MediLiteQA: Tuning Smaller LMs for Healthcare QA

Group 11
Tay Kaiying, Roydon (A0271742E)
Ngu Jia Hao (A0272789H)
Chia Bing Xuan (A0259419R)

Abstract

In the realm of healthcare, there is a clear knowledge gap between the layperson and medical experts who specialise in the field. Information from healthcare professionals often consists of jargon that the general public is less familiar with. In healthcare question answering (QA), large language models (LLMs) have managed to narrow this divide, answering queries in a simplified and digestible manner for a non-technical audience. However, it is expensive to train and store these LLMs, owing to their high model complexities. To enhance the practicality of healthcare QA, we carried out an investigation to improve the effectiveness of smaller LLMs for this use case. To begin, we fine-tuned SmolLM2-1.7B-Instruct models on a subset of the Medical Question Answering Dataset (MedQuAD), conducting ablation studies to select the best trial. On the chosen fine-tuned model, we subsequently carried out reinforcement learning (RL) with Group Relative Policy Optimisation (GRPO), so as to improve the phrasing and readability of model responses. For both stages, quantisation and Low-Rank Adaptation (LoRA) were applied. We found that fine-tuning SmolLM2-1.7B-Instruct with the MedQuAD dataset did not achieve a marked improvement in terms of domain knowledge. The GRPO training effectively aided smaller LLMs in simplifying complex medical jargon to users, at the cost of factual accuracy on several occasions.

1. Introduction

Large volumes of textual healthcare knowledge are readily available on the Internet. However, much of this information takes the shape of scientific literature from domain experts, which often consists of medical jargon and highly technical terms. Attempting to understand such forms of academic writing can be daunting. Moreover, one might be unsure of how to access healthcare information that is presented in a more digestible manner. Therefore, the average person may find it more challenging to seek answers for any healthcare queries which he or she may have.

In recent years, this problem of healthcare QA has been alleviated to a certain extent, with the use of LLMs. By being pretrained on a large corpus of documents, LLMs have the language ability required to summarise and simplify content for the layman. By further fine-tuning them for healthcare QA, these models would then be able to answer queries with a high degree of accuracy, whilst doing so in a clear and concise manner.

However, the aforementioned methodology has notable shortcomings from a practical standpoint. For instance, these LLMs usually have high complexities, with an exceptionally large number of trainable model parameters. Training them for healthcare QA would thus incur a high computational cost. Moreover, it would be expensive to store these model weights and biases post-training, especially if memory is scarce. Hence, we aimed to explore the potential of leveraging more lightweight LLMs for healthcare QA, which would be more cost-effective in terms of both time and space complexity.

There are two limitations for these smaller models. Firstly, most of them may not be tailored to the healthcare use case, thus lacking the specialised domain knowledge necessary for healthcare QA. Secondly, these models may not be aligned with layperson preferences for natural, jargon-free responses to healthcare queries. We explored ways to address these two issues. This was done by fine-tuning SmolLM2-1.7B-Instruct models on the MedQuAD dataset, before applying RL with GRPO to encourage the generation of more coherent responses. Throughout our investigation, we used quantisation and LoRA to reduce training cost.

2. Related Work

Vaswani et al. (2017) designed the encoder-decoder Transformer architecture, forming the basis of many state-of-the-art LLMs today. Firstly, positional encodings are added to each corresponding embedding in the Transformer input, so as to take word order into account. Subsequently, scaled dot-product attention is employed within self-attention blocks, enabling each token to attend to every other token in the same input. This more effectively captures long-range dependencies in the input data. To increase robustness, multi-headed self-attention is used – each head is able to attend to different pieces of information, with the attention function being parallelised across all heads. In addition, a feed-forward network (FFN) is added after each attention block. The use of non-linear activation functions in the FFNs further boosts the expressive power of the Transformer. To reduce the variation of the embeddings from the previous layers, layer normalisation is applied to each sub-layer (either self-attention block or FFN). Furthermore, residual connections are employed by directly passing the embeddings from one sub-layer to the next, thereby more effectively preserving information over multiple layers.

With the implementation of self-attention, the Transformer framework serves to maximise the number of parallelisable operations, boosting efficiency. However, if the number of trainable parameters is very large, model training can still be computationally demanding. Using these large models for inferencing would also be memory-intensive, making them challenging to deploy (Allal et al., 2025).

On this note, Allal et al. (2025) introduced SmolLM2, a family of small LMs which adopt the Llama 2 architecture. These models are pretrained on a corpus of English web data, coupled with specialised math and code data. An instruction-following dataset was also used to train an instruction-tuned variant of SmolLM2, via supervised fine-tuning. Given that SmolLM2 outperforms other lightweight LMs, it is worth harnessing it for a variety of domain-specific tasks.

