MediLoRA: Training Small Models for Medical QA with QLoRA

Yuxuan Zhang
Halıcıoğlu Data Science Institute
yuz226@ucsd.edu

Ananay Sharma
Halıcıoğlu Data Science Institute
ans037@ucsd.edu

Swapnil Ghosh
Halıcıoğlu Data Science Institute
s4ghosh@ucsd.edu

Shreyan Sood
Halıcıoğlu Data Science Institute
s1sood@ucsd.edu

Abstract

We approach the challenge of proprietary data's dominance in medical language modeling for diagnosis conversation and medical reasoning capabilities with a core hypothesis: a good base model checkpoint is as crucial as dataset size and quality. We train and release Medilora[1], a series of low-rank adapters on a diverse yet modestly-sized corpus of medical notes using QLoRA on the strongest generalist open-weights model available in the 7B size family. Experiment results show that our adapter models approach the performance of that from a fine-tuned 70B sized large model trained on 2000 times of our data, demonstrating the promising prospect of leveling the data and compute playing fields with parameter-efficient fine-tuning methods on a good generalist base model. We culminate with a sharp critique of our method's limitations.

1 Introduction

Auto-regressive language models, acknowledged as proficient instruction followers and domain experts [20], are pivotal in healthcare applications like medical question answering and diagnosis. Unlike large-scale projects like Meditron, which require substantial resources such as 128 A100 GPUs, our focus is on open-weight models from smaller entities using synthetic data [23] and optimized architectures [2]. We hypothesize that base model selection influences data efficiency in medical language modeling. To test this, we train 7B-sized models on a small, relevant medical dataset using quantized low-rank adaptation, a resource-efficient approach.

Confronted with a lack of robust diagnosis benchmarks, we resorted to standard medical benchmarks like MMLU-Medical, PubMedQA [15], and MedQA [28] for evaluation. Our models, trained on just 0.05% of the data used by state-of-the-art counterparts, show promising results. Notably, our PubMedQA adapter achieved over 75% accuracy, and our MedQA adapter approached the performance of the leading Meditron-7B model.

A limitation of our study is understanding performance degradation relative to the base model. We found that our adapters scored slightly lower than our base model, OpenHermes-2.5-Mistral-7B, on the MMLU-Medical benchmark. We discuss these limitations in detail at the end of the paper.

[1] https://huggingface.co/Medilora

Unpublished. Reference Only.

2 Background & Related Work

2.1 Medical Langauge Models

The landscape of medical language models has witnessed significant strides, largely propelled by large transformer models. These models, have the ability to follow instructions and engage in coherent conversations. They are well-suited for applications in medical language modeling, particularly in conversational settings and question-answering formats. Notable examples of open-source conversational language models include ChatDoctor [16], MedAlpaca [9], Clinical-Camel [24], Med-PaLM [21], and Meditron [3]. Their effectiveness can be attributed to innovative decoding system designs, such as self-consistency chain-of-thoughts and monte-carlo tree search decoding, which enhance their overall performance in medical dialogue and reasoning tasks. These advancements lay a solid foundation for our exploration into mitigating the dominance of proprietary data in medical language modeling, emphasizing the importance of both model architecture and decoding strategies in achieving superior performance.

Building upon the progress made in medical language models, a pivotal aspect of recent research lies in the continual refinement and adaptation of these models. The endeavor to optimize their performance involves exploring innovative approaches such as continual retraining and domain-specific fine-tuning. As demonstrated by Gupta et al. [6], the continual pretraining of large language models presents an efficient strategy, enabling the incorporation of new data without restarting the training process. This aligns with the broader trend seen in models like PMC-LLaMA [27], which emphasize the importance of adapting pretrained models to the evolving landscape of medical knowledge.

2.2 Continued Pretraining

Continued pretraining is crucial for large language models (LLMs), even though it demands substantial data. Despite the resource challenges, research highlights its effectiveness in enhancing model performance. While earlier studies focused on encoder-decoder models like BERT and T5, recent research explores continued pretraining nuances across various model architectures. These investigations deepen our understanding and provide insights for optimizing continued pretraining strategies,

In Gupta et al's [6] study on efficient continual pretraining for large language models, the authors advocate for updating models with new data instead of restarting training. They emphasize the crucial role of learning rate adjustments during the warm-up phase, finding that re-increasing the learning rate enhances compute efficiency when adapting to new datasets.

Gururangan et al's [8] research explores the relevance of tailoring pretrained language models to specific task domains in NLP. Across four domains and eight tasks, they find that a second phase of in-domain pretraining improves performance in both high- and low-resource settings. Task-adaptive pretraining further enhances performance, and adapting to a task corpus using simple data selection strategies is effective in resource-constrained scenarios. Overall, multi-phase adaptive pretraining consistently leads to substantial gains in task performance.

The significance of continual retraining was also highlighted by the study on PMC-LLaMA [27]. While LLMs exhibit commendable natural language understanding capabilities, their precision in domain-specific applications, particularly in fields like medicine, can be compromised due to a lack of specialized knowledge. PMC-LLaMA addresses this challenge through a systematic adaptation process, injecting domain-specific knowledge from a vast repository of biomedical papers and textbooks. Importantly, the continual retraining approach allows the model to evolve and align more closely with evolving medical knowledge and requirements.

