	$75.1_{\pm 0.05}$	$71.2_{\pm 0.80}$	$81.9_{\pm 0.10}$	$54.0_{\pm 1.24}$	$69.1_{\pm 0.00}$	71.8
Hid-CLS-Contrast	$69.3_{\pm 0.60}$	$85.3_{\pm 0.56}$	$76.2_{\pm 0.45}$	$71.1_{\pm 0.85}$	$83.1_{\pm 0.69}$	$53.6_{\pm 0.23}$	$69.0_{\pm 0.12}$	72.5
Hid-CLS	$75.7_{\pm 0.57}$	$85.8_{\pm 0.34}$	$77.0_{\pm 0.10}$	$71.3_{\pm 0.41}$	$83.8_{\pm 1.63}$	$54.0_{\pm 2.17}$	$68.4_{\pm 0.35}$	73.2
Hid-Seq	$83.3_{\pm 0.13}$	$87.4_{\pm 0.13}$	$78.3_{\pm 0.13}$	$\mathbf{72.9}_{\pm 0.50}$	$87.6_{\pm 0.00}$	$51.8_{\pm 1.10}$	$69.2_{\pm 0.55}$	75.8
Att-MSE	$84.3_{\pm 0.18}$	$89.2_{\pm 0.40}$	$78.6_{\pm 0.25}$	$71.1_{\pm 0.41}$	$88.7_{\pm 0.05}$	$54.4_{\pm 1.03}$	$69.3_{\pm 0.17}$	76.5
+Hid-Seq	$84.6_{\pm 0.29}$	$89.2_{\pm 0.21}$	$78.9_{\pm 0.10}$	$71.8_{\pm 0.51}$	$88.8_{\pm 0.00}$	$54.0_{\pm 0.93}$	$\mathbf{69.5}_{\pm 0.48}$	77.0
Att-KL	$85.3_{\pm 0.14}$	$89.0_{\pm 0.26}$	$79.4_{\pm 0.08}$	$71.4_{\pm 0.29}$	$89.0_{\pm 0.05}$	$55.5_{\pm 2.05}$	$69.3_{\pm 0.13}$	77.0
+Hid-Seq	$84.6_{\pm 0.21}$	$89.1_{\pm 0.46}$	$79.5_{\pm 0.17}$	$72.4_{\pm 0.39}$	$89.0_{\pm 0.06}$	$57.2_{\pm 0.86}$	$69.3_{\pm 0.21}$	77.3
+Val-KL	$\mathbf{85.5}_{\pm 0.24}$	$\mathbf{89.6}_{\pm 0.31}$	$\mathbf{79.6}_{\pm 0.10}$	$72.2_{\pm 0.39}$	$\mathbf{89.1}_{\pm 0.05}$	$\mathbf{57.5}_{\pm 0.70}$	$69.2_{\pm 0.15}$	**77.5**

Table 1: Task-specific distillation results on GLUE dev sets. Student models are initialised with every 4th layer of the teacher model. We report the average and standard deviation over 4 runs. Attention based objectives consistently outperform hidden states transfer and vanilla KD.

Objectives	QNLI Acc	SST-2 Acc	MNLI Acc	MRPC F1	QQP Acc	RTE Acc	CoLA Mcc	Avg
DistilBERT*	89.2	91.3	82.2	87.5	88.5	59.9	51.3	78.5
TinyBERT†	90.5	91.6	83.5	88.4	90.6	72.2	42.8	79.9
MiniLM§	**91.0**	92.0	**84.0**	88.4	**91.0**	**71.5**	49.2	81.0
Vanilla KD*	88.6	91.4	82.4	86.5	90.6	61.0	**54.4**	79.3
Hid-CLS	86.5	90.6	79.3	73.0	89.7	61.0	33.9	73.4
Hid-Seq	89.2	91.5	82.3	89.2	90.3	67.2	48.2	79.7
Att-MSE	89.8	91.6	83.2	90.6	90.7	69.7	53.5	**81.3**
+Hid-Seq†	89.7	**92.4**	82.8	90.4	90.8	68.6	52.8	81.1
Att-KL	88.0	89.7	81.1	90.1	90.3	66.1	43.6	78.4
+Hid-Seq	88.9	91.6	82.4	90.0	90.5	66.8	47.9	79.7
+Val-KL§	89.8	91.6	82.4	**91.0**	90.6	66.7	47.7	80.0

Table 2: Task-agnostic distillation: Performance on GLUE dev sets of three existing distilled 6-layer Transformer models and our 6-layer students distilled. All the students are randomly initialised and distilled from $\text{BERT}_{\text{BASE}}$. We report the best fine-tuning result with grid search over learning rate and batch size. Att-MSE performs the best among all the objectives.

