Vocabulary-level Memory Efficiency for Language Model Fine-tuning

Miles Williams and **Nikolaos Aletras**
University of Sheffield
{mwilliams15, n.aletras}@sheffield.ac.uk

Abstract

The extensive memory footprint of language model (LM) fine-tuning poses a challenge for both researchers and practitioners. LMs use an embedding matrix to represent extensive vocabularies, forming a substantial proportion of the model parameters. While previous work towards memory-efficient fine-tuning has focused on minimizing the number of trainable parameters, reducing the memory footprint of the embedding matrix has yet to be explored. We first demonstrate that a significant proportion of the vocabulary remains unused during fine-tuning. We then propose a simple yet effective approach that leverages this finding to minimize memory usage. We show that our approach provides substantial reductions in memory usage across a wide range of models and tasks. Notably, our approach does not impact downstream task performance, while allowing more efficient use of computational resources.[1]

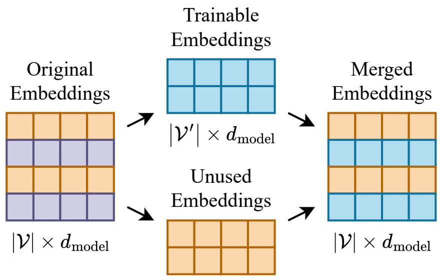

Figure 1: Memory-efficient language model fine-tuning with Partial Embedding Matrix Adaptation (PEMA).

1 Introduction

Language models (LMs) (Chung et al., 2022; Touvron et al., 2023; Warner et al., 2024) form the foundation of contemporary natural language processing (NLP), however they require extensive computational resources to train (Kaplan et al., 2020; Hoffmann et al., 2022). This is contrary to the democratization of NLP, exacerbating economic inequalities and hindering inclusivity (Schwartz et al., 2020; Weidinger et al., 2022). Consequently, there is a growing focus towards developing efficient methods for LM training and fine-tuning (Treviso et al., 2023; Lialin et al., 2023).

The memory footprint of LMs is a major challenge for their application. Storing model parameters requires extensive amounts of memory, constraining the size and architecture of the model (Paleyes et al., 2022). This problem is especially prominent during training as gradients and optimizer states must also be retained (Kingma and Ba, 2017). This can be problematic when using consumer hardware or facing an academic budget (Izsak et al., 2021; Ciosici and Derczynski, 2022).

LMs ordinarily use fixed vocabularies to derive vector representations from text, known as word embeddings. Each element of the vocabulary has a corresponding word embedding, which collectively form an embedding matrix within the LM. The size of the embedding matrix scales with both the vocabulary size and embedding dimension, comprising a substantial proportion of the model parameters (Table 5, Appendix A). This proportion is usually even greater for multilingual LMs, which benefit from larger vocabularies (Conneau et al., 2020; Liang et al., 2023). However, we hypothesize that a significant proportion of LM vocabulary remains unused during fine-tuning on many downstream tasks.

In this paper, we first demonstrate that our hypothesis holds for a variety of downstream tasks, with only a small subset of vocabulary used. We then propose a method to reduce memory usage during fine-tuning by excluding unused embeddings. Finally, we empirically demonstrate the memory savings from our approach across a range of models and tasks. Notably, our approach does not impact downstream task performance and is orthogonal to many existing LM memory efficiency techniques.

[1] https://github.com/mlsw/partial-embedding-matrix-adaptation

Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025), pages 185–196

2 Related Work

Tokenization. Transformer LMs (Vaswani et al., 2017) typically adopt subword tokenization (Schuster and Nakajima, 2012; Sennrich et al., 2016) to encode text using a finite vocabulary. The use of large subword vocabularies enables improved task performance (Gallé, 2019), inference efficiency (Tay et al., 2022), and multilingual performance (Liang et al., 2023). Conversely, character or byte level tokenization can be used (Clark et al., 2022; Xue et al., 2022), reducing the size of the embedding matrix at the cost of increasing the sequence length.

Reducing embedding parameters. To reduce the size of the embedding matrix, LMs can be trained with embedding factorization (Sun et al., 2020; Lan et al., 2020), albeit with slightly lower task performance. Alternatively, embeddings can be generated from hash functions (Sankar et al., 2021; Xue and Aletras, 2022; Cohn et al., 2023), although this may harm performance due to the many-to-one mapping from tokens to embeddings.