3 Methods

3.1 Datasets

Our project began by discovering the The Medical Information Mart for Intensive Care (MIMIC) III dataset and expanded into the textbook corpus contained in the MedQA United States Medical License Exam training data and the Clinical Practice Guideline corpus generously released by the Meditron

Training Datasets		Fine-tuning Datasets	
Name	Training Tokens	Name	Training Tokens
MIMIC-III	144.3M	PubMedQA	100.3M
Textbook	23.6M	MedQA	3.6M
CPG (Guideline)	107M		
Total Training Tokens	274.9M	Total Fine-Tuning Tokens	103.9M

Table 1: **Dataset sizes in token.** Note that the Clinical Practice Guideline token count was reported by the Meditron team. We tokenized the dataset using sentencepiece and obtained 98.2M tokens.

team. Given the completeness and literary granularity of the texts, we designed our experiment such that we use these three datasets for completion training, equivalent to continued pretraining. We use the MedQA and PubMedQA training data for conversation fine-tuning our model. For a detailed breakdown of the dataset sizes, see Table 1.

3.1.1 MIMIC-III Dataset

The Medical Information Mart for Intensive Care (MIMIC) III dataset, covering over 46,000 patients, includes demographics, vital signs, lab results, medications, and clinical notes. This comprehensive dataset is crucial for our project, offering detailed insights into patient health records.

Our data cleaning process for the MIMIC-III dataset, particularly the 'noteevents' table, involved iterative refinement to extract pertinent medical information while preserving anonymity. We connected to the PostgreSQL database, concentrating on the 'text' column with medical notes. Key sections like patient history and lab studies were identified for their clinical relevance. Regular expressions (regex) were employed to segment and integrate these texts, ensuring data coherence. The cleaning process involved removing irrelevant data and placeholders (often in square brackets), improving format for readability, and maintaining medical accuracy and patient confidentiality.

The Medical Information Mart for Intensive Care (MIMIC) III encompasses de-identified health-related data associated with a vast cohort of over 46,000 patients. The dataset encapsulates diverse information, including demographics, vital sign measurements, laboratory test results, medications, and detailed narrative clinical notes. Its extensive and varied nature positions it as a pivotal resource for our project, promising in-depth insights into patient health records.

We perform extensive data cleaning on the MIMIC-III dataset to make sure the model learns from it effectively. The data cleaning methodology applied to the 'noteevents' table within the MIMIC-III dataset was an iterative process aimed at extracting meaningful medical information while addressing anonymized placeholders and ensuring patient confidentiality. Initially, the connection to the PostgreSQL database housing the MIMIC-III dataset was established using $psycopg2$, focusing on the 'text' column containing diverse medical notes. Within these notes, a structured subset of sections vital to patient records was identified, delineated by headers such as history of present illness, past medical history, laboratory studies, etc. These sections encapsulate essential medical details crucial for patient care and treatment planning.

To delineate these sections accurately, regular expression (regex) patterns were utilized to parse and extract content to merge and consolidate these segmented text elements into cohesive and comprehensive units. Further refining steps involved eliminating extraneous information, formatting adjustments to enhance readability, and the removal of content enclosed in square brackets, often indicative of non-essential data or placeholders within the structured data were observed, likely employed to safeguard patient privacy by anonymizing personal details. The data cleaning procedure focused on maintaining the integrity of medical information while respecting confidentiality by addressing these anonymized segments.

3.1.2 Synthetic Data Generation on the MIMIC-III

Inspired by [26], we initially planned to generate Q&A pairs for each medical note in the MIMIC-III dataset using the self-instruction concept and the GPT-4 API. The self-instruction concept involves utilizing the generative capabilities of a language model to create its own instruction data. GPT-4

was tasked with generating general Q&A pairs for each medical note in the dataset. This approach effectively bootstrapped the instruction tuning process, eliminating the need for manually crafted instruction sets.

This method proved to be effective in producing high-quality Q&A pairs. However, the token-based pricing model of the GPT-4 API proved to be too expensive for the project's budget. To address this issue, we experimented with limiting the number of tokens used and employing the GPT-3.5 Turbo model. While this approach reduced costs, the generated questions became overly specific and lacked the desired level of quality. We acknowledge the importance of prudence in leveraging synthetic generated data and our budget limit, and choose to not include any synthetic conversational data in our training.

3.1.3 US Medical License Exam Textbooks

We further sought out the US Medical License Exam Textbooks corpus as part of the MedQA open dataset. It consists of 18 textbooks. Each text file is formatted in a way that page and literature elements, such as title, subtitle and footnotes, are put into its own lines. Further, each valid lines of texts are separated by one empty line. We process the text data by removing the empty lines and concatenating lines shorter than 5 tokens into its subsequent line. Then, we remove non-alphanumeric characters and punctuation marks to streamline text content and reduce noise. Due to the nature of this corpus, we speculate that training on this dataset will further improve the model's capability in generating helpful answers to hard or unseen queries.

3.1.4 Clinical Practice Guidelines

The Clinical Practice Guidelines [1] serve as comprehensive, research-oriented guides in healthcare, aiding in evidence-based decision-making related to diagnosis, treatment, and management. Formulated through expert consensus and synthesizing recent evidence, they offer top-tier, practical advice. Their production spans different scales, from international to hospital-specific, involving diverse entities like medical associations, governmental bodies, and hospitals. The Meditron team released 35,733 articles across many medical domains and geographical granularities from CCO, CDC, CMA, ICRC, NICE, SPOR, WHO and WikiDoc.

3.1.5 PubMedQA

PubMedQA is a large medical research-oriented question-answering dataset with elaborated question, context, long answers and ground truth labels between "yes," "no" and "maybe." The questions are constructed using PubMed paper abstracts and the model should give a deterministic ground truth label based on the relevant abstracts added into the prompt as context. PubMedQA contains 1,000 expert labeled entries, 61.2k unlabeled and 211.3k artificially generated QA instances.

