Artificial Intelligence In Medicine 157 (2024) 103002

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

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

Research paper

Efficiency at scale: Investigating the performance of diminutive language models in clinical tasks

Niall Taylor [a,*], Upamanyu Ghose [a,b,c], Omid Rohanian [d,f], Mohammadmahdi Nouriborji [e,f], Andrey Kormilitzin [a], David A. Clifton [d,g], Alejo Nevado-Holgado [a]

[a] Department of Psychiatry, University of Oxford, Oxford, United Kingdom
[b] Centre for Artificial Intelligence in Precision Medicines, University of Oxford, United Kingdom
[c] King Abdulaziz University, Saudi Arabia
[d] Department of Engineering Science, University of Oxford, Oxford, United Kingdom
[e] Sharif University of Technology, Tehran, Iran
[f] NLPie Research, Oxford, United Kingdom
[g] Oxford-Suzhou Centre for Advanced Research, Suzhou, China

ARTICLE INFO

Keywords:
Large language models
Artificial intelligence
PEFT
Fine-tuning
NLP

ABSTRACT

The entry of large language models (LLMs) into research and commercial spaces has led to a trend of ever-larger models, with initial promises of generalisability. This was followed by a widespread desire to downsize and create specialised models without the need for complete fine-tuning, using Parameter Efficient Fine-tuning (PEFT) methods. We present an investigation into the suitability of different PEFT methods to clinical decision-making tasks, across a range of model sizes, including extremely small models with as few as 25 million parameters.

Our analysis shows that the performance of most PEFT approaches varies significantly from one task to another, with the exception of LoRA, which maintains relatively high performance across all model sizes and tasks, typically approaching or matching full fine-tuned performance. The effectiveness of PEFT methods in the clinical domain is evident, particularly for specialised models which can operate on low-cost, in-house computing infrastructure. The advantages of these models, in terms of speed and reduced training costs, dramatically outweighs any performance gain from large foundation LLMs. Furthermore, we highlight how domain-specific pre-training interacts with PEFT methods and model size, finding the domain pre-training to be particularly important in smaller models and discuss how these factors interplay to provide the best efficiency-performance trade-off. Full code available at: https://github.com/nlpie-research/efficient-ml.

1. Introduction

The Natural Language Processing (NLP) research space is now dominated by large language models, with a steady influx of different so-called foundation models from major AI companies every few months. The vast majority of recent LLMs are designed for *generative* tasks and chat-style interactions, reliant on very large models with several billion (even well over 100 billion) model parameters trained using a mixture of autoregressive LM pre-training with follow-up reinforcement learning from human feedback (RLHF) to create the likes of ChatGPT [1], Llama-2 [2], Claude 2 [3], or Mixtral 8X7B [4]. However, the performance of these generative LLMs on classic NLP tasks such as sequence classification, relation extraction, named entity recognition, and embedding similarity search, especially in the clinical domain remains lacklustre [5–10] and typically requires further training or adaptation techniques. In many such cases, much smaller, BERT-style LLMs trained with masked language modelling (BERT [11], and RoBERTa [12]) can be easily fine-tuned to be competitive, or even surpass the performance of their larger counterparts [10,13], whilst only having approximately 100 million model parameters.

1.1. Scales of LLM

Recent LLM research has predominantly focused on exceptionally large models from the more prolific AI companies, including ChatGPT

* Corresponding author.
E-mail address: niall.taylor@st-hughs.ox.ac.uk (N. Taylor).

[1] Recently LLMs with much fewer model parameters are referred to as SLMs, but LLMs can be used interchangeably. When discussing Language Models generally, we use the term LLM.

https://doi.org/10.1016/j.artmed.2024.103002
Received 28 March 2024; Received in revised form 24 September 2024; Accepted 16 October 2024
Available online 23 October 2024

from OpenAI [1] and Llama [2] from Meta. Although recent models from OpenAI are proprietary, it is widely recognised that the size of foundation models spans a broad range, from approximately 3 to 175 billion parameters, and with GPT-4 potentially more than one trillion parameters. In contrast, there exist smaller LMs (SLMs)[1] such as BERT [11], which contains approximately 108 million parameters. The relative cost, simplicity, and re-usability of these variously scaled models are crucial aspects to consider, and we aim to provide a holistic analysis of the interplay between different efficiency metrics and model size.

1.2. Fine-tuning and PEFT

Even SLMs are relatively compute-intensive when compared to simpler machine learning alternatives, such as TF-IDF or Bag-of-Words paired with random forest classifiers. Moreover, adapting numerous LLMs to new tasks can become unfeasible in low-resource settings where GPUs are scarce or non-existent. Common approaches to reduce model size include: knowledge distillation [14,15], architecture compression [16], and pruning [17]. These approaches generally aim to maintain a high level of performance in compressed models by harnessing the knowledge from the much larger *teacher* LLMs. Although these approaches have had great success in producing smaller LLMs, adapting to new tasks still requires full fine-tuning of all model parameters to achieve optimal performance on specific downstream tasks. This may necessitate a plethora of domain- or task-specific LLMs, which cannot be used interchangeably due to catastrophic forgetting [18]. A more prevalent approach today is to adapt the fine-tuning approach itself. Traditional approaches to adapting LLMs to downstream tasks involve the introduction of task-specific neural network layers (often referred to as heads) to provide the extra flexibility required to complete a task, such as sequence classification. This training occurs in a supervised manner, involving updates to all model parameters, including task-specific parameters (full fine-tuning). Full fine-tuning of smaller LLMs, such as BERT-base [11] with merely 108 million parameters, has been feasible with modern GPUs, requiring only a single GPU with full precision. However, with the advent of models like Llama-2 [2] with 65 billion parameters, the practicality of fine-tuning these models on low-end hardware dwindles.

Several strategies exist to address this issue, one of which is being the reduction of model size in terms of floating-point precision, bits, and the physical memory needed to store the weights through quantisation. This enables full fine-tuning of moderately sized models. [19]. Pruning model parameters to reduce the *redundant* weights for given downstream tasks has also been effective in certain cases [17]. Another approach is to avoid full fine-tuning altogether, opting instead for zero-shot task adaption through prompting (prompt engineering), or by reducing the number of trainable parameters necessary for fine-tuning the LLM for its new task, a process known as Parameter Efficient Fine-tuning. Notable PEFT methods include: Prompt tuning [20], Prefix tuning [21], Low Rank Adaptation (LoRA) [22], and Inhibit Activations (IA^3) [23]. These PEFT methods have become popular across various NLP tasks, and in this work, we will explore the utility of a select few for differently sized LLMs in the clinical domain.

1.3. Clinical domain - LLM adaptation

Unstructured clinical notes form a large portion of electronic health records (EHRs) and can offer a substantial amount of clinically salient information given appropriate representation, such as that given by a LLM. Foundation LLMs are typically developed and trained for a broad-stroke, general-purpose set of applications: trained on open, web-based text data and intended to be applied to *similar* open, web-based text data. When taking foundation LLMs and applying them to biomedical and clinical texts, performance often drops significantly [5–9,13,24–27]. Achieving state-of-the-art (SoTA) performance in the clinical domain still involves training generic LLMs on biomedical or clinical domain data, and PEFT methods can provide efficient ways to adapt open LLMs to the clinical domain. The clinical domain is also inherently a compute-limited environment, with sensitive data that typically cannot be sent to third-party APIs. Thus, small, efficient LLMs that can perform specific tasks well and potentially run on edge devices are highly sought after [27,28].