4 Experimental Setup

We evaluate our model on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) tasks, including linguistic acceptability (CoLA), sentiment analysis (SST-2), semantic equivalence (MRPC, QQP), and natural language inference (MNLI, QNLI, RTE).

For task-specific distillation, we distill a fine-tuned $\text{RoBERTa}_{\text{BASE}}$ (Liu et al., 2019) into a 3-layer transformer model on each GLUE task, using the Fairseq (Ott et al., 2019) implementation and the recommended hyperparameters presented in Liu et al. (2019). We follow the training procedure from TinyBERT to perform *intermediate layer* and *prediction layer* distillation sequentially for 10 epochs each, freeing us from tuning the loss weights. For intermediate layer distillation, the student learns from the same teacher's layers that were used for initialising the student. In addition, we always initialise the embedding layer with the teacher's embedding layer.

For task-agnostic distillation, we distill the uncased version of $\text{BERT}_{\text{base}}$ into a 6-layer student model, based on the implementation by Izsak et al. (2021). Here we perform last-layer knowledge transfer since we see no improvement when transferring multiple layers in our experiments. We train the student model for 100k steps with batch size 1024, a peaking learning rate of 5e-4 and a maximum sequence length of 128. The distilled student model is then fine-tuned on the GLUE datasets with grid search over batch size {16, 32} and learning rate {1e-5, 3e-5, 5e-5, 8e-5}. We follow the original training corpus of BERT: English Wikipedia and BookCorpus (Zhu et al., 2015).