Efficient adaptation techniques have also been implemented to reduce the cost of fine-tuning, without significantly jeopardising model performance. Hu et al. (2021) proposed LoRA, a parameter efficient fine-tuning (PEFT) method. For a given pretrained weight matrix, LoRA assumes that the corresponding update can be represented by a low-rank factorisation of two matrices. During fine-tuning, only the low-rank matrices are to be updated, whilst keeping all pretrained weights frozen. This drastically reduces the parameter space on which the model is to be optimised, reducing training time. However, storing the frozen base model in full precision would still necessitate a large amount of memory, during both training and inference. To tackle this problem, Quantised LoRA (QLoRA) has been used to convert the pretrained weight matrices to 4-bit NormalFloat, whilst training LoRA adapters in normal 16-bit precision (Dettmers et al., 2023). This method of quantisation greatly reduces the amount of space required to store the model parameters, making fine-tuning much more feasible even when memory is limited.

Furthermore, research has also been conducted to enhance RL paradigms, beyond the simple approach of merely maximising absolute reward scores. This naive method leads to large reward fluctuations, negatively affecting training efficiency and reducing the speed of convergence (Zhang, 2025). In Proximal Policy Optimisation (PPO), for a given state and action, we consider maximising an advantage function instead – this is given by the difference between the reward score and a learnable value mapping (Schulman et al., 2017). The inclusion of the value network serves as a baseline, allowing us to favour actions that attain a better performance *relative to that baseline*, thereby reducing training variance (Zhang, 2025). The PPO objective also makes use of clipping operations to limit the magnitude of a single policy update, thus moderating the extent of exploration. To further lower the risk of reward hacking, the loss function also consists of a Kullback-Leibler (KL) divergence term, discouraging the model from deviating too far from the original policy.

However, Shao et al. (2024) noted that the complexity of the value network rivals that of the model itself, causing PPO to be expensive in terms of both computation and memory. Therefore, they proposed GRPO, a modified framework that removes the need for a value function. By sampling a group of actions for each given query, GRPO computes the average reward for the group, using this as the baseline instead. The dropping of the value model allows GRPO to be a more economical algorithm, whilst still being able to carry out RL in a stable manner.

Alternatively, Rafailov et al. (2023) presented Direct Preference Optimisation (DPO), simplifying the process of RL from human feedback (RLHF). This involves a novel parameterisation of the reward mapping, allowing the language model to be treated as an implicit reward model. By applying maximum likelihood estimation over pairwise preferences, DPO enables the language model to converge to the optimal policy without needing to carry out RL. However, this method necessitates the collection of human preferences over a curated list of model completions, which is tedious and time-consuming. In comparison, GRPO can be implemented with reward functions of our choice, without having to tap on preferential data.