Multilingual vocabulary trimming. The closest work to our own is Abdaoui et al. (2020), which creates smaller multilingual LMs by permanently reducing the number of supported languages. This can harm performance as the removed vocabulary may later be required for a downstream task. Moreover, selecting which vocabulary to remove requires the computationally expensive processing of a large corpus. Ushio et al. (2023) further examine the performance impact of permanently removing LM vocabulary either before or after fine-tuning. However, the same fundamental limitations persist.

Parameter-efficient fine-tuning. PEFT methods, such as adapters (Houlsby et al., 2019), soft prompts (Lester et al., 2021; Li and Liang, 2021), ladder side-tuning (Sung et al., 2022), and low-rank adaptation (Hu et al., 2022), effectively adapt LMs by fine-tuning only a small number of parameters. However, these methods still require all LM parameters to be held in accelerator memory.

Offloading. To minimize accelerator (e.g. GPU) memory usage, LM parameters can be held in separate (e.g. CPU) memory until needed (Pudipeddi et al., 2020; Ren et al., 2021). However, this approach substantially increases inference latency.

Model compression. In Appendix B, we discuss a variety of orthogonal LM compression methods, such as quantization, pruning, and distillation.

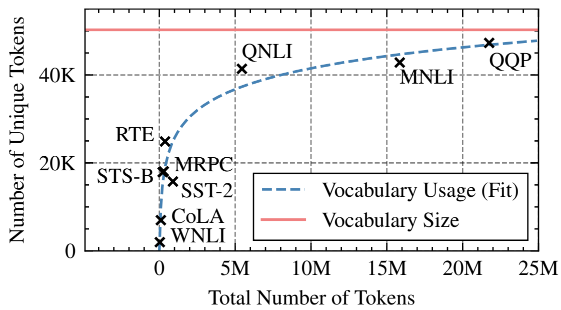

Figure 2: The trend in vocabulary use for the datasets in GLUE when using the vocabulary from GPT-2.

#	Token
49,990	natureconservancy
50,072	;;;;;;;;;;;;
50,160	PsyNetMessage
50,174	rawdownloadcloneembedreportprint
50,243	SolidGoldMagikarp

Table 1: Five examples of tokens from the GPT-2 vocabulary that do not occur within English Wikipedia.

3 Vocabulary Usage Analysis

To empirically assess the level of vocabulary usage during fine-tuning, we first examine the popular GLUE benchmark (Wang et al., 2019). This comprises a series of tasks that are varied in both size and domain (Appendix C). For tokenization, we use the subword vocabulary from GPT-2, which was later adopted by models including RoBERTa (Liu et al., 2019), BART (Lewis et al., 2020), GPT-3 (Brown et al., 2020), and OPT (Zhang et al., 2022).

Figure 2 illustrates the relationship between unique tokens and total tokens in each of the GLUE datasets. Notably, six out of nine datasets fail to use more than half of the vocabulary. Moreover, the smallest dataset, WNLI, uses less than 4%. Interestingly, we observe that the GLUE datasets follow a trend resembling Heaps' Law (Heaps, 1978). This states that as the size of a corpus grows, there are diminishing gains in new vocabulary. However, our use of a finite subword vocabulary means that the trend is asymptotic to the vocabulary size.

Separately, the statistical construction of subword vocabularies can reflect anomalies in their training data, creating tokens that may never be used. To examine the extent of the issue, we identify such tokens by evaluating a processed dump of English Wikipedia, comprising over 20GB of text. Peculiarly, we identify nearly 200 anomalous tokens without a single occurrence (see Table 1).[2]

[2]We refer readers interested in such anomalous tokens to Rumbelow and Watkins (2023) and Land and Bartolo (2024).

4 Partial Embedding Matrix Adaptation

Our empirical analysis (Section 3) suggests that many fine-tuning datasets only use a fraction of LM vocabulary. We leverage this insight to propose Partial Embedding Matrix Adaptation (PEMA), a method that achieves substantial memory savings by selecting only the minimum subset of word embeddings needed for fine-tuning. Notably, this does not impact task performance, as unused word embeddings are not updated during backpropagation.