It is natural to seek out the original research texts used by the PubMedQA due to its question-generation technique on paper abstracts. We discover large scale PubMed paper abstracts corpus on HuggingFace uploaded by the user Younwoo Choi ($ywchoi$)[2] who also hosts cleaned PubMed Central research text corpus. The PubMed Abstract dataset is split into 10 splits ($ywchoi/pubmed\_abstract\_0$, etc.) During our experiment, we fire training runs on $ywchoi/pubmed\_abstract\_0$ and found that training on even one corpus would shoot out of our budget by factors in the 50s, not to mention all the abstracts and PMC text corpus. We unfortunately do not go down this route.

3.1.6 Unused Medical Conversational Dataset

During our data discovery, we evaluate the effectiveness of several conversational datasets. Notably among them is the HelathCareMagic-100k dataset released by the ChatDoctor team. It contains 100,000 one-turn conversations between patients and healthcare professionals on a medical chat forum. Each conversation spans diverse medical topics such as coughs, rashes, and pain. We randomly sampled 50 entries for manual examination and found that the chat qualities were underwhelming. Specifically, the doctor responses contain incomplete sentences and paragraphs, and the response

[2]https://huggingface.co/ywchoi

quality exhibit large variance. We determine the necessity of careful cleaning and pruning, and unfortunately left it out of this project's scope.

3.2 Base Model Selection

Following the success of ChatGPT [13], many strong open-weights models were trained and released. The Llama model family [25] determined the primary open-weight model taxonomy into 7B, 13B, 34B and 70B sizes. In this work, we consider 7B, the lowest model size in this range with an abundance of self-proclaimed "powerful" models. Llama2 7B is a strong pretrained model with open-weights and open training data; Mistral-7B [14] is an open-weights model with exceptional performance on all major benchmarks, even surpassing Llama2-13B on some of them. We consider the effects of using different starting models from several perspectives. On one hand, larger model sizes perform better than smaller ones in the same family trained on the same datasets, on the other hand, we have to consider the objective existence, availability and ease of access of stronger fine-tunes of open models, many of which perform significantly better on benchmarks than models larger in size.

The Orca-2 models [17] came out conveniently as we began our base model selection phase. It is a family of models claimed to have increased reasoning capabilities via cautious reasoning. Due to the complicated nature and high stakes of medical conversations, we originally planned to incorporate both the Orca-2-7B and 13B models in our subsequent training runs as a speculative performance upperbound, however, an experiment ran by Nous Research [3] showed that the leading base model in our experiment, the OpenHermes-2.5-Mistral-7B, beats Orca-13B on the BigBench [22] benchmark by a wide margin. Given the information, we chose the OpenHermes-2.5-Mistral-7B model from Nous Research as our base model.

We remain aware that the Mistral model was trained using undisclosed data in contrast to the open data information from the Llama model family. While we could have not fine-tuned on more data even with access to data information, knowing what went into the model would give us stronger confidence in our evaluations, as well as comparing pretrained data sizes based on neural scaling laws. [12]

4 Experiments

Our project's aim is to make medical language modeling more accessible. We utilized the Axolotl framework from the OpenAccess Collective[4] for its user-friendly yaml configuration and compatibility with frameworks like Huggingface Transformers and DeepSpeed. Because our budget is limited and unpredictable at time, we used various GPU sizes for each dataset. Specifically, we used 2 A5000 GPUs for the textbook corpus, 4 RTX 4090s for MIMIC-III and guideline datasets, and 4 A100 GPUs for fine-tuning on PubMedQA and MedQA. Our development was challenged by the lack of a stable, dedicated instance, leading to unexpected effort in debugging training systems, despite Axolotl's relative simplicity.

4.1 Training Settings and Hyperparameters

We train all our checkpoints using quantized low-rank adaptation, loading the model weights in 4-bits while performing training operations in bf16 precision. We adopt most QLoRA training settings from the QLoRA paper [4], using rank $r = 8$, LoRA $\alpha = 16$, drop-out = 0.05, and targeting the q_proj and v_proj modules in the model. We accumulate gradients for 4 steps and use 2 mini-batch size, which means accumulating gradients over 2 mini-batches and updating the model weights after 4 training steps. This technique enables training using larger effective batch sizes without requiring a corresponding increase in memory usage. We use the $adamw\_bnb\_8bit$ optimizer, which is an 8-bit processed Adam optimizer with weight decay. We schedule our learning rates with cosine annealing and set the initial learning rate as $2 \cdot 10^{-4}$. The cosine learning rate uses 100 steps for warm up.

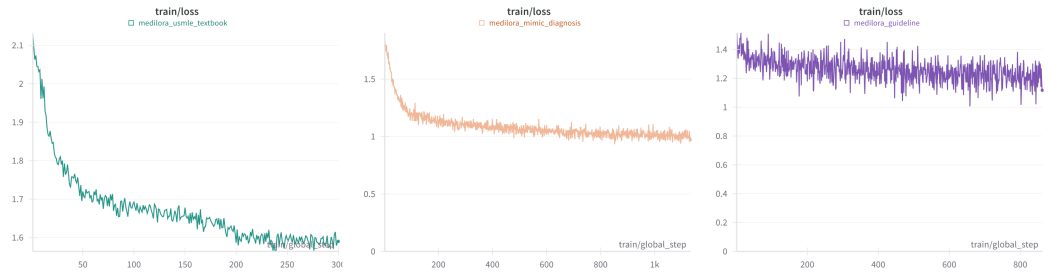

Figure 1: **Completion Losses.** Left: loss on textbook dataset; middle: loss on MIMIC-III dataset; right: loss on the CPG (Guideline) dataset.