1.4. Related work and motivation

Utilising smaller LLMs as an efficient alternative to their larger counterparts has had increasing attention, with recent releases such as Phi-2 (2.7 billion parameters) from Microsoft [29,30] achieving similar performance on certain benchmark suites as 70 billion parameter model alternatives. Whilst Phi-2 is relatively small, they typically still require high-end GPUs to allow any further training for specific tasks, and deploying to production in any real-time setting becomes non-trivial in terms of cost and time, even with the use of quantisation and PEFT methods. The applicability of LLMs for the clinical domain is well researched, with a great deal of attention on creating generative LLMs that excel in question-answering style tasks [31,32], but there has been little research into the utility of PEFT methods in this space, nor the comparative effectiveness for traditional NLP tasks. Recent efforts have extensively explored the use of PEFT methods for large-scale models, aiming to align them with new domains or tasks [19,22,33], but few have extended this to SLMs or the clinical domain [34]. One group has recently investigated PEFT for traditional clinical NLP tasks with Llama models, and our work follows a very similar path [35]. The major distinction in our work is the emphasis on the efficiency of these methods and their applicability to much smaller LLMs and how this translates to time and cost demands.

Our key contributions involve a comparison of recent Parameter Efficient Fine-Tuning (PEFT) methods for their applicability to clinical decision tasks. We explore the suitability of these PEFT methods for small LLMs such as Mobile and TinyBert architectures, which have significantly fewer parameters compared to their larger counterparts. Additionally, we investigate the effectiveness of PEFT methods when applied to knowledge distilled LLMs like DistilBERT. Furthermore, we delve into the interplay between the pre-training domain of the LLMs, the sample size of the clinical data, and the performance of various PEFT methods, providing insights into the optimal combinations for efficient adaptation of LLMs to the clinical domain. Finally, we provide a comparison of efficiency when adapting differently sized LLMs to provide insights into associated time and financial requirements.

2. Methods

2.1. Model architectures

We evaluate the performance of PEFT across various transformer-based LLMs architectures of differing sizes, including: TinyBERT [36], MobileBERT [16], DistilBERT [15] and standard BERT [11] which are our SLMs. For a further set of experiments we also include the much larger LLM Llama-2-7b [2]. A table of relevant architecture details is provided in Table 1.

2.2. Domain pre-training

In addition to exploring various transformer-based LLM architectures of different sizes, we examine three domain variants for each from previous research where model checkpoints have been released and are available to download via HuggingFace [37]:

- **General:** Original, unadapted models [11,15,16,36].
- **Biomedical:** Models pre-trained or distilled with biomedical literature [38]

Table 1
Model architectures and their associated number of parameters, Video Random Access Memory (VRAM), and Floating Point Operations (FLOPs). FLOPs were based on a random sample of 10 tokens.

Model architecture	# Params (mil)	GPU (VRAM GB)	FLOPs
Tiny-BERT	13.87	0.052	3.66×10^7
Mobile-BERT	24.58	0.092	1.62×10^8
Distil-BERT	65.78	0.245	3.41×10^8
BERT	108.31	0.403	6.81×10^8
Llama2-7b	6607.34	24.6	5.18×10^{10}
Llama2-7b (bfloat16)	6607.34	12.37	5.18×10^{10}

- **Clinical:** Models pre-trained with clinical EHR data [28]

Using domain trained LLMs allows us to investigate the interplay between domain pre-training, model size, and the chosen PEFT methods.

2.3. Downstream fine-tuning

We opt to compare performance using a traditional fine-tuning setup, whereby each LLM is adapted with a task-specific head to perform the respective downstream task. For each task, we will utilise additional linear layer(s) on top of the base LLM (classification head), with a task-specific loss that is used to update all model parameters (the base LLM and the additional task head). This approach remains the most suitable across all model architectures and aligns with previous research [28,34].

2.4. PEFT

Parameter Efficient Fine-tuning methods are numerous, but they typically fall into two categories: introducing new trainable parameters or selectively freezing existing ones. Based on previous works [33,35] and some preliminary experiments, we opt to only focus on LoRA and IA^3 for our main experiments, which generally demonstrate significantly better performance compared to alternative PEFT methods (prefix and prompt tuning). Moreover, aligning prefix tuning and prompt tuning with NER tasks is not straightforward and we believed it offered limited value to adapt these methods for NER specifically. In addition to the trainable parameters specific to each PEFT method described below, the task-specific heads 2.3 are also updated during training.

Low-rank adaptation of large language models. Low-Rank Adaptation of LLMs or LoRA [22] is a reparameterisation technique that works by injecting two trainable matrices (A and B) that act as an approximation of a singular value decomposition (SVD) of the weight update ΔW for any weight matrix $W \in \mathbb{R}^{d \times k}$ in the LLM. The approximation works as $\Delta W = BA$, where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$ and $r \ll min(d, k)$ is the rank of the LoRA matrices, which is a tunable parameter. The new forward pass is updated from $h = Wx$ (where x is the input embedding to the layer/operation and h the output embedding) to $h = (W + \Delta W)x = (W + AB)x = Wx + ABx$. While it is possible to introduce the LoRA matrices in any layer of the LLM, it is common practice to introduce them as weight update approximations for the key, query and value matrices. The underlying assumption is that the weight updates in LLMs intrinsically have a lower rank than their dimensions, and thus can be well approximated by their SVD. Additionally, once fully trained, the LoRA matrices can be integrated into the model as $W_{updated} = W_0 + BA$, thereby introducing no inference latency. With LoRA the original weight matrices of the LLM remain frozen during the fine-tuning phase.

IA^3. Infused Adapter by Inhibiting and Amplifying Inner Activation (IA^3) shares similarities with other adapter methods that introduce new parameters to scale activations using learned vectors [23]. While these learnable vectors can be applied to any set of activations, applying them to the keys and values in the relevant attention mechanism and the intermediate activation of the position-wise feed-forward networks was found to be both efficient and sufficient. For a transformer based architecture, we have a query $Q \in \mathbb{R}^{d_q}$, key $K \in \mathbb{R}^{d_k}$, value $V \in \mathbb{R}^{d_v}$, and a position-wise feed-forward network with hidden dimension d_{ff}. IA^3 introduces learnable vectors $l_k \in \mathbb{R}^{d_k}$, $l_v \in \mathbb{R}^{d_v}$ and $l_{ff} \in \mathbb{R}^{d_{ff}}$ and modifies the attention and feed-forward calculation as follows:

$$softmax\left(\frac{Q(l_k \odot K)}{\sqrt{d_k}}\right)(l_v \odot V) \tag{1}$$

$$(l_{ff} \odot \gamma(W_1 x))W_2 \tag{2}$$

where $\odot$ represents the element-wise product, and γ, W_1 and W_2 are the activation function and weight matrices of the feed-forward network. W_1 is a matrix of dimension $d_{ff} \times d_v$, and W_2 is of dimension $d_v \times d_{ff}$. The equations use Numpy's 'broadcasting notation'[39] where the (i, j)th entry of l⊙x is $l_j.x_{i,j}$. Similar to LoRA, the learnable vectors can be merged into the model as $l \odot W$ because any operation $l \odot Wx$ is equivalent to $(l \odot W)x$. Hence, this method does not introduce any inference latency either. Once again, with IA^3 the original weight matrices of the LLM remain frozen during fine-tuning.

2.5. Few-shot training

A prevalent challenge in real-world scenarios is the scarcity of training samples, especially in the clinical domain where certain diseases are inherently rare and generating gold-standard annotations demands clinical expertise and considerable time, both of which are limited resources. Therefore, the ability to train a viable model with few training samples is another angle of efficiency we explore. This is achieved by supplying only a limited number of training samples per class to a specific model. We carry out a series of experiments with an escalating number of samples per class to determine the effect of different model sizes and PEFT methods.

2.6. Datasets and tasks

We utilise a number of commonly used clinical datasets for downstream evaluation, focusing on the following tasks: named entity recognition (NER), sequence classification and relation extraction (RE), in line with earlier clinical NLP research [40,41], and see Table 2.