Preliminaries. Let each token in the vocabulary $\{w_1, \ldots, w_k\}$ be denoted by a unique integer i such that $\mathcal{V} = \{i \in \mathbb{N} \mid i \leq k\}$. The embedding matrix $E \in \mathbb{R}^{|\mathcal{V}| \times d}$ is then used to project each token to a corresponding d-dimensional vector.

Before fine-tuning. Suppose we have fine-tuning dataset $D \in \mathcal{V}^{m \times n}$ where m is the number of examples and n is the length of each example. We compute the partial vocabulary $\mathcal{V}' \subset \mathcal{V}$ consisting of *only* the tokens in D. As the elements of $\mathcal{V}'$ are not necessarily consecutive integers, we define an arbitrary mapping $f \colon \mathcal{V}' \to \{i \in \mathbb{N} \mid i \leq |\mathcal{V}'|\}$. We then construct the partial embedding matrix $E' \in \mathbb{R}^{|\mathcal{V}'| \times d}$ with entries $E'[:, f(i)] = E[:, i]$ for all $i \in \mathcal{V}'$. That is, E' retains only embedding vectors corresponding to tokens in $\mathcal{V}'$. To adapt D for the partial vocabulary $\mathcal{V}'$, we create an intermediary dataset D' where each entry $D'[i, j] = f(D[i, j])$. Finally, we use D' and E' in place of D and E.

After fine-tuning. Following fine-tuning, our partial embedding matrix E' holds the newly learned embeddings for the partial vocabulary. However, we do not wish to keep only the partial vocabulary, as this would limit future use of the model (i.e. tasks with different vocabulary). Therefore, we merge the newly learned embeddings into the original embedding matrix (stored on-disk). More formally, we update E such that $E[:, f^{-1}(i)] = E'[:, i]$ for all $i \in \mathcal{V}'$. This ensures that the model remains structurally identical, with embeddings for the complete vocabulary.

5 Experimental Setup

Datasets. To offer a fair selection of datasets, we follow existing PEFT literature (Houlsby et al., 2019; Hu et al., 2022; Sung et al., 2022; Zhang et al., 2023) and focus our evaluation on the popular GLUE benchmark. We additionally employ XNLI (Conneau et al., 2018) to assess the performance of our approach with multilingual data. Complete data sources and implementation details are listed in Appendix C and Appendix D, respectively.

Models. Similarly, we select a variety of popular models used in existing work. However, we place an emphasis on having a variety of vocabularies (Table 5, Appendix A). For monolingual models, we use BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and DeBERTaV3 (He et al., 2023). For multilingual models, we use mBERT (Devlin et al., 2019), XLM-RoBERTa (Conneau et al., 2020), and XLM-V (Liang et al., 2023). To evaluate the performance of distilled models, we also use the available distilled counterparts: DistilBERT, DistilRoBERTa, and DistilmBERT (Sanh et al., 2020a). For a fair comparison between models, we consistently select the base size ($d_{\text{model}} = 768$).

Memory efficiency metrics. Following convention in the PEFT literature (Houlsby et al., 2019; Hu et al., 2022; Ben Zaken et al., 2022), we report memory efficiency in terms of model parameters. This is advantageous as it avoids confounding factors such as weight precision, optimizer choice, software implementation, and batch size.

6 Results

Larger vocabularies see more memory savings. Table 2 presents the reduction in parameters for each model across the GLUE benchmark. Following our expectations from Section 3, we generally observe that as vocabulary sizes increase (Table 5, Appendix A), so do the potential memory savings. For example, an average reduction in embedding parameters of 47.3% is achieved for BERT, 52.1% for RoBERTa, and 72.4% for DeBERTaV3.

Memory savings vary between datasets. In line with our expectations from Section 3, the memory savings vary substantially between datasets. For BERT, the embedding matrix can be reduced by 94.3% for the smallest dataset (WNLI), yet only 11.5% for the largest (QQP). We demonstrate that downstream task performance remains consistent across models and datasets in Appendix E.