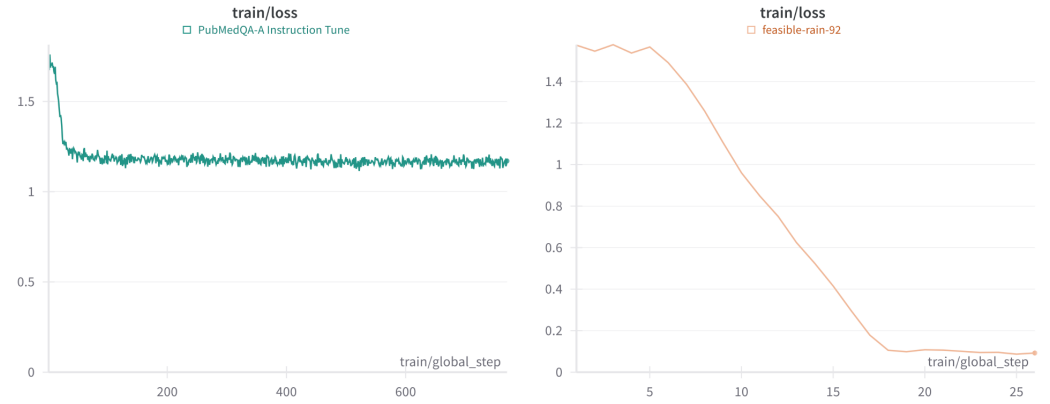

Figure 2: **Fine-tuning Losses.** Left: loss on PubMedQA training set; right: loss on MedQA 5-option training set.

4.2 Supervised Fine-tuning

After we train the base model on completion, we perform supervised instruction tuning on both the PubMedQA and MedQA train splits. For each entry in PubMedQA, we concatenate question after the context as input, and concatenate long answers behind ground truth labels. For MedQA, we concatenate the options together, add it behind each question and use that as input. We use the ground truth label and option text as target output, and we retain the parenthesis around the ground truth label. We also prepare system prompts for each downstream dataset.

4.2.1 Chat Formatting

The base model, OpenHermes-2.5-Mistral-7B, is trained on conversational data in the ChatML format, therefore, we also format both PubMedQA and MedQA datasets according to the ChatML format. ChatML conversations are represented by lists of dictionaries which contents are the messages, with special token <|im_start|>, the message role, a new line, and the message text content. Each message is post-pended with another special token <|im_end|>. During supervised fine-tuning, we only compute the loss with respect to the output tokens, including <|im_start|> and <|im_end|> as they count towards the system's output. The system prompts use for each dataset is shown in Table 2. We append chat format examples in the appendix.

4.3 Inference

Various methods enhance language model outputs, like Self-consistency and reasoning augmentation [10]. However, we limit augmentation to preserve raw model performance assessment.

[3]https://x.com/Teknium1/status/1726846755344634020
[4]https://github.com/OpenAccess-AI-Collective/axolotl

Dataset	System Prompts
PubMedQA	Consider yourself a practicing medical professional tackling real-world medical queries. Draw upon your expertise in basic and clinical sciences, medical knowledge, and understanding of health mechanisms, diseases, patient care, and therapy modalities. Answer the following multiple-choice question by selecting one of the options: 'yes,' 'no,' or 'maybe.' Choose the most appropriate response based on your expertise and understanding.
MedQA	You are a medical doctor taking the US Medical Licensing Examination. You need to demonstrate your understanding of basic and clinical science, medical knowledge, and mechanisms underlying health, disease, patient care, and modes of therapy. Show your ability to apply the knowledge essential for medical practice. For the following multiple-choice question, select one correct answer from A to E. Base your answer on the current and standard practices referenced in medical guidelines.

Table 2: **System prompts for each dataset.** MMLU-Medical does not take customized system prompts. For MedQA, we modify "from A to E" to "from A to D" with 5 and 4-option test sets respectively.


```
[
  {
    "from": "user",
    "value": "Context: Psammaplin A (PsA) is a natural product isolated from marine sponges, which has
    been demonstrated to have anticancer activity against several human cancer cell lines via the
    induction of cell cycle arrest and apoptosis. New drugs that are less toxic and more effective
    against multidrug-resistant cancers are urgently needed. We tested cell proliferation, cell cycle
    progression and autophagic cell death pathway in doxorubicin-resistant MCF-7 (MCF-7/adr) human
    breast cancer cells. The potency of PsA was further determined using an in vivo xenograft model.
    Question: Does psammaplin A induce Sirtuin 1-dependent autophagic cell death in
    doxorubicin-resistant MCF-7/adr human breast cancer cells and xenografts?"
  },
  {
    "from": "assistant",
    "value": "yes. PsA significantly inhibited MCF-7/adr cells proliferation in a
    concentration-dependent manner, with accumulation of cells in G2/M phase of the cell cycle. PsA
    significantly decreased SIRT1 enzyme activity and reduced expression of SIRT1 protein in the
    cultured cells with greater potency than sirtinol or salermide. Acetylation of p53, a putative
    target of SIRT1, increased significantly following PsA treatment. In addition, PsA markedly
    increased the expression levels of autophagy-related proteins. In support of this, it was found
    that PsA significantly increased the expression of damage-regulated autophagy modulator (DRAM), a
    p53-induced protein."
  }
]
```


Figure 3: **PubMedQA in ChatML Format.** System prompt is not included here, but is added during fine-tuning.