2.6.1. Sequence classification tasks

Sequence classification tasks involve predicting a class label of a given sequence of text, such as a clinical note. Here the entire text is processed by the LLM and the produced representation of that text is passed as features to the task-specific head as outlined above2.3.

MIMIC-III ICD-9 triage. A common task with the MIMIC-III dataset [42] involves classifying patient records according to their medical diagnoses, which are coded using a system known as ICD-9. We utilise a simplified version of this task, where the top 20 most commonly occurring ICD-9 codes are categorised into seven *triage* groups: [*Cardiology, Obstetrics, Respiratory, Neurology, Oncology, AcuteMedicine, Gastroenterology*]. This grouping was developed in collaboration with clinicians. For further information, please refer to the original paper [34].

MIMIC-III - clinical outcomes. Two clinical outcome tasks associated with the MIMIC-III dataset [42] are Mortality Prediction (MP) and Length of Stay (LoS) prediction [43]. MP involves analysing discharge summaries from the ICU to assess a patient's mortality risk, constituting a binary classification problem. The LoS task also uses ICU discharge summaries to forecast the duration of a patient's hospital stay, with duration's binned into four classes: under 3 days, 3 to 7 days, 1 week to 2 weeks, and more than 2 weeks.

Table 2
Dataset details.

Dataset	Task type	# classes	# train samples	# eval samples
MIMIC-III MP	Seq. CLS	2	33,954	9,822
MIMIC-III LoS	Seq. CLS	3	30,421	8,797
MIMIC-III ICD-9 Triage	Seq. CLS	7	9,559	3,172
I2B2 2010 RE	Seq. CLS	9	22,256	43,000
I2B2 2010	NER	7	6726	27,626
I2B2 2012	NER	13	6797	5,664
I2B2 2014	NER	42	45 974	32,586

I2B2 2010 relation extraction. We used several curated datasets from the I2B2 series, including the 2010 medical relation extraction dataset [44] which aims to classify text based on the apparent medical relationship being described, with the following derived labels:

1. Treatment improves medical problem (TrIP)
2. Treatment worsens medical problem (TrWP)
3. Treatment causes medical problem (TrCP)
4. Treatment is administered for medical problem (TrAP)
5. Treatment is not administered because of medical problem (TrNAP)
6. Test reveals medical problem (TeRP)
7. Test conducted to investigate medical problem (TeCP)
8. Medical problem indicates medical problem (PIP)
9. No Relations

We follow the same pre-processing procedure outlined in previous works [28].

2.6.2. Named entity recognition

Named Entity Recognition (NER) is the task of locating and identifying named entities such as persons, locations, organisations, etc. within unstructured text. NER involves labelling words in a text that refer to types such as person, organisation, place, date, etc. Example: Identifying "Warfarin" as a drug and "DVT" as the condition in the sentence "Patient was started on warfarin therapy due to left lower extremity DVT". NER is often formed as a token classification task, whereby each token in the sequence is labelled as the different possible entities.

I2B2 - 2010 and 2012. These two NER tasks involve classifying text spans related to temporal relations [44,45] within discharge summaries, as delineated by expert annotations. The classification is based on four primary categories: clinical concepts, clinical departments, evidentials, and occurrences. These categories are further broken down into more specific entities: *medical problem (PR)*, *medical treatment (TR)*, *medical test (TE)*, *clinical department (CD)*, *evidential (EV)*, *occurrence (OC)*, and *none (NO)*.

I2B2 - 2014. A deidentification task, whereby spans of text within clinical notes are classified using different protected health information (PHI) such as name, address, and postcode [46].

For further dataset and task details, see Appendix A and for hardware and implementation details see Appendix C.

3. Results

3.1. Model size vs PEFT

The number of trainable parameters is an important factor in determining the efficiency in model performance and has a strong correlation with cost and time of training. We detail the performance metrics for various PEFT methods applied to each model type across different clinical tasks. In Table 3, we present the results for sequence classification and NER across different PEFT methods and model sizes.

The results demonstrate that LoRA consistently outperforms other PEFT methods across all models and tasks, often approaching the performance of full fine-tuning.

We also compare the number of trainable parameters as a function of the different PEFT methods in Fig. 1. There is a clear correlation between the number of trainable parameters and performance, and LoRA appears to provide larger models an advantage over fully fine-tuned smaller models. The performance disparity between full fine-tuning and LoRA becomes more pronounced with smaller models.

3.2. Differential effect of LoRA rank according to model size

Given the superior performance of LoRA over other PEFT methods, as evidenced in Fig. 1, we aimed to methodically evaluate the impact of the LoRA rank hyperparameter across models of varying sizes. For this purpose, we employed the Optuna package [47] to conduct 20 trials of hyperparameter optimisation, holding the LoRA rank constant at $r \in 8, 16, 32, 64, 128$. The hyperparameters adjusted during tuning included LoRA dropout ($d \in 0.1, 0.3, 0.5$), LoRA alpha ($\alpha \in 0.3, 0.5, 1.0$), and learning rate ($lr \in [10^{-5}, 10^{-3}]$). The Llama model was excluded from this experiment due to its significantly larger size compared to BERT-based models, which would have imposed an excessive computational load for hyperparameter tuning. Following the hyperparameter search, we selected the optimal performing model for each r value to analyse its effect on models with differing parameter counts (Fig. B.4).

Increasing the rank r in TinyBioBERT led to improved performance up to $r = 64$, after which a slight decline was observed at $r = 128$. A similar pattern was noted in BioDistilBERT, with the turning point at $r = 32$. The impact of rank on BioMobileBERT was more variable, with a noticeable performance dip only at $r = 64$. This variability might be attributed to the distinct architecture of BioMobileBERT compared to other BERT-based models [16]. For BioBERT, the larger model in the BERT family, there was a modest improvement at $r = 16$, but performance tended to decrease at higher ranks. Conversely, for the RoBERTa model, performance enhancements were seen at ranks $r = 32$ and $r = 128$, yet no clear pattern between rank and performance emerged. Despite these fluctuations, the overall impact on model performance was relatively minor, with the greatest increase in AUROC being 0.0125 and the largest decrease being 0.0078. Hence, even for models with varying number of parameters, the default LoRA rank of 8 is a good trade-off between computational time taken to tune the models and performance. However, if the task at hand would practically benefit from a small increase in the performance metric, tuning the LoRA parameters may be beneficial.

3.3. General vs biomedical vs clinical domain pre-training

Another aspect of efficiency with regards to LLM downstream adaptation is the domain in which the model was pre-trained. We have conducted direct comparisons between models pre-trained in general, biomedical, and clinical domains across our various model architectures. For the sake of brevity, we focus solely on the i2b2-2010 relation extraction task. The performance differences are greatest in the smaller models, with clinically pre-trained models generally performing best with a 1–4 percent improvement based on model size. For results across all tasks and their dependence on domain pre-training, please see Appendix C.6 (see Fig. 2).

Table 3
PEFT results for all downstream tasks using biomedical models, with values representing the median from 3 distinct training runs under varied random seeds for PyTorch weight initialisations. Standard Deviation (SD) is provided in brackets. Micro-averaged F1 scores are reported for the i2b2-2010-RE and all NER tasks. Macro-averaged Receiver Operating Characteristic area under the curve ($ROC\,AUC$) is used for MIMIC-LoS and -MP tasks, while macro-averaged F1 scores are reported for the ICD-9 triage task. Bold results indicate the best PEFT performance and values underlined are top performance across all fine-tuning methods.