Distilled models substantially benefit. Considering the distilled models, we observe that they all achieve an identical reduction in embedding parameters to their original counterparts. This is because they use the same vocabulary and embedding size (Sanh et al., 2020a). However, they offer substan-

Model	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI	Mean
Reduction in Embedding Parameters (%)										
DistilBERT	80.1	14.8	54.9	13.1	11.5	41.5	57.9	57.2	94.3	47.3
DistilRoBERTa	86.1	14.8	64.0	17.7	5.9	51.6	68.6	64.4	96.0	52.1
DistilmBERT	94.9	76.9	88.2	73.8	72.7	85.0	91.9	88.8	98.4	85.6
BERT	80.1	14.8	54.9	13.1	11.5	41.5	57.9	57.2	94.3	47.3
RoBERTa	86.1	14.8	64.0	17.7	5.9	51.6	68.6	64.4	96.0	52.1
DeBERTaV3	95.0	44.3	85.7	47.1	28.5	79.0	87.5	85.9	98.6	72.4
mBERT	94.9	76.9	88.2	73.8	72.7	85.0	91.9	88.8	98.4	85.6
XLM-RoBERTa	97.8	88.8	94.9	87.6	85.4	93.3	96.3	94.9	99.3	93.1
XLM-V	99.3	93.2	98.0	92.8	90.5	97.1	98.3	98.0	99.8	96.3
Reduction in Model Parameters (%)										
DistilBERT	28.0	5.2	19.2	4.6	4.0	14.5	20.3	20.0	33.0	16.5
DistilRoBERTa	40.5	7.0	30.1	8.3	2.8	24.3	32.3	30.3	45.1	24.5
DistilmBERT	64.4	52.2	59.9	50.1	49.3	57.7	62.3	60.2	66.8	58.1
BERT	17.1	3.2	11.8	2.8	2.5	8.9	12.4	12.2	20.2	10.1
RoBERTa	26.7	4.6	19.8	5.5	1.8	16.0	21.2	19.9	29.7	16.1
DeBERTaV3	50.7	23.6	45.7	25.1	15.2	42.1	46.7	45.8	52.6	38.6
mBERT	49.0	39.7	45.5	38.1	37.5	43.9	47.4	45.8	50.8	44.2
XLM-RoBERTa	67.5	61.3	65.5	60.5	59.0	64.4	66.5	65.5	68.5	64.3
XLM-V	88.3	82.9	87.2	82.6	80.5	86.4	87.5	87.2	88.8	85.7

Table 2: The reduction in embedding and model parameters (%) for each model across the GLUE benchmark.

Size	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI	Mean
XSmall	46.7	21.8	42.2	23.2	14.0	38.8	43.1	42.3	48.5	35.6
Small	93.4	43.6	84.3	46.3	28.0	77.7	86.1	84.5	97.0	71.2
Base	93.4	43.6	84.3	46.3	28.0	77.7	86.1	84.5	97.0	71.2
Large	124.6	58.1	112.4	61.8	37.3	103.6	114.8	112.7	129.4	95.0

Table 3: The reduction in model parameters (millions) for each size of DeBERTaV3 across the GLUE benchmark.

tially higher overall savings, as there are fewer parameters allocated to the transformer layers.

Memory savings scale with model size. Table 3 presents the reduction in model parameters for each model from the DeBERTaV3 family. We observe that this reduction continues to increase with model size. On average, the extra small size is reduced by 35.6M parameters, while the large size is reduced by 95.0M parameters. Although the same fixed-size vocabulary is shared across models, the embedding dimension continues to grow (Table 6, Appendix A), offering further memory savings. The exception to this is the small and base sizes, where the only difference is the number of layers.

Multilingual models achieve extreme savings. Unsurprisingly, multilingual models demonstrate extreme memory savings across the monolingual GLUE benchmark. On average, a reduction in model parameters of 44.2% is achieved for mBERT, 64.3% for XLM-RoBERTa, and 85.7% for XLM-V. Table 4 presents the reduction in parameters for the multilingual models when fine-tuning on different subsets of XNLI. Even when fine-tuning on all fifteen languages, these models still demonstrate substantial memory savings from 23.0% to 58.4%.

Model	en	en-de	en-zh	All
Reduction in Embedding Parameters (%)				
DistilmBERT	77.1	71.7	73.0	44.6
mBERT	77.1	71.7	73.0	44.6
XLM-RoBERTa	89.2	86.0	84.4	56.9
XLM-V	93.6	90.0	90.0	65.7
Reduction in Model Parameters (%)				
DistilmBERT	52.3	48.6	49.6	30.3
mBERT	39.8	37.0	37.7	23.0
XLM-RoBERTa	61.6	59.4	58.3	39.3
XLM-V	83.2	80.0	80.0	58.4

Table 4: The reduction in parameters across different subsets of XNLI, in addition to all fifteen languages.