Objectives	Init.	QNLI Acc	SST-2 Acc	MNLI Acc	MRPC F1	QQP Acc	RTE Acc	CoLA Mcc	Avg
Vanilla KD	4,8,12	66.5$_{\pm 1.49}$	84.7$_{\pm 0.16}$	75.1$_{\pm 0.05}$	71.2$_{\pm 0.80}$	81.9$_{\pm 0.10}$	54.0$_{\pm 1.24}$	69.1$_{\pm 0.00}$	71.8
	1,8,12	82.9$_{\pm 0.31}$	88.5$_{\pm 0.51}$	76.6$_{\pm 0.08}$	71.2$_{\pm 0.88}$	87.8$_{\pm 0.06}$	55.5$_{\pm 1.07}$	70.8$_{\pm 0.29}$	76.2
	1,2,3	**86.2**$_{\pm 0.35}$	**90.4**$_{\pm 0.28}$	**78.7**$_{\pm 0.18}$	**78.6**$_{\pm 0.18}$	**89.8**$_{\pm 0.05}$	**57.1**$_{\pm 1.46}$	**74.9**$_{\pm 0.54}$	**79.4**
Hid-CLS-Contrast	4,8,12	69.3$_{\pm 0.60}$	85.3$_{\pm 0.56}$	76.2$_{\pm 0.45}$	71.1$_{\pm 0.85}$	83.1$_{\pm 0.69}$	53.6$_{\pm 0.23}$	69.0$_{\pm 0.12}$	72.5
	1,8,12	82.9$_{\pm 0.36}$	88.6$_{\pm 0.29}$	77.0$_{\pm 0.58}$	72.8$_{\pm 0.61}$	88.0$_{\pm 0.13}$	55.4$_{\pm 0.75}$	70.4$_{\pm 0.30}$	76.4
	1,2,3	**86.1**$_{\pm 0.22}$	**89.6**$_{\pm 0.38}$	**79.0**$_{\pm 0.12}$	**73.9**$_{\pm 1.43}$	**90.1**$_{\pm 0.10}$	**55.1**$_{\pm 0.67}$	**71.1**$_{\pm 1.09}$	**77.8**
Hid-CLS	4,8,12	75.7$_{\pm 0.57}$	85.8$_{\pm 0.34}$	77.0$_{\pm 0.10}$	71.3$_{\pm 0.41}$	83.8$_{\pm 1.63}$	54.0$_{\pm 2.17}$	68.4$_{\pm 0.35}$	73.2
	1,8,12	83.4$_{\pm 0.15}$	88.1$_{\pm 0.38}$	77.7$_{\pm 0.10}$	71.9$_{\pm 0.10}$	88.6$_{\pm 0.06}$	56.1$_{\pm 0.88}$	71.5$_{\pm 0.40}$	76.7
	1,2,3	**85.7**$_{\pm 0.05}$	**90.3**$_{\pm 0.29}$	**78.6**$_{\pm 0.14}$	**74.3**$_{\pm 1.00}$	**90.1**$_{\pm 0.00}$	**57.1**$_{\pm 1.37}$	**73.6**$_{\pm 0.24}$	**78.5**
Hid-Seq	4,8,12	83.3$_{\pm 0.13}$	87.4$_{\pm 0.13}$	78.3$_{\pm 0.13}$	72.9$_{\pm 0.50}$	87.6$_{\pm 0.00}$	51.8$_{\pm 1.10}$	69.2$_{\pm 0.55}$	75.8
	1,8,12	84.3$_{\pm 0.10}$	88.6$_{\pm 0.28}$	78.2$_{\pm 0.08}$	72.0$_{\pm 0.70}$	88.6$_{\pm 0.10}$	55.2$_{\pm 1.40}$	71.6$_{\pm 0.37}$	77.6
	1,2,3	**85.9**$_{\pm 0.24}$	**90.7**$_{\pm 0.08}$	**78.9**$_{\pm 0.10}$	**75.5**$_{\pm 1.14}$	**90.0**$_{\pm 0.05}$	**56.6**$_{\pm 0.74}$	**74.2**$_{\pm 0.45}$	**78.8**
Att-KL	4,8,12	85.3$_{\pm 0.14}$	89.0$_{\pm 0.26}$	**79.4**$_{\pm 0.08}$	71.4$_{\pm 0.29}$	89.0$_{\pm 0.05}$	55.5$_{\pm 2.05}$	69.3$_{\pm 0.13}$	77.0
	1,8,12	84.7$_{\pm 0.26}$	**89.6**$_{\pm 0.13}$	78.2$_{\pm 0.10}$	**72.5**$_{\pm 0.24}$	88.6$_{\pm 0.08}$	56.5$_{\pm 0.44}$	**70.4**$_{\pm 0.26}$	77.2
	1,2,3	**86.2**$_{\pm 0.06}$	88.6$_{\pm 0.19}$	77.9$_{\pm 0.17}$	71.3$_{\pm 0.24}$	**89.0**$_{\pm 0.05}$	**61.2**$_{\pm 0.72}$	69.5$_{\pm 0.80}$	**77.7**
Att-MSE	4,8,12	84.3$_{\pm 0.18}$	89.2$_{\pm 0.40}$	**78.6**$_{\pm 0.25}$	71.1$_{\pm 0.41}$	88.7$_{\pm 0.05}$	54.4$_{\pm 1.03}$	69.3$_{\pm 0.17}$	76.5
	1,8,12	84.3$_{\pm 0.25}$	**89.8**$_{\pm 0.39}$	77.5$_{\pm 0.14}$	**72.5**$_{\pm 1.36}$	88.4$_{\pm 0.05}$	57.2$_{\pm 0.96}$	**70.6**$_{\pm 0.45}$	77.2
	1,2,3	**86.2**$_{\pm 0.13}$	88.2$_{\pm 0.43}$	77.8$_{\pm 0.13}$	72.4$_{\pm 0.49}$	**88.8**$_{\pm 0.00}$	**60.3**$_{\pm 1.49}$	69.6$_{\pm 0.90}$	**77.6**

Table 3: Task-specific distillation: Performance of the student initialised with different teacher layers over 4 runs. For vanilla KD and Hid-CLS transfer, the performance on QNLI is significantly improved when initialising with lower teacher layers. Attention transfer benefits less from initialising from lower teacher layers.