(a) Sequence classification task results

Model name	PEFT	ICD9-Triage (F1-macro)	i2b2-2010-RE (F1-micro)	MIMIC-LoS (ROC AUC)	Mimic-MP (ROC AUC)
BioBERT	Full	0.864 (0.002)	0.935 (0.004)	0.709 (0.002)	0.819 (0.020)
	IA3	0.703 (0.19)	0.896 (0.004)	0.634 (0.001)	0.769 (0.005)
	LoRA	**0.827** (0.002)	**0.925** (0.001)	**0.697** (0.002)	0.828 (0.002)
BioDistilBERT	Full	0.862 (0.010)	0.927 (0.003)	0.706 (0.003)	0.825 (0.006)
	IA3	0.792 (0.008)	0.906 (0.002)	0.677 (0)	0.797 (0.001)
	LoRA	**0.855** (0.005)	0.928 (0.003)	**0.702** (0.001)	**0.825** (0.001)
BioMobileBERT	Full	0.851 (0.004)	0.932 (0.003)	0.704 (0.004)	0.819 (0.011)
	IA3	0.744 (0.012)	0.897 (0.003)	0.639 (0.001)	0.774 (0.002)
	LoRA	**0.808** (0.004)	**0.918** (0.002)	**0.671** (0.004)	**0.798** (0.002)
TinyBioBERT	Full	0.727 (0.012)	0.910 (0.005)	0.684 (0.001)	0.802 (0.001)
	IA3	0.390 (0.035)	0.852 (0.002)	0.588 (0.003)	0.607 (0.003)
	LoRA	**0.599** (0.008)	**0.895** (0.003)	**0.649** (0.006)	**0.764** (0.003)

(b) NER task results

Model name	PEFT	i2b2-2010-NER (F1-micro)	i2b2-2012-NER (F1-micro)	i2b2-2014-NER (F1-micro)
BioBERT	Full	0.819 (0.003)	0.824 (0.001)	0.967 (0.001)
	IA3	0.473 (0.002)	0.485 (0.006)	0.850 (0.001)
	LoRA	**0.696** (0.003)	**0.753** (0.001)	**0.935** (0)
BioDistilBERT	Full	0.803 (0.003)	0.795 (0.006)	0.967 (0.001)
	IA3	0.498 (0.003)	0.503 (0.001)	0.883 (0)
	LoRA	**0.718** (0.008)	**0.729** (0.006)	**0.940** (0.001)
BioMobileBERT	Full	0.796 (0.003)	0.772 (0.006)	0.966 (0)
	IA3	0.515 (0.003)	0.515 (0.003)	0.908 (0)
	LoRA	**0.638** (0.010)	**0.650** (0.004)	**0.941** (0.001)
TinyBioBERT	Full	0.655 (0.004)	0.705 (0.008)	0.906 (0.003)
	IA3	0.328 (0.009)	0.381 (0.003)	0.715 (0.002)
	LoRA	**0.438** (0.007)	**0.561** (0.009)	**0.8051** (0.013)

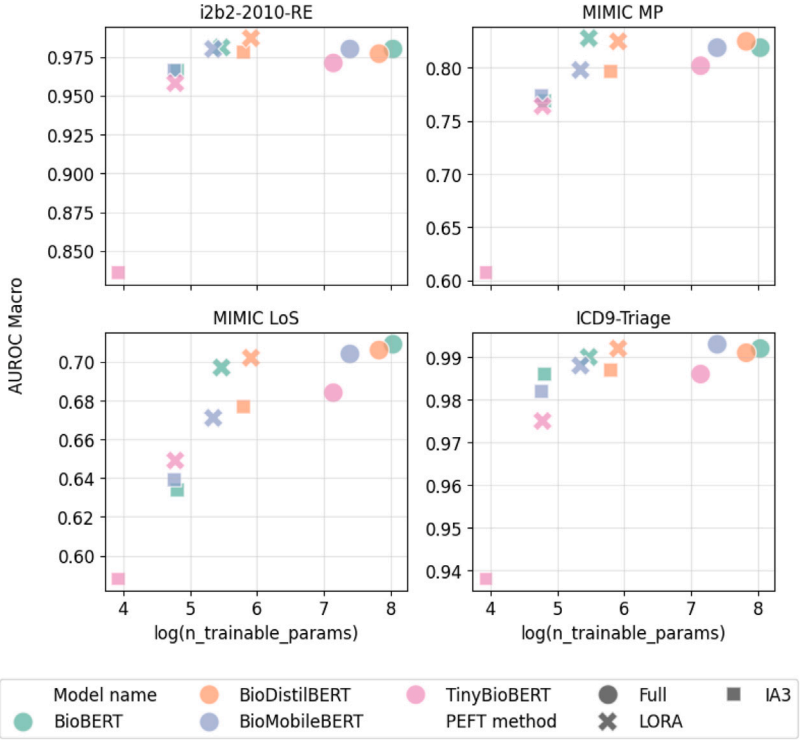

Fig. 1. Sequence classification performance across the different LLM model sizes and the associated number of trainable parameters.

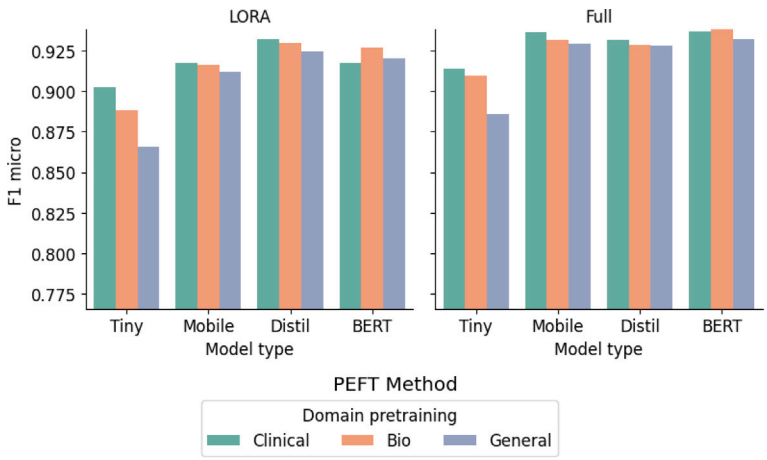

Fig. 2. Comparison of F1 micro scores on the I2B2 2010 relation extraction task dependent on whether the model received biomedical, clinical, or general domain pre-training. Model type refers to the base model(architecture) used: TinyBERT, MobileBERT, DistilBERT or standard BERT as described in Table 1.

3.4. Budget

The primary advantage of employing PEFT methods lies in their ability to reduce training times, lower GPU memory demands, minimise storage requirements and enhance model reusability (all of which lower financial burden). In our study, we examined the trade-offs among these aspects for various model architectures, focusing on the most effective PEFT method identified in our experiments, namely, LoRA. For each defined budget, we used MIMIC mortality prediction as the benchmark task and macro-averaged AUROC as the metric of evaluation. In addition to training the LoRA versions of each model, we also conducted full fine-tuning on each model to determine whether any budget level could achieve efficiency improvements comparable to those provided by PEFT approaches. The only exception was the Llama model, which was exclusively trained with LoRA due to computational constraints.

3.4.1. Time

A key measure of efficiency is the training time and the speed at which different models converge within a constrained period, particularly a relatively short one. We set an initial time limit of 2000 s (33 minutes) for all models. To evaluate the performance of the models that seemed to show an increasing trend in performance after the budget of 2000 s (Fig. 3), we raised the budget to 6000 s (100 minutes). An exception was made for the Llama model, which remained under-trained even after 6000 s, necessitating an extension of the training period to approximately 21,500 s (6 hours) to attain optimal performance.