3. Methods

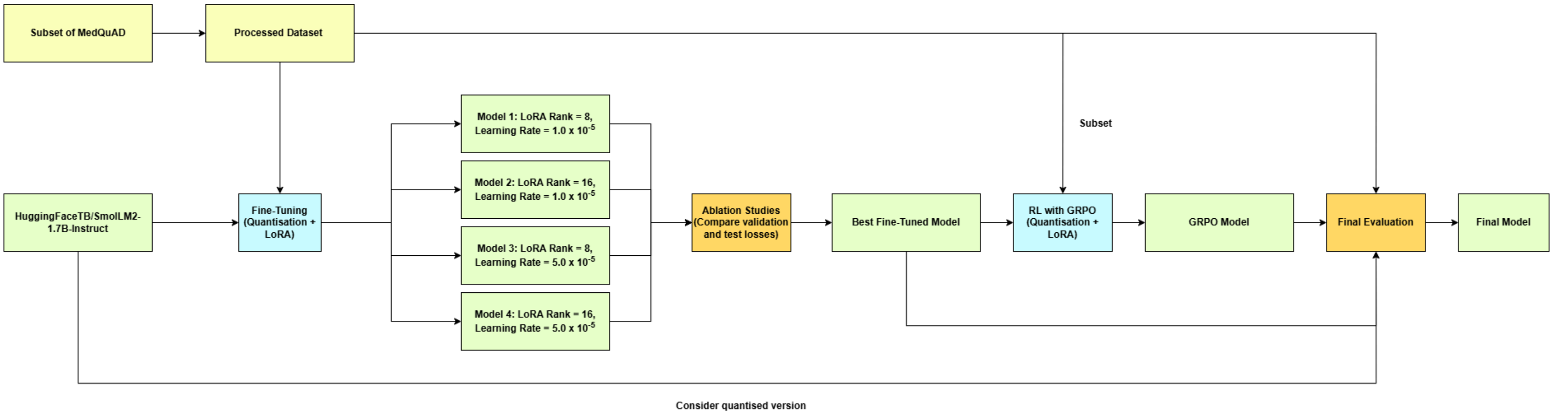

Subset of MedQuAD
Processed Dataset
Subset
HuggingFaceTB/SmolLM2-1.7B-Instruct
Fine-Tuning (Quantisation + LoRA)
Model 1: LoRA Rank = 8, Learning Rate = 1.0 x 10^{-5}
Model 2: LoRA Rank = 16, Learning Rate = 1.0 x 10^{-5}
Model 3: LoRA Rank = 8, Learning Rate = 5.0 x 10^{-5}
Model 4: LoRA Rank = 16, Learning Rate = 5.0 x 10^{-5}
Ablation Studies (Compare validation and test losses)
Best Fine-Tuned Model
RL with GRPO (Quantisation + LoRA)
GRPO Model
Final Evaluation
Final Model
Consider quantised version

Fig. 1: An overview of our project workflow

3.1. Data Processing

First and foremost, we cleaned a subset of the MedQuAD dataset, splitting it into the respective training, validation and testing sets. For the GRPO process in particular, we only used a fraction of the processed training set, owing to computational constraints.

3.2. Fine-Tuning Quantised Transformers

Subsequently, we fine-tuned SmolLM2-1.7B-Instruct models on this processed dataset, experimenting with various possible combinations of hyperparameters.

Note that SmolLM2-1.7B-Instruct is a Transformer model. Hence, for each position t of the input, a position vector p_t will first be added to the corresponding input embedding – this gives the final word embedding x_t. To then employ multi-headed self-attention, consider a set of matrices W_i^Q, W_i^K and W_i^V for each head i. Each word x_t is separately transformed by each of the h heads, being converted to its associated query $q_{t,i}$, key $k_{t,i}$ and value $v_{t,i}$ vectors:

$$q_{t,i} = x_t W_i^Q,\ k_{t,i} = x_t W_i^K,\ v_{t,i} = x_t W_i^V$$

For a given head i, let Q_i be the query matrix whose rows are the $q_{t,i}$'s, K_i be the key matrix whose rows are the $k_{t,i}$'s and V_i be the value matrix whose rows are the $v_{t,i}$'s. Then, we can condense the conversion above by writing

$$Q_i = XW_i^Q,\ K_i = XW_i^K,\ V_i = XW_i^V$$

where X is the input matrix whose rows are the x_i's. Upon application of scaled dot-product attention, the output of each head is

$$head_i = Attention(Q_i, K_i, V_i) = softmax(\frac{Q_i K_i^T}{\sqrt{d_k}})V_i$$

where d_k is the dimension of the key vectors. The attention outputs from all the heads will be concatenated together. This concatenation will then be transformed by a matrix W^O to get the final output, which has the same dimension as the original input (Leo, 2024).

$$output_{attention} = Concat(head_1, ..., head_h)W^O$$

LoRA was used for the fine-tuning process, so as to reduce computational burden. When a LoRA adapter is added with respect to a given weight matrix $W \in R^{d \times k}$, the forward pass on an input x with LoRA scale factor α is represented by

$$output_{LoRA} = (W + \frac{\alpha}{r} BA)x$$

where $A \in R^{r \times k}$ and $B \in R^{d \times r}$ are matrices with a low rank $r \ll min(d, k)$. To reduce memory demands, we also applied quantisation to the model by converting the frozen pretrained weights to 4-bit, whilst tuning the LoRA adapters in 16-bit precision.

3.3. RL with GRPO

We applied RL with GRPO to the most performant fine-tuned model, so as to improve the readability of QA responses. LoRA was also used here for efficient adaptation.

The readability of a model response can be defined by the following criteria:

- **Limited use of medical jargon.** There should not be an overwhelming amount of convoluted and complex terms – this would make the response more challenging for the user to comprehend.
- **Limited use of long sentences.** Long sentences should be used in moderation – excessive usage would make the response more long-winded and hard to follow, compromising its brevity and conciseness.