7 Conclusion

In this paper, we identified that many fine-tuning datasets do not use the majority of LM vocabulary. We then proposed Partial Embedding Matrix Adaptation (PEMA), a simple yet effective approach to minimize LM memory use during fine-tuning, that is orthogonal to many existing methods. Finally, we empirically demonstrated that our approach offers substantial memory savings across a variety of popular tasks and models, without compromising performance. As future work, we are interested in adapting our approach for the output embedding matrix to offer further memory savings.

Limitations

Processing the fine-tuning dataset to assess vocabulary usage incurs a runtime cost. However, we observe that this cost is negligible. We provide a detailed analysis of this matter in Appendix F.

Ethical Considerations

Our approach improves the memory efficiency of LM fine-tuning, therefore facilitating the use of less powerful hardware. Although we hope that this can reduce the environmental footprint of LM fine-tuning, we acknowledge that it could be used to support the fine-tuning of even larger LMs. We also recognize the dual-use nature of LMs and concede that efforts towards improving efficiency, including our own, can lower the barrier to entry for their misuse (Weidinger et al., 2022).

Acknowledgments

We are sincerely grateful to Nafise Sadat Moosavi, Huiyin Xue, Atsuki Yamaguchi, and the anonymous reviewers for their invaluable feedback. MW is supported by the Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UK Research and Innovation grant EP/S023062/1.

References

Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load what you need: Smaller versions of mutililingual BERT. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 119–123, Online. Association for Computational Linguistics.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7947–7969, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 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)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Manuel R. Ciosici and Leon Derczynski. 2022. Training a T5 using lab-sized resources. *Preprint*, arXiv:2208.12097.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Gabrielle Cohn, Rishika Agarwal, Deepanshu Gupta, and Siddharth Patwardhan. 2023. EELBERT: Tiny models through dynamic embeddings. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 451–459, Singapore. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification,* *and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*.

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.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Elias Frantar and Dan Alistarh. 2023. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

Matthias Gallé. 2019. Investigating the effectiveness of BPE: The power of shorter sequences. 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 1375–1381, Hong Kong, China. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Harold Stanley Heaps. 1978. *Information Retrieval: Computational and Theoretical Aspects*. Library and information science series. Academic Press.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, and 3 others. 2022. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030. Curran Associates, Inc.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Shankar Iyer, Nikhil Dandekar, Kornél Csernai, and 1 others. 2017. First Quora dataset release: Question pairs.

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.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Eldar Kurtic, Daniel Campos, Tuan Nguyen, Elias Frantar, Mark Kurtz, Benjamin Fineran, Michael Goin, and Dan Alistarh. 2022. The optimal BERT surgeon: Scalable and accurate second-order pruning for large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4163–4181, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Eldar Kurtic, Torsten Hoefler, and Dan Alistarh. 2024. How to prune your language model: Recovering accuracy on the "sparsity may cry" benchmark. In *Conference on Parsimony and Learning*, volume 234 of *Proceedings of Machine Learning Research*, pages 542–553. PMLR.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Sander Land and Max Bartolo. 2024. Fishing for magikarp: Automatically detecting under-trained tokens in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11631–11646, Miami, Florida, USA. Association for Computational Linguistics.

Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, pages 552–561. AAAI Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, and 13 others. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *Preprint*, arXiv:2303.15647.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore. 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. *Preprint*, arXiv:1907.11692.

Andrei Paleyes, Raoul-Gabriel Urma, and Neil D. Lawrence. 2022. Challenges in deploying machine learning: A survey of case studies. *ACM Comput. Surv.*, 55(6).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Bharadwaj Pudipeddi, Maral Mesmakhosroshahi, Jinwen Xi, and Sujeeth Bharadwaj. 2020. Training large neural networks with constant memory using a new execution algorithm. *Preprint*, arXiv:2002.05645.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. 2021. ZeRO-Offload: Democratizing billion-scale model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 551–564. USENIX Association.

Jessica Rumbelow and Matthew Watkins. 2023. SolidGoldMagikarp (plus, prompt generation).