We observed that the fully fine-tuned version of the models, regardless of size, was quicker to converge than the LoRA versions, followed by eventually overfitting. The LoRA versions of the models eventually converged to the performance (or close to the performance) of the fully fine-tuned models. This observation suggests that fully fine-tuning a model on a small time budget could theoretically obtain an efficiency gain similar to the PEFT methods. However, from a practical standpoint, the LoRA version of all models converged to similar performance within ~1 h of training (Fig. 3) while being more memory efficient. A caveat to this analysis is that the learning rates for LoRA and full fine-tuning were different due to drastic fluctuations in performance, whereby one approach would under- or over-fit massively. A more detailed analysis of the difference in efficiency between the methods is discussed in Section 3.4.3 It is also important to acknowledge that larger models, such as Llama, deliver superior performance but incur significantly higher time and memory costs.

3.4.2. Few-shot training

Another focus for efficient training involves restricting the number of training samples, reflecting real-world situations with especially rare outcomes or cases where producing labels is challenging. We explored sample budgets that ranged from 8 to 4096 samples, increasing incrementally by a factor of 2.

As expected, we observed a direct relationship between sample budget and model performance, regardless of the model type and training method used. While we noticed the fully fine-tuned models generally performing better than their LoRA counterparts for smaller sample budgets, the difference became negligible for higher budget values (Fig. 3). The fully fine-tuned models on a budget of 4096 samples under performed when compared against the LoRA versions on all samples. Hence, for sample budget to be considered as an effective method for efficiency gain, we would need more than 4096 samples.

3.4.3. Memory and cost

The GPU and storage requirements for training differ massively between model types, and fine-tuning method. Whilst performance has generally increased with model size, there is a trade-off between performance and compute required, as well as speed of training and inference. We provided the model size and memory requirements in Table 1 and we extend this analysis by calculating the estimated costs of training and storage of the differently sized models in Table 4. As observed in previous results, larger models like Llama-2-7b achieve higher performance on most tasks but at 20 and 94 times the monetary value of models like BioBERT and TinyBioBERT, respectively. If the objective is to fine-tune a model for multiple tasks, BioBERT and similar models can be a good trade-off between monetary cost and performance.

4. Discussion

4.1. PEFT with small LLMs

We have explored the use of different-sized LLMs for various clinical downstream tasks, assessing both traditional fine-tuning and different PEFT methods. From the methods we studied (IA^3 and *LoRA*), we found LoRA to be superior across all tasks, leading us to select it as the preferred PEFT method for all subsequent analyses. Whilst full fine-tuning generally outperforms LoRA, in certain models and tasks the performance is at least matched or even surpassed. Although LoRA works well for all model sizes, the relative performance gap between full fine-tuning and LoRA appears to increase with the smaller models, which was only partially mitigated by increasing the LoRA rank. In fact, it is potentially more resource-effective to use LoRA with a medium or

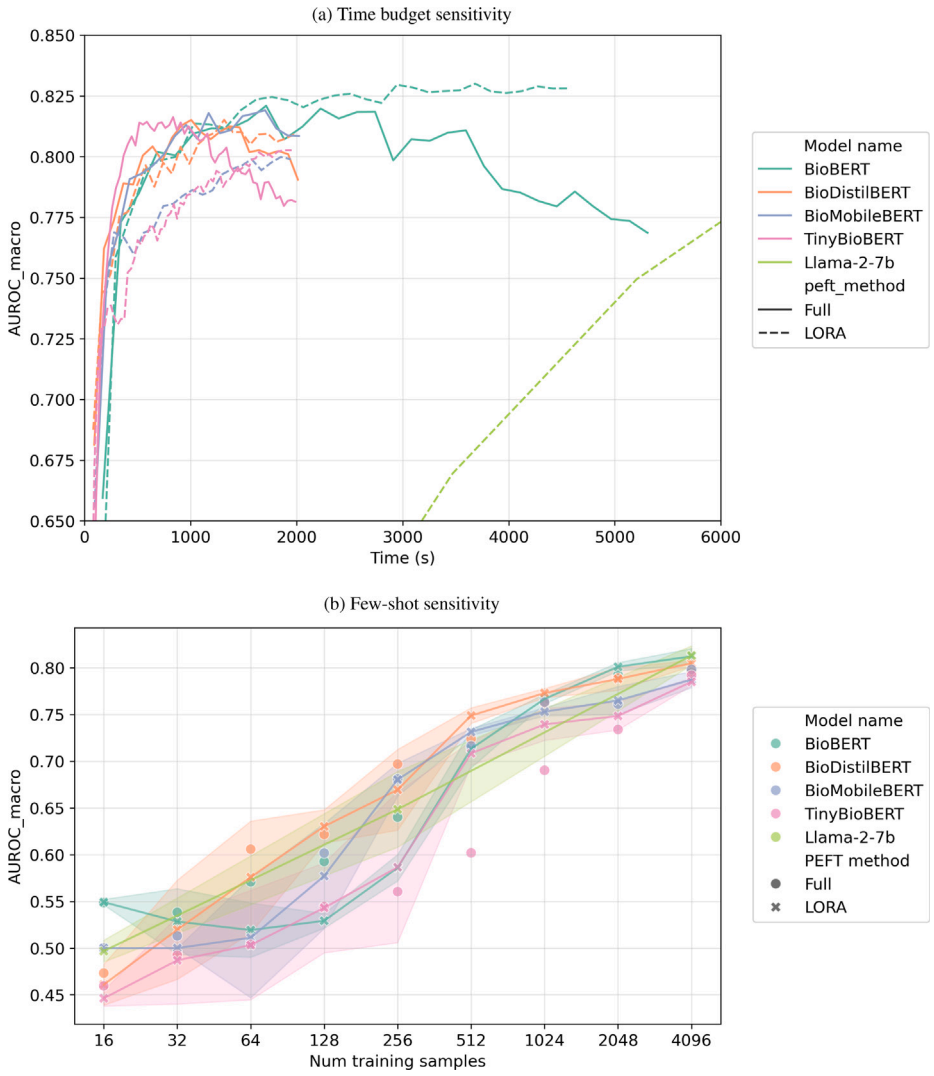

Fig. 3. Effect of training time (**a**) and few-shot sampling (**b**) on models of varying sizes, trained using full fine-tuning as well as LoRA. The connected points reflect the LoRA results to highlight the trend. The task used for this experiment was MIMIC mortality prediction and the highlighted regions shows the standard deviation across 3 runs with different random seeds.

Table 4
Costs for training each model on a task with approximately 30,000 training samples for 10 epochs, followed by running it in inference mode for 100,000 samples. The costs were estimated using AWS EC2 rates. The instances used for estimating training and inference costs were g5.16xlarge and g4dn.16xlarge, respectively.

Model name	PEFT Method	Train time (h)	Inference time (h)	Total cost (GBP)
Llama-2-7b	LoRA	51.07	4.06	112.22
BioBERT	Full	2.51	0.22	5.56
BioBERT	LoRA	2.16	0.22	4.84
BioMobileBERT	Full	1.57	0.14	3.48
BioMobileBERT	LoRA	1.35	0.14	3.03
BioDistilBERT	Full	1.35	0.12	2.99
BioDistilBERT	LoRA	1.21	0.13	2.73
TinyBioBERT	Full	0.53	0.06	1.20
TinyBioBERT	LoRA	0.46	0.06	1.06

large LLM in place of fully fine-tuning the smallest LLMs. This finding highlights the potential of utilising PEFT methods with very small LLMs.

4.2. Comparison of LLM size

The performance of various model sizes was evaluated on a specific task within a fixed time frame, including the 7 billion parameter Llama-2 model. This comparison revealed significant differences in the learning capabilities of models of varying sizes. Numerous smaller LLMs completed 5 epochs of training well before the Llama-2 model achieved comparable performance levels. Nevertheless, when given sufficient time, Llama-2 did reach the highest evaluation performance by a few percentage points in the target task. Llama-2 model is approximately 500 times the size of the TinyBERT models, indicating that the computational demand, even with the implementation of LoRA for Llama-2,

is significantly higher. The duration required for the Llama-2 model to achieve comparable performance on downstream tasks, using the same GPU, was considerable. It took roughly ten times longer to match the performance of smaller LLMs and exceeded six hours of training to attain its peak performance.