5 Results

Distillation Objectives Distillation objective performances are compared in Table 1 and Table 2 for task-specific and task-agnostic settings, respectively. In the task-specific setting, attention transfer is the best choice with initialisation from every k-th teacher layer. However, the performance of hidden states transfer and *vanilla KD* can be drastically improved under other initialisation settings, which we discuss in the next section.

In the task-agnostic setting, the *Att-MSE* objective outperforms *Att-KL*, which performs similarly to *vanilla KD* and hidden states transfer. This contradicts the observation in MiniLM (Wang et al., 2020), where their *Att-KL* based objective outperforms TinyBERT (Jiao et al., 2020) with *Att-MSE*. However, MiniLM has more training iterations and a larger batch size, which makes comparison difficult. The performance drop of *Att-KL* compared to *Att-MSE* is mainly due to its poor performance on CoLA (linguistic acceptability of a sentence), on which MiniLM also performs poorly. We hypothesise that MSE can transfer the linguistic knowledge embedded in the attention matrix more effectively because the MSE loss function gives more direct matching than KL-divergence, which was also concluded by Kim et al. (2021).

For reference, we report the result of 3 existing works that use the same objectives in our experiments. The result of DistilBERT and MiniLM are taken from the respective papers. The result of TinyBERT is taken from Wang et al. (2020) for fair comparison since TinyBERT only reported task-specific distillation result with data augmentation. We denote the prior works and the corresponding objective we evaluate with the same superscript symbol.

Initialisation We also studied the impact of the choice of teacher layers for initialising the student. Evaluation score on GLUE task development sets under different teacher layer choices for initialisation are reported in Table 3 and Table 4 for task-specific and task-agnostic distillation, respectively.

We observe that initiatlization of layers has a huge impact in the task-specific setting. The performance of *vanilla KD* and Hidden states transfer was significantly improved when initialising from lower layers of the teacher (e.g. from 68.1% to 85.9% on QNLI for Vanilla KD). This explains the impressive result of PKD (Sun et al., 2019b), which initialised the student with first k teacher layers. We believe this is an important observation that will motivate further research into investigating the effectiveness of the different layers of the pre-trained transformer model.

In the task-agnostic setting, we only observe

Objectives	Init.	QNLI Acc	SST-2 Acc	MNLI Acc	MRPC F1	QQP Acc	RTE Acc	CoLA Mcc	Avg
Vanilla KD	random	**88.6**	**91.4**	**82.4**	86.5	90.6	61.0	54.4	79.3
	first 6	88.3	91.2	82.2	**87.0**	90.6	**62.8**	**55.4**	**79.6**
Hid-CLS	random	86.5	90.6	79.3	73.0	89.7	61.0	33.9	73.4
	first 6	**87.0**	**91.2**	**80.7**	**88.0**	**90.2**	**66.0**	**42.5**	**77.9**
Hid-Seq	random	**89.2**	91.5	82.3	89.2	90.3	**67.2**	48.2	79.7
	first 6	87.5	91.5	82.3	**90.0**	**90.5**	66.4	**50.6**	**79.9**
Att-MSE	random	**89.8**	91.6	**83.2**	90.6	90.7	**69.7**	**53.5**	**81.3**
	first 6	89.5	**91.7**	82.8	**91.0**	**90.8**	66.1	53.4	80.8

Table 4: Task-agnostic distillation: Performance of the student initialised with random weights vs first 6 teacher layers. Attention transfer performs the best in both initialisation settings.

considerable improvement with the objective *Hid-CLS*, which performs poorly when randomly initialized, compared to other objectives. This contradicts Sanh et al. (2019) with a *vanilla KD* objective, where they instead showed improvement of 3 average score when initialising from the teacher over random initialisation. However, our *vanilla-KD* approach initialised with random weights outperforms their best result (79.3 vs 78.5). Therefore, we hypothesise that the advantage of pre-loading teacher layers over random initialisation diminishes as the student is fully distilled during pre-training.