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020a. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Victor Sanh, Thomas Wolf, and Alexander Rush. 2020b. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems*, volume 33, pages 20378–20389. Curran Associates, Inc.

Chinnadhurai Sankar, Sujith Ravi, and Zornitsa Kozareva. 2021. ProFormer: Towards on-device LSH projection based transformers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2823–2828, Online. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commun. ACM*, 63(12):54–63.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2020. Q-BERT: Hessian based ultra low precision quantization of bert. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8815–8821.

Pranaydeep Singh and Els Lefever. 2022. When the student becomes the master: Learning better and smaller monolingual models from mBERT. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4434–4441, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. 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.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. LST: Ladder side-tuning for parameter and memory efficient transfer learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 12991–13005. Curran Associates, Inc.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Marcos Treviso, Ji-Ung Lee, Tianchu Ji, Betty van Aken, Qingqing Cao, Manuel R. Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Colin Raffel, Pedro H. Martins, André F. T. Martins, Jessica Zosa Forde, Peter Milder, Edwin Simpson, Noam Slonim, Jesse Dodge, Emma Strubell, Niranjan Balasubramanian, and 3 others. 2023. Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 11:826–860.

Asahi Ushio, Yi Zhou, and Jose Camacho-Collados. 2023. Efficient multilingual language model compression through vocabulary trimming. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14725–14739, Singapore. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *Preprint*, arXiv:2412.13663.

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

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, and 4 others. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 214–229, New York, NY, USA. Association for Computing Machinery.

Adina Williams, Nikita Nangia, and Samuel Bowman. 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)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Huiyin Xue and Nikolaos Aletras. 2022. HashFormers: Towards vocabulary-independent pre-trained transformers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7862–7874, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. ZeroQuant: Efficient and affordable post-training quantization for large-scale transformers. In *Advances in Neural Information Processing Systems*, volume 35, pages 27168–27183. Curran Associates, Inc.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8BERT: Quantized 8bit bert. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*, pages 36–39.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Sanqiang Zhao, Raghav Gupta, Yang Song, and Denny Zhou. 2021. Extremely small BERT models from mixed-vocabulary training. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2753–2759, Online. Association for Computational Linguistics.

Yi Zhou, Jose Camacho-Collados, and Danushka Bollegala. 2023. A predictive factor analysis of social biases and task-performance in pretrained masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11082–11100, Singapore. Association for Computational Linguistics.

A Language Model Vocabulary Sizes

Table 5 presents the vocabulary sizes ($|\mathcal{V}|$) for the models used in our experiments, as identified by the Hugging Face Hub. We also report the number of embedding parameters (N_{emb}), the number of model parameters (N), and the overall proportion of embedding parameters (N_{emb}/N). These metrics are also presented in Table 6 for each size of DeBERTa, in addition to model hyperparameters.

Model	$\|\mathcal{V}\|$	N_{emb}	N	N_{emb}/N
DistilBERT	28,996	22.3M	65.8M	33.9%
DistilRoBERTa	50,265	38.6M	82.1M	47.0%
DistilmBERT	119,547	91.8M	135.3M	67.8%
BERT	28,996	22.3M	108.3M	20.6%
RoBERTa	50,265	38.6M	124.6M	31.0%
DeBERTaV3	128,100	98.4M	184.4M	53.3%
mBERT	119,547	91.8M	177.9M	51.6%
XLM-RoBERTa	250,002	192.0M	278.0M	69.1%
XLM-V	901,629	692.5M	778.5M	88.9%

Table 5: The vocabulary size and allocation of parameters for each of the models used in our experiments. In all cases, we select the base model size ($d_{\text{model}} = 768$).

Size	l	h	d_{model}	N_{emb}	N	N_{emb}/N
XSmall	12	6	384	49.2M	70.8M	69.4%
Small	6	12	768	98.4M	141.9M	69.3%
Base	12	12	768	98.4M	184.4M	53.3%
Large	24	16	1024	131.2M	435.1M	30.2%

Table 6: The DeBERTaV3 (He et al., 2023) family of models. Columns l, h, and d_{model} show the number of hidden layers, number of attention heads, and hidden embedding size, respectively.

B Language Model Compression

Supplementary to our discussion of related work (Section 2), we additionally discuss the relation to variety of popular LM compression approaches. We emphasize that these methods are orthogonal to our proposed approach.