4.3. Domain pre-training

The pre-training of LLMs was helpful on average in obtaining a performance gain on the various clinical domain tasks. The advantage of pre-training was more pronounced in tasks such as i2b2-2010-NER and i2b2-2012-NER, where the increase in F1-micro is 2%–4% on average. In contrast, for tasks such as i2b2-2010-RE, Mimic-MP and MIMIC-LOS, the performance gain was just about 1% (Appendix C.6). We do note that the *clinical* LLMs, such as ClinicalBioBERT have been trained on MIMIC-III notes themselves and this does give them an unfair advantage. In line with previous works [25], it could be argued that developing specialised clinical LLMs through pre-training on relevant clinical language remains optimal for subsequent downstream task adaptation. Nevertheless, the trade-off between the time and resources taken to pre-train the models, and the magnitude of performance gain is not consistent across models and tasks.

4.4. Limitations and future work

The selection of PEFT methods investigated in this study reflected the state of the field at the time; however, we acknowledge that this is an evolving research area, and we cannot be certain that other methods would not have outperformed those presented here. Indeed, since conducting these experiments, the PEFT library [48] has introduced several new methods worth exploring.

When comparing various model sizes, we chose to limit training to a single GPU. This approach might disadvantage larger models, particularly the Llama-2 model, which was forced to employ reduction in bit-precision to allow any training. Furthermore, this constraint hindered our ability to thoroughly investigate Llama-2 across all tasks and conduct any hyperparameter optimisation. Future work could seek to explore this further, although the resources required are extensive and arguably yield diminishing returns.

4.5. Conclusion

Overall, we believe this work highlights the power of PEFT methods for small LLMs and demonstrates how domain pre-training can be leveraged to create efficient clinical models. While the capabilities of much larger LLMs are evident, they come with significantly higher time and financial demands.

CRediT authorship contribution statement

Niall Taylor: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Upamanyu Ghose:** Writing – original draft, Visualization, Software, Methodology, Formal analysis, Conceptualization. **Omid Rohanian:** Writing – review & editing, Data curation, Conceptualization. **Mohammadmahdi Nouriborji:** Data curation. **Andrey Kormilitzin:** Writing – review & editing, Supervision. **David A. Clifton:** Writing – review & editing. **Alejo Nevado-Holgado:** Writing – review & editing, Supervision, Funding acquisition.

Funding

NT was supported by the EPSRC Center for Doctoral Training in Health Data Science (EP/S02428X/1). UG was supported by Alzheimer's Research UK, and the Centre for Artificial Intelligence in Precision Medicines (University of Oxford and King Abdulaziz University). DAC was supported by the Pandemic Sciences Institute at the University of Oxford; the National Institute for Health Research (NIHR) Oxford Biomedical Research Centre (BRC); an NIHR Research Professorship; a Royal Academy of Engineering Research Chair; and the InnoHK Hong Kong Centre for Cerebro-cardiovascular Engineering (COCHE). AK and ANH were supported in part by the NIHR AI Award for Health and Social Care (AI-AWARD02183).

Declaration of competing interest

AK and ANH and by a research grant from GlaxoSmithKline unrelated to this work. ANH also receives grant funding from Novo Nordisk Pharmaceuticals unrelated to this work. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Dataset details

A.1. MIMIC-III

Mimic-III is a large, freely-available database comprising deidentified health data associated with over 40,000 patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012 [42]. The data includes demographics, vital signs, laboratory tests, medications, and more collected from a variety of hospital systems. It encompasses over 2 million notes including discharge summaries, radiology reports, and more.

A.2. i2b2

Originally released on the i2b2 website, but is now hosted via the Department of BioMedical Informatics (DBMI) data portal. The dataset is now referred to as the National NLP Clinical Challenges research datasets (n2c2), and is based on fully deidentified notes from the Research Patient Data Registry at Partners Healthcare System in Boston.

Appendix B. LoRA rank analysis

We provide a comparison of different LoRA ranks on task performance across each model in Fig. B.4.

Appendix C. Hyperparameters and hardware for downstream tasks

For the core experiments we utilised the HuggingFace [49] and Parameter Efficient Finetuning (PEFT) [48] libraries. For consistency and equal footing between model types, all experiments utilised a single NVIDIA RTX 3090 graphics card with 24 GB of VRAM. Due to this, however, the experiments utilising Llama-2-7b, even with LoRA, required a reduction in the precision of the model weights from fp32 to bfloat16 (see Table C.5).

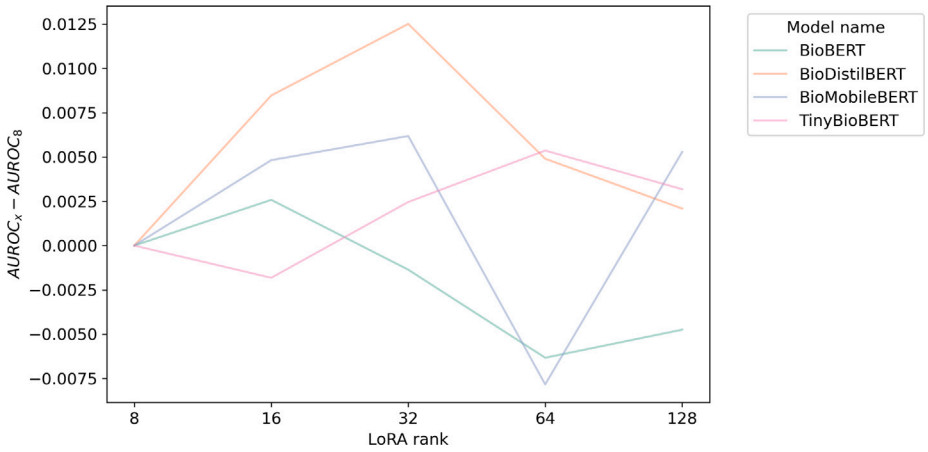

Fig. B.4. Differential effect of LoRA rank on performance of a model. The y-axis represents the difference in AUROC between the rank on the x-axis and rank=8.

Table C.5
The default hyperparameters for LoRA and IA^3 used in all experiments prior to the hyperparameter optimisation. For full fine-tuning the same learning rate ($3e{-}4$) and dropout (0.1) was used.

PEFT	Hyperparameter	Value
LoRA	r	8
	alpha	8
	dropout	0.1
	learning rate	$3e{-}4$
	target modules	[key, value]
	layers	all
IA^3	dropout	0.1
	learning rate	$3e-4$
	target modules	[key, value, feed-forward]
	layers	all

Table C.6
PEFT results for sequence classification and NER tasks dependent on domain pre-training received.