Significance Test We conducted paired t-testing for all the distillation objectives in Table 1 and the three initialisation choices within each objective in Table 3. For Table 1, all the pairs of objectives are statistically significant ($p < 0.05$) except four: (Att-KL, Att-MSE), (Att-KL, Att-KL + Hid-Seq), (Att-KL, Att-MSE + Hid-Seq), (Att-MSE, Att-MSE + Hid-Seq). This further supports our conclusion that when initialised from every K teacher layer, it is important to do attention transfer, and the specific objective matters less. For Table 3, all three initialisation choices are statistically significantly different from each other for all the objectives, except the pair (1,8,12, 1,2,3) for Att-KL and Att-MSE, which indicates the robustness of attention transfer under different initialisation choices.

Training Time Since task-agnostic distillation is computationally expensive, we also focus on optimizing our distillation framework for faster training. Our training time is about 58 GPU hours on 40GB A100, compared to TinyBERT (576 GPU hours on 16GB V100) and DistilBERT (720 GPU hours on 16GB V100). This is achieved by using a shorter sequence length and an optimized transformer pre-training framework by Izsak et al. (2021). We see no improvement when using a longer sequence length of 512.

Guidance To sum up, our observations, trade-offs and recommendations are:

- For task-specific KD, we recommend attention transfer in general, due to its consistently high performance in various initialisation settings (Table 3). The exact attention distillation objective matter less (Table 1). Considering the excellent performance of the vanilla KD approach (Table 3) when initialising with lower teacher layers, we also recommend lower teacher layer initialisation with the vanilla KD approach for its shorter training time and simple implementation.
- For task-agnostic KD, attention transfer with Mean-Squared-Error is the best choice based on our result (Table 2, 4).
- We recommend readers to use our task-agnostic distillation framework and short sequence length for fast training.

6 Conclusion

We extensively evaluated distillation objectives for the transformer model and studied the impact of weight initialisation. We found that attention transfer performs consistently well in both task-specific and task-agnostic settings, regardless of the teacher layers chosen for student initialization. We also observed that initialising with lower teacher layers significantly improved task-specific distillation performance compared to higher layers. We release our code and hope this work motivates further research into developing better distillation objectives and compressing in-house models.

7 Limitations

We evaluated the most widely used distillation objectives including prediction layer transfer, hidden states transfer and attention transfer. However, some objectives are not included in our evaluation due to missing implementation details in their paper. For example, we only implemented the contrastive intermediate layer distillation objective proposed by Sun et al. (2020a) in task-specific setting, since code and implementation details are missing for task-agnostic setting. New objectives are increasingly appearing for model compression in the field of computer vision, such as Wasserstein contrastive representation distillation (Chen et al., 2021) and distillation with Pearson correlation (Huang et al., 2022), which can be included to have a broader scope of distillation objectives evaluation.

This work empirically studied the impact of the teacher layer choice for initialization and training objectives, however, further analysis is needed to understand why lower teacher layers are essential for initialisation, and why attention transfer behaves consistently well under various teacher layer choices in the task-specific setting, while hidden state transfer does not.

Acknowledgements

We thank the anonymous reviewers as well as the members of the MaiNLP research lab for their constructive feedback. This research is supported by ERC Consolidator Grant DIALECT 101043235.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin. 2021. Wasserstein contrastive representation distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16296–16305.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *stat*, 1050:9.

Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. 2022. Knowledge distillation from a stronger teacher. *arXiv preprint arXiv:2205.10536*.

Peter Izsak, Moshe Berchansky, and Omer Levy. 2021. How to train BERT with an academic budget. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10644–10652, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Taehyeon Kim, Jaehoon Oh, Nakyil Kim, Sangwook Cho, and Se-Young Yun. 2021. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. In *IJCAI*.

Tianda Li, Ahmad Rashid, Aref Jafari, Pranav Sharma, Ali Ghodsi, and Mehdi Rezagholizadeh. 2021. How to select one among all ? an empirical study towards the robustness of knowledge distillation in natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 750–762, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Chengqiang Lu, Jianwei Zhang, Yunfei Chu, Zhengyu Chen, Jingren Zhou, Fei Wu, Haiqing Chen, and Hongxia Yang. 2022. Knowledge distillation of transformer-based language models revisited. *arXiv preprint arXiv:2206.14366*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. 2021. Alp-kd: Attention-based layer projection for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13657–13665.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019a. 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)*, pages 4323–4332, Hong Kong, China. Association for Computational Linguistics.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019b. 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)*, pages 4323–4332.