Knowledge distillation. Knowledge distillation (Hinton et al., 2015) aims to achieve comparable performance by training a smaller model using the predictions from a larger model. This approach has been successfully applied to LMs (Sanh et al., 2020a; Sun et al., 2020). It can also be used to train models with a smaller vocabulary than the original (Zhao et al., 2021; Singh and Lefever, 2022).

Pruning. Neural network pruning (LeCun et al., 1989) seeks to remove redundant weights while preserving performance. Existing approaches focus on pruning the linear and attention weights in LMs (Sanh et al., 2020b; Kurtic et al., 2022; Frantar and Alistarh, 2023). However, pruning the embedding matrix is widely avoided, as it can substantially harm performance (Kurtic et al., 2024).

Quantization. The aim of quantization is to represent neural network weights using lower precision, therefore reducing computational costs. Recent LM quantization efforts generally focus on quantizing the linear layers (Dettmers et al., 2022; Yao et al., 2022; Frantar et al., 2023). The embedding matrix can also be quantized (Zafrir et al., 2019; Bondarenko et al., 2021), although Shen et al. (2020) find that it is more sensitive to quantization.

C Datasets

In all cases, we use the publicly available version of each dataset available from Hugging Face (Lhoest et al., 2021). The GLUE benchmark comprises a diverse range of tasks, including linguistic acceptability (CoLA, Warstadt et al. 2019), sentiment analysis (SST-2, Socher et al. 2013), paraphrasing/sentence similarity (MRPC, Dolan and Brockett 2005; STS-B, Cer et al. 2017; QQP, Iyer et al. 2017), and natural language inference (RTE, Dagan et al. 2006; WNLI, Levesque et al. 2012; QNLI, Rajpurkar et al. 2016; MNLI, Williams et al. 2018). The number of examples per split in each dataset are listed in Table 7. The XNLI dataset (Conneau et al., 2018) extends MNLI to 15 languages: Arabic, Bulgarian, Chinese, English, French, German, Greek, Hindi, Russian, Spanish, Swahili, Thai, Turkish, Vietnamese, and Urdu.

D Implementation & Hardware

We implement our experiments using PyTorch (Paszke et al., 2019), Hugging Face Transformers (Wolf et al., 2020) and Hugging Face Datasets (Lhoest et al., 2021). Since downstream task performance is not relevant to this study, we do not perform hyperparameter tuning. Instead, we broadly follow the hyperparameters from Devlin et al. (2019), listed in Table 8.

We fine-tune all models using a single NVIDIA Tesla V100 (SXM2 32GB) GPU and Intel Xeon Gold 6138 CPU. For consistency, each model type is evaluated on the same physical hardware.

E Fine-tuning on GLUE

Table 10 presents the task performance for each model across the GLUE benchmark. We observe that the performance is largely identical, although there are occasional fluctuations where PEMA performs fractionally better or worse than the baseline. Finally, we note that XLM-RoBERTa and XLM-V both demonstrate very low performance on CoLA, although this issue has also been observed in other studies, e.g. Zhou et al. (2023).

F Runtime Impact

Table 9 presents the mean duration and standard deviation of applying PEMA to RoBERTa and the subsequent fine-tuning process. It also shows the proportion of time spent applying PEMA relative to fine-tuning. We observe that for five of the nine datasets in GLUE, applying PEMA takes less than half a second. For eight out of nine datasets, applying PEMA takes less than 1% of the fine-tuning duration. We note that the time taken to apply PEMA correlates with the size of the fine-tuning dataset (Figure 2). Overall, we note that the time taken to apply PEMA is generally fractional compared to the fine-tuning duration, even though we made no effort to optimize our implementation. As guidance for future optimization efforts, we note that the dataset processing operations in PEMA are trivially parallelizable.

Dataset	Train	Validation	Test	Total
CoLA	8,551	1,043	1,063	10,657
MNLI	392,702	19,647	19,643	431,992
MRPC	3,668	408	1,725	5,801
QNLI	104,743	5,463	5,463	115,669
QQP	363,846	40,430	390,965	795,241
RTE	2,490	277	3,000	5,767
SST-2	67,349	872	1,821	70,042
STS-B	5,749	1,500	1,379	8,628
WNLI	635	71	146	852

Table 7: The number of examples per split in each of the GLUE datasets.