(a) Sequence classification task results

Model name	PEFT	ICD9-Triage	i2b2-2010-RE	MIMIC-LoS	Mimic-MP
BERTbase	Full	0.991	0.975	0.702	0.799
BERTbase	LORA	0.983	0.980	0.679	0.811
BioBERT	Full	0.991	0.982	0.711	0.812
BioBERT	LORA	0.991	0.985	0.697	0.828
BioClinicalBERT	Full	0.993	0.978	0.697	0.793
BioClinicalBERT	LORA	0.990	0.981	0.701	0.822
BioDistilBERT	Full	0.992	0.979	0.697	0.803
BioDistilBERT	LORA	0.993	0.988	0.704	0.822
BioMobileBERT	Full	0.992	0.980	0.697	0.809
BioMobileBERT	LORA	0.987	0.982	0.670	0.792
ClinicalDistilBERT	Full	0.994	0.980	0.697	0.822
ClinicalDistilBERT	LORA	0.995	0.989	0.710	0.836
ClinicalMobileBERT	Full	0.995	0.983	0.720	0.826
ClinicalMobileBERT	LORA	0.994	0.982	0.690	0.824

(b) NER task results

Model name	PEFT	i2b2-2010-NER	i2b2-2012-NER	i2b2-2014-NER
BERTbase	Full	0.806	0.792	0.974
BERTbase	LORA	0.673	0.697	0.951
BioBERT	Full	0.822	0.823	0.969
BioBERT	LORA	0.713	0.757	0.935
BioClinicalBERT	Full	0.846	0.820	0.960
BioClinicalBERT	LORA	0.704	0.746	0.920
BioDistilBERT	Full	0.809	0.794	0.965
BioDistilBERT	LORA	0.704	0.726	0.939
BioMobileBERT	Full	0.794	0.774	0.966
BioMobileBERT	LORA	0.649	0.654	0.938
ClinicalDistilBERT	Full	0.816	0.817	0.961
ClinicalDistilBERT	LORA	0.671	0.740	0.920

References

[1] OpenAI. GPT-4. Technical Report, 2023, URL http://arxiv.org/abs/2303.08774. arXiv:2303.08774 [cs].

[2] Touvron Hugo, Martin Louis, Stone Kevin, Albert Peter, Almahairi Amjad, Babaei Yasmine, et al. Llama 2: open foundation and fine-tuned chat models. 2023, URL http://arxiv.org/abs/2307.09288. arXiv:2307.09288 [cs].

[3] Claude 2. URL URL https://www.anthropic.com/news/claude-2.

[4] Jiang Albert Q, Sablayrolles Alexandre, Roux Antoine, Mensch Arthur, Savary Blanche, Bamford Chris, et al. Mixtral of experts. 2024, URL http://arxiv.org/abs/2401.04088. arXiv:2401.04088 [cs].

[5] Moradi Milad, Blagec Kathrin, Haberl Florian, Samwald Matthias. GPT-3 models are poor few-shot learners in the biomedical domain. 2021, URL https://arxiv.org/abs/2109.02555v2.

[6] Tunstall Lewis, Reimers Nils, Jo Unso Eun Seo, Bates Luke, Korat Daniel, Wasserblat Moshe, et al. Efficient few-shot learning without prompts. 2022, URL http://arxiv.org/abs/2209.11055. arXiv:2209.11055 [cs].

[7] Gutiérrez Bernal Jiménez, McNeal Nikolas, Washington Clay, Chen You, Li Lang, Sun Huan, et al. Thinking about GPT-3 in-context learning for biomedical IE? think again. 2022, http://dx.doi.org/10.48550/ARXIV.2203.08410, URL https://arxiv.org/abs/2203.08410. Publisher: arXiv Version Number: 3.

[8] Sun Xiaofei, Li Xiaoya, Li Jiwei, Wu Fei, Guo Shangwei, Zhang Tianwei, et al. Text classification via large language models. 2023, URL http://arxiv.org/abs/2305.08377. arXiv:2305.08377 [cs].

[9] Tang Ruixiang, Han Xiaotian, Jiang Xiaoqian, Hu Xia. Does synthetic data generation of LLMs help clinical text mining?. 2023, URL http://arxiv.org/abs/2303.04360. arXiv:2303.04360 [cs].

[10] Rohanian Omid, Nouriborji Mohammadmahdi, Clifton David A. Exploring the effectiveness of instruction tuning in biomedical language processing. 2023, URL http://arxiv.org/abs/2401.00579. arXiv:2401.00579 [cs].

[11] Devlin Jacob, Chang Ming-Wei, Lee Kenton, Toutanova Kristina. BERT: pre-training of deep bidirectional transformers for language understanding. 2019, URL http://arxiv.org/abs/1810.04805. arXiv:1810.04805 [cs].

[12] Liu Yinhan, Ott Myle, Goyal Naman, Du Jingfei, Joshi Mandar, Chen Danqi, et al. RoBERTa: A robustly optimized BERT pretraining approach. 2019, URL http://arxiv.org/abs/1907.11692. arXiv:1907.11692 [cs].

[13] Chen Qingyu, Du Jingcheng, Hu Yan, Keloth Vipina Kuttichi, Peng Xueqing, Raja Kalpana, et al. Large language models in biomedical natural language processing: benchmarks, baselines, and recommendations. 2023, URL https://arxiv.org/abs/2305.16326v1.

[14] Hinton Geoffrey, Vinyals Oriol, Dean Jeff. Distilling the knowledge in a neural network. 2015, URL http://arxiv.org/abs/1503.02531. arXiv:1503.02531 [cs, stat].

[15] Sanh Victor, Debut Lysandre, Chaumond Julien, Wolf Thomas. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2020, URL http://arxiv.org/abs/1910.01108. arXiv:1910.01108 [cs].

[16] Sun Zhiqing, Yu Hongkun, Song Xiaodan, Liu Renjie, Yang Yiming, Zhou Denny. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In: Proceedings of the 58th annual meeting of the association for computational linguistics. Association for Computational Linguistics; 2020, p. 2158–70. http://dx.doi.org/10.18653/v1/2020.acl-main.195, Online, URL https://aclanthology.org/2020.acl-main.195.

[17] Frantar Elias, Alistarh Dan. SparseGPT: massive language models can be accurately pruned in one-shot. 2023, URL http://arxiv.org/abs/2301.00774. arXiv:2301.00774 [cs].

[18] Luo Yun, Yang Zhen, Meng Fandong, Li Yafu, Zhou Jie, Zhang Yue. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. 2023, URL http://arxiv.org/abs/2308.08747. arXiv:2308.08747 [cs].

[19] Dettmers Tim, Pagnoni Artidoro, Holtzman Ari, Zettlemoyer Luke. QLoRA: efficient finetuning of quantized LLMs. 2023, URL https://arxiv.org/abs/2305.14314v1.

[20] Lester Brian, Al-Rfou Rami, Constant Noah. The power of scale for parameter-efficient prompt tuning. 2021, URL http://arxiv.org/abs/2104.08691. arXiv:2104.08691 [cs].

[21] Li Xiang Lisa, Liang Percy. Prefix-tuning: optimizing continuous prompts for generation. 2021, URL http://arxiv.org/abs/2101.00190. arXiv:2101.00190 [cs].

[22] Hu Edward J, Shen Yelong, Wallis Phillip, Allen-Zhu Zeyuan, Li Yuanzhi, Wang Shean, et al. LoRA: low-rank adaptation of large language models. 2021, URL http://arxiv.org/abs/2106.09685. arXiv:2106.09685 [cs].

[23] Liu Haokun, Tam Derek, Muqeeth Mohammed, Mohta Jay, Huang Tenghao, Bansal Mohit, et al. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. 2022, URL http://arxiv.org/abs/2205.05638. arXiv:2205.05638 [cs].

[24] Alsentzer Emily, Murphy John R, Boag Willie, Weng Wei-Hung, Jin Di, Naumann Tristan, et al. Publicly available clinical BERT embeddings. 2019, URL http://arxiv.org/abs/1904.03323.

[25] Lehman Eric, Hernandez Evan, Mahajan Diwakar, Wulff Jonas, Smith Micah J, Ziegler Zachary, et al. Do we still need clinical language models?. 2023, URL http://arxiv.org/abs/2302.08091. arXiv:2302.08091 [cs].

