A APPENDIX

A.1 DATA

Continual Pre-training. We use the same pre-training data as BERT: Wikipedia (English Wikipedia dump8 ; 12GB) and BookCorpus ((Zhu et al., 2015)) (6GB). We clean the corpus by removing tables, lists and references following BERT. We then pre-process the cleaned corpus by concatenating all sentences in a paragraph and truncating the concatenated passage by length of 128 following TinyBERT (Jiao et al., 2019)[4]. We tokenize the corpus with the vocabulary of BERT (30k).

Fine-tuning. GLUE is a commonly used natural language understanding benchmark containing nine tasks. The benchmark includes question answering (Rajpurkar et al., 2016b), linguistic acceptability (CoLA, Warstadt et al. 2019), sentiment analysis (SST, Socher et al. 2013), text similarity (STS-B, Cer et al. 2017), paraphrase detection (MRPC, Dolan & Brockett 2005), and natural language inference (RTE & MNLI, Dagan et al. 2006; Bar-Haim et al. 2006; Giampiccolo et al. 2007; Bentivogli et al. 2009; Williams et al. 2018) tasks. Details of the GLUE benchmark, including tasks, statistics, and evaluation metrics, are summarized in Table 9. SQuaD v1.1/v2.0 are the Stanford Question Answering Datasets (Rajpurkar et al., 2018; 2016a), two popular machine reading comprehension benchmarks from approximately 500 Wikipedia articles with questions and answers obtained by crowdsourcing. The SQuAD v2.0 dataset includes unanswerable questions about the same paragraphs.

A.2 CONTINUAL PRE-TRAINING IMPLEMENTATIONS

Table 6 presents the hyper-parameter configurations for continual pre-training HomoBERT models on the open-domain data. We set the distillation temperature as 2. We empirically observe setting t_f within the range of $0.5T \sim 0.9T$ can achieve similarly good downstream performances. Furthermore, we observe that different weight modules may prefer different t_fs: 1) it is better to finish the pruning of output projection matrices in the attention module, the feed-forward module and the embedding module early, because pruning them late will induce a large increment in distillation loss and the student performance is difficult to recover. 2) the student performance is less sensitive to the pruning of key and query projection matrices in the attention module and the input projection matrix in the feed-forward module, and can often easily recover. Based on this observation, we set $t_f = 0.5$ for the output projection matrices in the attention module, the feed-forward module and the embedding module. For key and query projection matrices in the attention module and the input projection matrix in the feed-forward module, we set $t_f = 0.9$. For other matrices, we set $t_f = 0.7$. This configuration brings around a small and consistent gain of $0.05 \sim 0.08$ on GLUE. The continual pre-training experiment runs for around 13 hours on 8 Nvidia A100 GPUs.

Table 6: Hyper-parameter configurations for task-agnostic distillation of HomoBERT models.

Hyper-parameters	HomoBERT-base	HomoBERT-s	HomoBERT-xs	HomoBERT-tiny
Learning Rates	3×10^{-4}	6×10^{-4}	6×10^{-4}	6×10^{-4}
Batch Size	4000			
Training Epochs	3			
Learning Rate Decay	Linear			
Learning Rate Warmup	0.1			
Max Sequence Length	128			
Weight Decay	0.01			
Adam β_1	0.9			
Adam β_2	0.999			
Adam ϵ	1×10^{-6}			
Gradient Clipping	None			
$\alpha_1, \alpha_2, \alpha_3, \alpha_4$	1			
t_i	0			

[4]https://github.com/yinmingjun/TinyBERT/blob/master/pregenerate_training_data.py

A.3 FINE-TUNING IMPLEMENTATIONS

Table 7 presents the hyper-parameter configurations for fine-tuning HomoBERT models on the GLUE benchmark. We fine-tune the MRPC, RTE and STS-B from a fine-tuned MNLI student. All experiments are conducted on 1 Nvidia A100 GPU.

Table 7: Hyper-parameter configurations for fine-tuning HomoBERT models on the GLUE benchmark. "Epoch" refers to the total training epochs; we adopt early-stopping strategy in practice.

Hyper-parameters	HomoBERT-base	HomoBERT-s	HomoBERT-xs	HomoBERT-tiny
Learning Rates	$\{2,3,4,5\} \times 10^{-5}$	$\{4,5,6,7\} \times 10^{-5}$	$\{6,7\} \times 10^{-5}$	$\{6,7\} \times 10^{-5}$
Batch Size	16 for RTE and MRPC; 32 for the others.			
Training Epochs	3 for MNLI and QNLI; 6 for the others.			
Learning Rate Decay	Linear			
Learning Rate Warmup	0.05			
Max Sequence Length	128			
Dropout of Task Layer	0.1			
Weight Decay	0			
Adam β_1	0.9			
Adam β_2	0.999			
Adam ϵ	1×10^{-6}			
Gradient Clipping	1			

Table 8 presents the hyper-parameter configurations for fine-tuning HomoBERT models on the SQuaD v1.1/2.0. All experiments are conducted on 1 Nvidia A100 GPU.

Table 8: Hyper-parameter configurations for fine-tuning HomoBERT models on SQuaD v1.1/2.0.

Hyper-parameters	HomoBERT-s/xs/tiny
Learning Rates	1×10^{-4}
Batch Size	12
Training Epochs	2
Learning Rate Decay	Linear
Learning Rate Warmup	0.2
Max Sequence Length	384
Dropout of Task Layer	0
Weight Decay	0
Adam β_1	0.9
Adam β_2	0.999
Adam ϵ	1×10^{-6}
Gradient Clipping	1

Table 9: Summary of the GLUE benchmark.

Corpus	Task	#Train	#Dev	#Test	#Label	Metrics
Single-Sentence Classification (GLUE)						
CoLA	Acceptability	8.5k	1k	1k	2	Matthews corr
SST	Sentiment	67k	872	1.8k	2	Accuracy
Pairwise Text Classification (GLUE)						
MNLI	NLI	393k	20k	20k	3	Accuracy
RTE	NLI	2.5k	276	3k	2	Accuracy
QQP	Paraphrase	364k	40k	391k	2	Accuracy/F1
MRPC	Paraphrase	3.7k	408	1.7k	2	Accuracy/F1
QNLI	QA/NLI	108k	5.7k	5.7k	2	Accuracy
Text Similarity (GLUE)						
STS-B	Similarity	7k	1.5k	1.4k	1	Pearson/Spearman corr