Siqi Sun, Zhe Gan, Yuwei Fang, Yu Cheng, Shuohang Wang, and Jingjing Liu. 2020a. Contrastive distillation on intermediate representations for language model compression. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 498–508, Online. Association for Computational Linguistics.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020b. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Yimeng Wu, Peyman Passban, Mehdi Rezagholizadeh, and Qun Liu. 2020. Why skip if you can combine: A simple knowledge distillation technique for intermediate layers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1016–1021, Online. Association for Computational Linguistics.

Yimeng Wu, Mehdi Rezagholizadeh, Abbas Ghaddar, Md Akmal Haidar, and Ali Ghodsi. 2021. Universal-KD: Attention-based output-grounded intermediate layer knowledge distillation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7649–7661, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Xiaofan Zhang, Zongwei Zhou, Deming Chen, and Yu Emma Wang. 2022. Autodistill: an end-to-end framework to explore and distill hardware-efficient language models. *arXiv preprint arXiv:2201.08539*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Hyperparameters

Table 5 shows the hyperparameters we use for task-agnostic distillation.

Hyperparameter	Our Model
Number of Layers	6
Hidden Size	768
FFN inner hidden size	3072
Attention heads	12
Attention head size	64
Learning Rate Decay	Linear
Weight Decay	0.01
Optimizer	AdamW
Adam ϵ	1e-6
Adam β_1	0.9
Adam β_2	0.99
Gradient Clipping	0.0
Warmup Proportion	6%
Peak Learning Rate	5e-4
Batch size	1024
Max Steps	100k

Table 5: Hyperparameter used for distilling our student model in the pre-training stage.

Hyperparameter	Search Space
Learning Rate	{1e-5, 3e-5, 5e-5, 8e-5}
Batch Size	{16, 32}

Table 6: The hyperparameter space used for fine-tuning our distilled student model on GLUE benchmark tasks.

As the distillation in the pre-training stage is computationally expensive and unstable, we suggest readers to follow our settings to avoid additional costs. For example, we observed training loss divergence when using a higher learning rate (1e-3).

Table 6 shows the search space of learning rate and batch size for fine-tuning the general-distilled student. We finetune for 10 epochs on each GLUE task.

For task-specific distillation, we follow the suggested hyperparameters shown in the repository of RoBERTa (Liu et al., 2019).

B Comparison to prior works

Table 7 compares the settings and computational costs of three prior works: DistilBERT (Sanh et al., 2019), TinyBERT (Jiao et al., 2020) and MiniLM (Wang et al., 2020), with our best-performing objective. There are some differences between our settings and theirs, such as layer matching strategies (which teacher layers to transfer), initialisation choices, training steps and batch size. Comparatively, our framework requires less training time and can achieve comparable or better results. Our training takes 58 GPU hours on A100 compared to 720 GPU hours on V100 for training DistilBERT (taking into consideration that an A100 GPU is about twice as fast as a V100).

	Iteration Steps	Batch Size	Layer Matching	Initialisation	Max Sequence Length	GPU hours	Avg-score
DistilBERT	-	4k	prediction layer	every second teacher layer	512	720h on 16GB V100	78.5
TinyBERT	-	256	every second hidden layer	random	128	576h on 16GB V100*	79.9
MiniLM	400k	1024	last hidden layer	random	512	-	81.0
Ours	100k	1024	last hidden layer	random	128	58h on 40GB A100	**81.3**

Table 7: Comparison of hyperparameter choices and training time between ours and prior works. Empty entries indicate that the papers do not report those numbers. $\star$: Number according to their GitHub issue answer.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☑ A2. Did you discuss any potential risks of your work?
appendix A

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

4

☑ B1. Did you cite the creators of artifacts you used?
4,5

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☑ Did you run computational experiments?

Left blank.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.