Hyperparameter	GLUE	XNLI
Adam ϵ	1e-8	
Adam β_1	0.9	
Adam β_2	0.999	
Batch Size	32	
Dropout (Attention)	0.1	
Dropout (Hidden)	0.1	
Learning Rate (Peak)	2e-5, 7.5e-6 (XLM)	
Learning Rate Schedule	Linear	
Sequence Length	128	
Training Epochs	3	2

Table 8: The hyperparameters used for each set of experiments.

Dataset	PEMA	Fine-tuning	%
CoLA	$0.4_{0.0}$	$172.7_{0.9}$	0.2
MNLI	$8.8_{0.2}$	$7817.8_{16.6}$	0.1
MRPC	$0.3_{0.0}$	$78.7_{0.7}$	0.4
QNLI	$2.4_{0.0}$	$2092.8_{2.0}$	0.1
QQP	$13.3_{0.5}$	$7235.5_{4.9}$	0.2
RTE	$0.4_{0.0}$	$55.4_{0.6}$	0.7
SST-2	$1.2_{0.0}$	$1329.2_{0.3}$	0.1
STS-B	$0.4_{0.0}$	$118.7_{0.5}$	0.3
WNLI	$0.3_{0.0}$	$18.3_{0.8}$	1.4

Table 9: The mean duration (seconds) and standard deviation over five runs of applying PEMA to RoBERTa and fine-tuning on the GLUE datasets.

Model	PEMA	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI	Mean
DistilBERT	✗	49.3	82.2	84.2	88.5	86.7	59.6	90.5	86.5	49.3	$75.2_{1.5}$
	✓	49.3	82.2	84.2	88.6	86.7	59.6	90.5	86.5	49.3	$75.2_{1.5}$
DistilRoBERTa	✗	56.4	84.2	85.0	90.9	87.2	65.7	92.3	87.2	53.0	$78.0_{0.9}$
	✓	56.4	84.2	85.0	90.9	87.2	65.7	92.3	87.2	53.0	$78.0_{0.9}$
DistilmBERT	✗	29.7	78.3	81.8	86.7	85.8	60.9	89.1	84.4	48.2	$71.6_{0.3}$
	✓	29.6	78.3	81.8	86.7	85.8	60.9	89.2	84.4	48.2	$71.6_{0.4}$
BERT	✗	56.4	84.3	84.3	91.1	87.9	64.4	92.6	88.1	37.7	$76.3_{0.7}$
	✓	56.7	84.3	84.3	91.3	87.8	64.4	92.5	88.1	37.7	$76.3_{0.8}$
RoBERTa	✗	57.6	87.8	88.4	92.8	88.4	71.1	94.2	89.9	52.1	$80.3_{1.2}$
	✓	57.6	87.8	88.4	92.7	88.4	71.1	94.2	89.9	52.1	$80.3_{1.2}$
DeBERTaV3	✗	67.4	90.2	88.5	93.9	89.9	79.8	95.6	90.9	53.0	$83.2_{0.8}$
	✓	67.4	90.2	88.3	93.9	89.9	79.8	95.5	90.9	53.0	$83.2_{0.8}$
mBERT	✗	35.3	82.3	85.8	91.1	87.1	69.0	91.0	88.0	53.0	$75.8_{2.0}$
	✓	35.4	82.2	85.8	91.1	87.2	69.0	90.8	88.0	53.0	$75.8_{2.0}$
XLM-RoBERTa	✗	22.6	83.9	76.9	89.5	86.9	57.3	92.2	84.2	52.1	$71.7_{2.0}$
	✓	22.4	84.0	76.8	89.5	86.8	57.3	92.0	84.2	52.1	$71.7_{2.0}$
XLM-V	✗	0.0	84.5	68.8	89.6	86.7	54.1	91.8	80.8	55.2	$68.0_{0.6}$
	✓	0.0	84.5	68.8	89.6	86.7	54.1	91.6	80.8	55.2	$67.9_{0.6}$

Table 10: Results on the validation set for each task from GLUE. We present the mean performance over five different seeds, accompanied by the overall mean and standard deviation. We report Matthews correlation for CoLA, F1 for QQP, Spearman correlation for STS-B, and accuracy for the remaining tasks.