For inference, we integrate adapters into base models to form checkpoints, then quantize them using $Q3\_K\_L$ quantization via llama.cpp[5], resulting in a single model file in $gguf$ format. $Q3\_K\_L$ employs 3-bit weight quantization, converting weights w from quants q as $w = d \cdot q$, where d is the block scale, using 3.4375 bits per weight compared to the 32-bit full-precision of base models[6].

Our inference utilizes top-40 token sampling, with manual substring extraction to determine valid response rates. For PubMedQA, responses are validated based on their start with "yes", "no", or "maybe", while for MedQA, responses matching the "(A)","(B)","(C)","(D)" format are considered valid. The valid response rates for our PubMedQA and MedQA adapters on their test sets are 99.9% and 99.8%, respectively.

There have been many effective inference methods and systems to improve the output of a language model, such as the Self-consistency techniques and reasoning augmentation [10]. However, we deliberate that excessive augmentation would deter accurate and convincing evaluation of a model's raw performance on tasks.

We prepare models for inference by merging the adapters with the corresponding base models to obtain checkpoints, and subsequently quantize each checkpoint with $Q3\_K\_L$ quantization using llama.cpp[7] to obtain a single model file in the $gguf$ format. In short, $Q3\_K\_L$ uses a type 0 3-bit weight quantization. Type 0 means the weights w are obtained from quants q using $w = d \cdot q$ where d is the block scale. 3-bit quantization performs quantization on super-blocks containing 16 blocks, where each block have 16 weights. Block scales are quantized with 6 bits.[8] This end up using 3.4375 bits per weight, in contrast to the 32-bit full-precision weights in base models.

We use the llama-cpp-python[9] for a pre-packaged chat-compatible inference framework. For each entry in the test sets of each benchmark, we pre-process the inputs following the same steps in our training phase and obtain responses from models without the special tokens <|im_start|> or <|im_end|>. We use top-40 token sampling with no in-context learning augmentation, meaning we do not test out the few-shot capabilities of our models. We take note that the llama-cpp-python inference has a default setting of top-40 token sampling, which gave us a fair comparison to the concurrent state-of-the-art evaluation results.

We uses top-40 token sampling to obtain the responses and perform manual substring extraction on the responses and check the valid response rate. For PubMedQA, we check whether a response starts with "yes", "no" or "maybe," case insensitive. For MedQA, we check whether the first 3-character substring is either "(A)","(B)","(C)","(D)", the same format used in our fine-tuning. On their corresponding fine-tuning test sets, our PubMedQA adapter has a valid response rate of 99.9%, our MedQA adapter has a valid response rate of 99.8%.

4.4 Medical Benchmarks

To this date, there are no open, robust and privacy-prioritized diagnosis evaluations for language models trained with autoregressive objectives. Due to the aforementioned quality issues in the HealthcareMagic dataset, we resort to using existing conventional language modeling benchmarks: MMLU-Medical, PubMedQA and MedQA. We subsequently evaluate our PubMedQA adapter on TruthfulQA, GPT4All and BigBench-Multiple-Choice to compare with the base model.

We obtain MMLU-Medical, TruthfulQA, GPT4All and BigBench evaluations with the Language Model Evaluation Harness framework[5], which can take a HuggingFace model. We obtain PubMedQA and MedQA evaluations with llama-cpp-python response generation. All benchmarks use accuracy score as the metric.

The choice of the medical benchmarks we choose is only based on their relevance to medical tasks, not for their ability to elicit reasoning and diagnosis capabilities from language models. We speculate that this remain true until a rigorous multi-turn multi-direction conversational diagnosis dataset is released.

[5] https://github.com/ggerganov/llama.cpp

[6] https://github.com/ggerganov/llama.cpp/pull/1684

[7] https://github.com/ggerganov/llama.cpp

[8] https://github.com/ggerganov/llama.cpp/pull/1684

[9] https://llama-cpp-python.readthedocs.io/en/latest/

4.4.1 MMLU-Medical

In this project, we define MMLU-Medical to be a subset of sub-tasks in the MMLU benchmark [11]: Anatomy, College Bio, College Medicine, Professional Medicine, Genetics, Virology, Clinical Knowledge, High-school Biology and Nutrition. We obtain the final result for MMLU-Medical with the average of our PubMedQA adapter's score across all 9 subjects.

We particularly recall an earlier discussion where Aghajanyan et al.[10] found that formatting multiple choice labels as (A) instead of "A" causes improvements in MMLU performance. Because MMLU is a multiple choice benchmark, we hold a lot more suspicion regarding the benchmark results. We also acknowledge the possibility of tracking whether the lm-eval framework we used have any techniques remedying this effect.

4.5 Evaluation Results

We use two baselines for our evaluation: Meditron-7B and our base model OpenHermes-2.5-Mistral-7B. Meditron [3] is a suite of 7B and 70B Llama2 fine-tune models trained on 2000 times more data than ours. It achieved dominating performance on medical benchmarks. Considering we already started with a different base model than Llama-2-7B, we evaluate against the Meditron-7B to reduce "model size" from the number of variables in question.

During our MMLU-Medical evaluation, we take the Meditron-7B from their HuggingFace repository[11] and report their result following the exact evaluation setup as ours. This gave us different scores than the official reported scores, but it is effectively explained via their extensive decoding system design (self-consistency, etc.)