[26] Lorge Isabelle, Joyce Dan W, Taylor Niall, Nevado-Holgado Alejo, Cipriani Andrea, Kormilitzin Andrey. Detecting the clinical features of difficult-to-treat depression using synthetic data from large language models. 2024, URL http://arxiv.org/abs/2402.07645. arXiv:2402.07645 [cs].

[27] Yu Hao, Yang Zachary, Pelrine Kellin, Godbout Jean Francois, Rabbany Reihaneh. Open, closed, or small language models for text classification?. 2023, URL http://arxiv.org/abs/2308.10092. arXiv:2308.10092 [cs].

[28] Rohanian Omid, Nouriborji Mohammadmahdi, Jauncey Hannah, Kouchaki Samaneh, ISARIC Clinical Characterisation Group, Clifton Lei, et al. Lightweight transformers for clinical natural language processing. 2023, URL http://arxiv.org/abs/2302.04725. arXiv:2302.04725 [cs].

[29] Li Yuanzhi, Bubeck Sébastien, Eldan Ronen, Giorno Allie Del, Gunasekar Suriya, Lee Yin Tat. Textbooks are all you need II: phi-1.5 technical report. 2023, URL http://arxiv.org/abs/2309.05463. arXiv:2309.05463 [cs].

[30] Hughes Alyssa. Phi-2: the surprising power of small language models. 2023, URL https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/.

[31] Singhal Karan, Azizi Shekoofeh, Tu Tao, Mahdavi S Sara, Wei Jason, Chung Hyung Won, et al. Large language models encode clinical knowledge. Nature 2023;620(7972):172–80. http://dx.doi.org/10.1038/s41586-023-06291-2, URL https://www.nature.com/articles/s41586-023-06291-2. Publisher: Nature Publishing Group.

[32] Kweon Sunjun, Kim Junu, Kim Jiyoun, Im Sujeong, Cho Eunbyeol, Bae Seongsu, et al. Publicly shareable clinical large language model built on synthetic clinical notes. 2023, URL http://arxiv.org/abs/2309.00237. arXiv:2309.00237 [cs].

[33] Ding Ning, Qin Yujia, Yang Guang, Wei Fuchao, Yang Zonghan, Su Yusheng, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. Nat Mach Intell 2023;5(3):220–35. http://dx.doi.org/10.1038/s42256-023-00626-4, URL URL https://www.nature.com/articles/s42256-023-00626-4. Number: 3 Publisher: Nature Publishing Group.

[34] Taylor Niall, Zhang Yi, Joyce Dan W, Gao Ziming, Kormilitzin Andrey, Nevado-Holgado Alejo. Clinical prompt learning with frozen language models. IEEE Trans Neural Netw Learn Syst 2023;1–11. http://dx.doi.org/10.1109/TNNLS.2023.3294633, URL https://ieeexplore.ieee.org/document/10215061.

[35] Gema Aryo Pradipta, Daines Luke, Minervini Pasquale, Alex Beatrice. Parameter-efficient fine-tuning of LLaMA for the clinical domain. 2023, URL http://arxiv.org/abs/2307.03042. arXiv:2307.03042 [cs].

[36] Jiao Xiaoqi, Yin Yichun, Shang Lifeng, Jiang Xin, Chen Xiao, Li Linlin, et al. TinyBERT: distilling BERT for natural language understanding. 2020, URL http://arxiv.org/abs/1909.10351. arXiv:1909.10351 [cs].

[37] Wolf Thomas, Debut Lysandre, Sanh Victor, Chaumond Julien, Delangue Clement, Moi Anthony, et al. Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. Association for Computational Linguistics; 2020, p. 38–45. http://dx.doi.org/10.18653/v1/2020.emnlp-demos.6, Online, URL https://aclanthology.org/2020.emnlp-demos.6.

[38] Rohanian Omid, Nouriborji Mohammadmahdi, Kouchaki Samaneh, Clifton David A. On the effectiveness of compact biomedical transformers. Bioinformatics 2023;39(3). http://dx.doi.org/10.1093/bioinformatics/btad103, btad103.

[39] Harris Charles R, Millman K Jarrod, van der Walt Stéfan J, Gommers Ralf, Virtanen Pauli, Cournapeau David, et al. Array programming with numpy. Nature 2020;585(7825):357–62. http://dx.doi.org/10.1038/s41586-020-2649-2, URL https://www.nature.com/articles/s41586-020-2649-2. Publisher: Nature Publishing Group.

[40] Meehan Alan J, Lewis Stephanie J, Fazel Seena, Fusar-Poli Paolo, Steyerberg Ewout W, Stahl Daniel, et al. Clinical prediction models in psychiatry: a systematic review of two decades of progress and challenges. Mol Psychiatry 2022;27(6):2700–8. http://dx.doi.org/10.1038/s41380-022-01528-4, URL https://www.nature.com/articles/s41380-022-01528-4. Number: 6 Publisher: Nature Publishing Group.

[41] Lee Jinhyuk, Yoon Wonjin, Kim Sungdong, Kim Donghyeon, Kim Sunkyu, So Chan Ho, et al. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 2020;36(4):1234–40. http://dx.doi.org/10.1093/bioinformatics/btz682, Publisher: Oxford University Press.

[42] Johnson Alistair EW, Pollard Tom J, Shen Lu, Lehman Li Wei H, Feng Mengling, Ghassemi Mohammad, et al. MIMIC-III, a freely accessible critical care database. Sci Data 2016;3. http://dx.doi.org/10.1038/sdata.2016.35, Publisher: Nature Publishing Groups.

[43] Aken Betty Van, Papaioannou Jens-Michalis, Mayrdorfer Manuel, Budde Klemens, Gers Felix, Loeser Alexander. Clinical outcome prediction from admission notes using self-supervised knowledge integration. In: Proceedings of the 16th conference of the european chapter of the association for computational linguistics: Main volume. Association for Computational Linguistics; 2021, p. 881–93. http://dx.doi.org/10.18653/v1/2021.eacl-main.75, Online, URL https://aclanthology.org/2021.eacl-main.75.

[44] Uzuner Özlem, South Brett R, Shen Shuying, DuVall Scott L. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. J Am Med Inform Assoc : JAMIA 2011;18(5):552–6. http://dx.doi.org/10.1136/amiajnl-2011-000203, URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3168320/.

[45] Sun Weiyi, Rumshisky Anna, Uzuner Ozlem. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. J Am Med Inform Assoc : JAMIA 2013;20(5):806–13. http://dx.doi.org/10.1136/amiajnl-2013-001628, URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3756273/.

[46] Stubbs Amber, Kotfila Christopher, Uzuner Özlem. Automated systems for the de-identification of longitudinal clinical narratives: overview of 2014 i2b2/uthealth shared task track 1. J Biomed Inform 2015;58:S11–9. http://dx.doi.org/10.1016/j.jbi.2015.06.007, URL https://www.sciencedirect.com/science/article/pii/S1532046415001173.

[47] Akiba Takuya, Sano Shotaro, Yanase Toshihiko, Ohta Takeru, Koyama Masanori. Optuna: A next-generation hyperparameter optimization framework. 2019, URL http://arxiv.org/abs/1907.10902. arXiv:1907.10902 [cs, stat].

[48] Mangrulkar Sourab, Gugger Sylvain, Debut Lysandre, Belkada Younes, Paul Sayak, Bossan Benjamin. Peft: state-of-the-art parameter-efficient fine-tuning methods. 2022, URL https://github.com/huggingface/peft.

[49] Wolf Thomas, Debut Lysandre, Sanh Victor, Chaumond Julien, Delangue Clement, Moi Anthony, et al. Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. Association for Computational Linguistics; 2020, p. 38–45. http://dx.doi.org/10.18653/v1/2020.emnlp-demos.6, Online, URL https://aclanthology.org/2020.emnlp-demos.6.