	Accuracy (%, ↑)									
Model	Anatomy	ClinicalKG	C-Bio	C-Medicine	HS-Bio	Genetics	Nutrition	Pro-Med	Virology	Average
Base-adapter	59.3	67.2	70.8	63.6	74.2	69	70.2	64.7	49.4	65.4
PubMedQA-adapter	56.3	67.6	66.0	62.4	73.9	67	71.6	65.8	48.2	64.3
MedQA-adapter	56.3	63.8	67.4	56.6	72	66	66.3	62.9	50.6	62.4
Meditron-7B	41.5	36.2	39.6	28.9	33.2	43	36.9	24.6	27.7	34.6
OpenHermes-Mistral	56.3	68.3	70.1	57.2	76.8	70	73.5	66.9	53.6	65.9

Table 3: **MMLU-Medical Results.** Note that we run the evaluation on Meditron-7B off-the-shelf from their HuggingFace repository.

4.5.1 MMLU-Medical

We evaluate our completion-trained adapter, PubMedQA adapter and the OpenHermes-2.5-Mistral-7B model on MMLU-Medical and report the results in Table 3.

The Average column provides an overall view of each model's performance across different medical domains. The base model, OpenHermes-2.5-Mistral-7B, achieved the highest average score of 0.659 followed closely by our model Medilora-base with an average score of 0.654.

Examining specific domains, Medilora-base demonstrated particularly strong performance in college-level subjects such as Biology (70.8) and Medicine (63.58). This proficiency might be attributed to its training on the USMLE dataset, where ample examples relating to college-level subjects were available. However, Medilora might have encountered limitations in accessing a substantial volume of examples across more specialized domains like Genetics, Virology, and others, potentially impacting its performance in those areas.

4.5.2 PubMedQA

The results from the evaluation of different models in the PubmedQA dataset demonstrated variations in their performance. Notably, models like meditron-70b and gpt-4-base showcased higher scores, achieving 81.6 and 80.4 respectively. On the other hand, Medilora-base achieved a moderate score of

[10] https://x.com/armenagha/status/1669084129261162497

[11] https://huggingface.co/epfl-llm/meditron-7b

Model Name	PubmedQA Accuracy
PubMedQA-adapter	75.8
Meditron-7B (base, few-shot)	69.3
Meditron-7B (fine-tune)	74.4
GPT-4-base	80.4
Meditron-70B (Top-Token)	80.0

Table 4: **PubmedQA Results**

57.4. Using specific prompts that matched the context significantly improved the model's performance to 75.8.

Medilora's performance saw substantial improvement when prompted with comprehensive and structured contextual information. This tailored approach evidently empowered the model to better comprehend and contextualize medical scenarios, resulting in more accurate responses. This outcome underscores the importance of structured and comprehensive prompts in harnessing the full potential of language models like Medilora, elevating their ability to interpret complex medical scenarios accurately.

4.5.3 MedQA

Model Name	MedQA Accuracy
MedQA-adapter	46.5
Base-adapter	46.1
Meditron-7B (base, few-shot)	28.7
Meditron-7B (fine-tune)	47.9
Meditron-70B (Top-Token)	60.7

Table 5: **MedQA Results**

The MedQA evaluation showed varied model performances, ranging from 28.7 to 47.9. Medilora-base scored 46.1, indicating a consistent, moderate performance. It excelled in general medical queries but might have struggled with more complex or specialized scenarios due to limited training data or contextual understanding. This highlights the need for fine-tuning or broader training to enhance its capabilities across diverse medical contexts.

4.5.4 Comparison to Base Model

Our study compared Medilora, our base model, with Meditron-7B and OpenHermes-2.5-Mistral-7B, both widely recognized in medical benchmarks due to their extensive training on abundant data.

In our MMLU-Medical evaluations (see Table 1), Medilora-base achieved an average score of 65.4, which closely trailed OpenHermes-2.5-Mistral-7B's 65.9. Notably, Medilora exhibited considerable proficiency in subjects resembling college-level Biology (70.8) and Medicine (63.6). This strength might be attributed to its training on the USMLE dataset, offering a rich pool of examples pertinent to these domains. However, in more specialized fields such as Genetics and Virology, Medilora encountered limitations, likely due to a scarcity of diverse examples during its training.

Comparing Medilora-base to OpenHermes-2.5-Mistral-7B's performance across various medical domains, Medilora showcased slight single-digit percentage decreases in most subjects. This indicates relative competence in comprehending a diverse range of medical concepts but highlights potential areas for improvement in handling more intricate and specialized medical scenarios.

5 Analysis

In this section, we identify and discuss additional analysis and limitations regarding our methods.

5.1 Epochs and Learning Rates

In the settings of data-limited language model training, the most we can get out of a fixed set of data is 4 epochs.[18] In our experiments, we only trained on 2 epochs for each dataset. Our choice of 2 is rather arbitrary, and we acknowledge its suboptimality. We also theorize that training on 4 epochs may have let the model learn more effectively on the MedQA dataset due to its humble size.

The setting of continued pretraining is a demonstrated conventional way to update a language model, yet it was shown that hyperparameters, such as the learning rate, matters to the effectiveness of continued training. Prior work[7] suggested that using a higher starting learning rate, such as $6e-4$, would help with retraining. We used 2e-4 across all our experiments and we note that this choice is also arbitrary. We reasonably speculate that applying a larger warm-up rate during continued training may help with benchmark performance, and we leave it for future work.

5.2 Prompting

One critical difference between our experiments and other language model papers is the use of effective prompting strategies. Through the rise of ChatGPT, "prompting" has converged to the inclusive acts of optimizing input to the model and output before contact to users. Prompting techniques have evolved from reasoning elicitation techniques, such as chain-of-thought or ensemble-based methods, to methodological decoding approaches, such as world model augmentation.

Recently, novel input-output composite prompt optimization[19] combines well-established prompting methods and demonstrated that prompt optimization can boost model performance without dedicated training. Specifically, it improves few-shot learning by introducing dynamic example look-up using k-nn based similarity search. This improved GPT-4, a generalist model, on dedicated medical benchmarks and broke the record as the first autoregressive language model to score over 90 on MedQA benchmark.

Our choice of using top-k sampling and no other prompting techniques is only due to a lack of time and engineering hours. Whereas we attempt to lower the bar of improving medical language modeling by efficient fine-tuning, Nora et al. achieved equivalent effects by augmenting non-training algorithmic techniques towards inference artifacts. As the generalist paper mentions, their approach is a direct challenge to one of our project's premises: a language model needs to undergo extensive training to achieve performance improvement on task-specific benchmarks. We believe incorporating such novel techniques will only enhance the performance of our model, Medilora. It opens up a new avenue for future work, where the focus could shift from extensive training to more efficient and innovative prompting strategies, which is less demanding on proprietary-hardwares. This could potentially lead to significant improvements in the model's performance on medical benchmarks, while also reducing the computational resources and time required for training.

5.3 Embeddings

Embeddings play a crucial role in enhancing language models. They can be leveraged for similarity search and clustering, enabling the identification of queries with similar contexts. This feature can be particularly useful in the medical domain, where multiple patient queries can be retrieved based on customizable similarity scores and grouped together to provide more comprehensive and contextually relevant diagnosis or suggestions.

Moreover, embeddings can be instrumental in constructing preference datasets for feedback learning, such as Reinforcement Learning from Human Feedback (RLHF). We contemplated creating a preference dataset based on the HealthcareMagic conversation dataset and deploying Direct-Preference-Optimization training. However, due to time constraints, we were unable to establish a reliable preference labeling pipeline.

The generalist paper presented one specific strength of incorporating embeddings, where they were used question embeddings to retrieve similar queries in the training set to amplify dynamic few-shot

prompting as a non-parametric performance boost. Inspired by the empirical results of that paper, we theorize the possibility of constructing a rigorous diagnosis dataset or benchmark. This could be achieved by augmenting patient queries with similar questions based on embedding similarity and incorporating few-shot prompting techniques into the procedure. This promising direction could lead to significant improvements in the model's performance on medical benchmarks, and we leave it for future work to explore concretely.

5.4 Dataset Sizes

In this project, we acknowledge that our total data size is relatively small compared to larger lab efforts aimed at solving medical tasks. This limitation is particularly evident when comparing the training data sizes of the MedQA and PubMedQA datasets. MedQA contains only 10,718 training samples and 3.6 million tokens, significantly smaller than the PubMedQA dataset, which contains 211,000 training entries and 100 million tokens. This represents a 20-fold and 33-fold increase in the number of entries and tokens, respectively.

Our experiments demonstrated that the size of the training data significantly impacts the model's performance on the corresponding test sets, but we did not successfully conducted controlled experiments to evaluate the individual effects from each factor. For instance, the discrepancy between our model's PubMedQA and MedQA accuracy could either be attributed to the smaller size of the MedQA training data or that some of the benchmark data were present in the training dataset. Future work could involve more rigorous statistical analysis of these datasets to better understand their characteristics and how they might impact model performance.

Additionally, while compute-optimal scaling laws suggest that model performance should improve with increased training data, our experiments did not systematically control for this variable. Future experiments could explore this relationship more thoroughly. For example, one could extract a smaller subset of the PubMedQA training set that maintains the same distribution as the main training set. This approach could provide insights into the model's generalization capabilities and the effects of training data size on model performance.

6 Safety Disclaimer

As with all language models, please treat the model output with a grain of salt and always remain critical. Don't use it for medical advice yet.

7 Conclusion

Fine-tuning large language models is excessively costly due to the expenses associated with hosting independent instances for various tasks. In this work, we finetune and evaluate the performance of quantized low-rank adaptation of state-of-the-art open-source models on curated medical datasets and compare it with fully trained medical LLMs. We show that our 7B model outperforms Meditron-7B for medical short answering on PubmedQA. Moreover, we achieve comparable performance on MedQA.

Our current work demonstrates the potential of efficient fine-tuning of large language models (LLMs) for medical tasks. Prospective enhancements could involve additional training on more extensive datasets and integrating Reinforcement Learning from Human Feedback (RLHF) for downstream tasks.

References

[1] W. T. Alfred O. Berg, David Atkins. Clinical practice guidelines in practice and education. *Journal of General Internal Medicine*, 1997.

[2] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer, 2020.

[3] Z. Chen, A. H. Cano, A. Romanou, A. Bonnet, K. Matoba, F. Salvi, M. Pagliardini, S. Fan, A. Köpf, A. Mohtashami, A. Sallinen, A. Sakhaeirad, V. Swamy, I. Krawczuk, D. Bayazit, A. Marmet, S. Montariol, M.-A. Hartley, M. Jaggi, and A. Bosselut. Meditron-70b: Scaling medical pretraining for large language models, 2023.

[4] T. Dettmers et al. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[5] L. Gao, J. Tow, B. Abbasi, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, A. Le Noac'h, H. Li, K. McDonell, N. Muennighoff, C. Ociepa, J. Phang, L. Reynolds, H. Schoelkopf, A. Skowron, L. Sutawika, E. Tang, A. Thite, B. Wang, K. Wang, and A. Zou. A framework for few-shot language model evaluation, 12 2023. URL `https://zenodo.org/records/10256836`.

[6] K. Gupta, B. Th'erien, A. Ibrahim, M. L. Richter, Q. G. Anthony, E. Belilovsky, I. Rish, and T. Lesort. Continual pre-training of large language models: How to (re)warm your model? *ArXiv*, abs/2308.04014, 2023. URL `https://api.semanticscholar.org/CorpusID:260704601`.

[7] K. Gupta, B. Thérien, A. Ibrahim, M. L. Richter, Q. Anthony, E. Belilovsky, I. Rish, and T. Lesort. Continual pre-training of large language models: How to (re)warm your model?, 2023.

[8] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL `https://aclanthology.org/2020.acl-main.740`.

[9] T. Han, L. C. Adams, J.-M. Papaioannou, P. Grundmann, T. Oberhauser, A. Löser, D. Truhn, and K. K. Bressem. Medalpaca – an open-source collection of medical conversational ai models and training data, 2023.

[10] S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, and Z. Hu. Reasoning with language model is planning with world model, 2023.

[11] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding, 2021.

[12] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre. Training compute-optimal large language models, 2022.

[13] Introducing ChatGPT, 2022. URL `https://openai.com/blog/chatgpt`.

[14] A. Q. Jiang et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[15] Q. Jin et al. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*, 2019.

[16] Y. Li, Z. Li, K. Zhang, R. Dan, S. Jiang, and Y. Zhang. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge, 2023.

[17] A. Mitra, L. D. Corro, S. Mahajan, A. Codas, C. Simoes, S. Agrawal, X. Chen, A. Razdaibiedina, E. Jones, K. Aggarwal, H. Palangi, G. Zheng, C. Rosset, H. Khanpour, and A. Awadallah. Orca 2: Teaching small language models how to reason, 2023.

[18] N. Muennighoff, A. M. Rush, B. Barak, T. L. Scao, A. Piktus, N. Tazi, S. Pyysalo, T. Wolf, and C. Raffel. Scaling data-constrained language models, 2023.

[19] H. Nori, Y. T. Lee, S. Zhang, D. Carignan, R. Edgar, N. Fusi, N. King, J. Larson, Y. Li, W. Liu, R. Luo, S. M. McKinney, R. O. Ness, H. Poon, T. Qin, N. Usuyama, C. White, and E. Horvitz. Can generalist foundation models outcompete special-purpose tuning? case study in medicine, 2023.

[20] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022.

[21] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, P. Payne, M. Seneviratne, P. Gamble, C. Kelly, N. Scharli, A. Chowdhery, P. Mansfield, B. A. y Arcas, D. Webster, G. S. Corrado, Y. Matias, K. Chou, J. Gottweis, N. Tomasev, Y. Liu, A. Rajkomar, J. Barral, C. Semturs, A. Karthikesalingam, and V. Natarajan. Large language models encode clinical knowledge, 2022.

[22] A. Srivastava et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2023.

[23] Teknium. Openhermes. Technical report, Nous Research, 2023. URL `https://huggingface.co/datasets/teknium/openhermes`.

[24] A. Toma, P. R. Lawler, J. Ba, R. G. Krishnan, B. B. Rubin, and B. Wang. Clinical camel: An open expert-level medical language model with dialogue-based knowledge encoding, 2023.

[25] H. Touvron et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[26] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Annual Meeting of the Association for Computational Linguistics*, 2022.

[27] C. Wu, W. Lin, X. Zhang, Y. Zhang, Y. Wang, and W. Xie. Pmc-llama: Towards building open-source language models for medicine, 2023.

[28] X. Zhang et al. Medical exam question answering with large-scale reading comprehension. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

8 Supplementary Material

8.1 Training Losses

We observed the training loss curves for our model across all datasets. Figure 1 shows the continued training loss and Figure 2 shows the loss on two benchmark fine-tune datasets.

8.1.1 Continued Training Loss

Figure 1 shows that the textbook and MIMIC-III datasets exhibited a sharp initial decline in loss, swiftly transitioning to a plateau, indicative of the model's rapid learning and subsequent convergence to an optimal state. This pattern aligns with expectations for a model effectively assimilating and fine-tuning knowledge from the data provided. Conversely, the training on guideline data demonstrated a persistent loss value, suggesting a potential plateau at a relatively early stage. This could imply a potential mismatch between model capacity and the complexity of the guideline data, or it might indicate a need for hyperparameter optimization. Further investigation into model architecture adjustments, data augmentation, or regularization strategies may be required to enhance learning efficacy for the guideline dataset.

8.1.2 Fine-tuning Loss

Figure 2 shows the loss curve for both the PubMedQA Instruction Tune dataset and the MedQA dataset. For PubMedQA, we observed a rapid decline followed by a stable, low loss, indicating quick learning and effective convergence, suggesting dataset-model compatibility. In contrast, the MedQA dataset fine-tuning run, with the name 'feasible-rain-92', exhibits a steady decline in loss without plateauing. It could imply ongoing learning and the presence of novel or complex patterns not captured in pretraining, or simply show the limited size of the MedQA training data. These observations suggest that while PubMedQA fine-tuning is sufficient, MedQA may either benefit from exploited extended training, or some other techniques to augment the model performance.

8.2 ChatML Conversation Samples

For both PubMedQA and MedQA training set, we format each entry into lists of conversations. Figure 3 shows a sample ChatML formatted PubMedQA entry, in which the context and questions are concatenated together as input, and the ground truth label and long answers are concatenated as target output.