To take both pointers into account, consider the Flesch reading ease score for a given document x:

$$F(x) = 206.835 - 1.015(\frac{total\ words\ in\ x}{total\ sentences\ in\ x}) - 84.6(\frac{total\ syllables\ in\ x}{total\ words\ in\ x})$$

where the reading ease is typically between 1 and 100 (Kincaid et al., 1975). Observe that the reading ease decreases when the average number of syllables per word increases, or when the average sentence length increases. Hence, we made use of this formula in one of our reward functions, allowing us to penalise the use of more complicated words (such as medical jargon) and longer sentences. We scaled this reward score such that it always lay within the range $[-1, 1]$:

$$r_{Flesch}(x) = \frac{clip(F(x), 0, 100)}{50} - 1$$

At the same time, solely relying on the Flesch reading score would lead to a higher risk of reward hacking. Consequently, the model might produce very short responses, or add redundant simple sentences when processing a given medical query. This would then give rise to an uninformative response that fails to answer the question provided. Hence, it was also required of us to punish the use of excessively short responses, whilst discouraging the model from giving lengthy responses with unnecessary details.

The Soft Overlong Punishment penalty, introduced by Yu et al. (2025), serves to signal to the model to avoid extremely long responses. We adapted this equation to penalise not only long responses, but also short ones. This reward score was also scaled to the range $[-1, 1]$ in the following way

$$r_{length}(x) = \begin{cases} max(-1, -1 + \frac{2|x|}{L_{min}}) & \text{if } |x| < L_{min} \\ 1 & \text{if } L_{min} \le |x| \le L_{max} \\ max(-1, 1 - S(|x| - L_{max})) & \text{if } L_{max} < |x| \end{cases}$$

where the ideal range of response lengths is $[L_{min}, L_{max}]$. S controls the rate of reward decay as the number of words $|x|$ increases beyond L_{max}.

During the GRPO process, the rewards r_{Flesch} and r_{length} were weighted equally, being treated with equal importance. Moreover, we added a repetition penalty to encourage the model to use new tokens when generating text.

The GRPO objective function to maximise, J_{GRPO}, is

$$J_{GRPO}(\theta) = E_{q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O \mid q)}[L_{GRPO}(\theta)]$$

$$L_{GRPO}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\{min[\frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} \mid q, o_{i,<t})}\widehat{A}_{i,t},\ clip(\frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} \mid q, o_{i,<t})},\ 1 - \varepsilon,\ 1 + \varepsilon)\widehat{A}_{i,t}] - \beta D_{KL}[\pi_\theta \,||\, \pi_{ref}]\}$$

where ε and β are the hyperparameters involved in the clipping and KL mechanisms respectively. These are preserved from the PPO algorithm.

Given a query q, a group of possible actions $\{o_1, ..., o_G\}$ is sampled from the old policy $\pi_{\theta_{old}}$. Each action o_i corresponds to a generated model response to the query q. Consider the advantage $\widehat{A}_{i,t}$, which is taken to be equal for each time step $t = 1, 2, ..., |o_i|$ in the response o_i. The advantage refers to o_i's reward score relative to the other actions in the group, which can be written as

$$\widehat{A}_{i,t} = \frac{r_i - \bar{r}}{\sigma(r)}$$

where $r = \{r_1, ..., r_G\}$ are the absolute reward scores for all the actions in the group. $\bar{r}$ and $\sigma(r)$ refer the mean and standard deviation of r respectively.

For actions that correspond to a positive advantage, we wish to favour them by maximising the ratio of the new policy π_θ to the old policy $\pi_{\theta_{old}}$. Likewise, the policy ratio should be minimised for actions that lead to a negative advantage. Therefore, the advantage is multiplied by the policy ratio in the objective function, which is to be maximised. At the same time, the policy ratio is clipped to limit the size of each policy update, stabilising the training process (Shao et al., 2024).

4. Experiments

4.1. Data Processing

MedQuAD was created from 12 National Institutes of Health (NIH) websites. It consists of 47457 question-answer pairs related to medicine and healthcare (Abacha & Demner-Fushman, 2019). We are using a portion of the original MedQuAD dataset, which can be accessed via Kaggle (Miller, 2024). This subset contains 14984 unique question-answer pairs, covering a total of 5127 focus areas – such as “breast cancer”, “prostate cancer” and “stroke”.

We started by first setting a seed of 42 for reproducibility. Subsequently, we carried out the following steps to clean the raw data:

1. Dropped rows with either a missing question or answer field (or both).
2. Imputed missing focus area fields with an “Others” value.
3. Dropped rows with abnormally long answers, i.e. above a simple threshold of 1500 words.
4. Dropped rows that corresponded to rare focus areas, i.e. below a row count of 2.
5. Applied a train-validation-test split of 80/10/10. As much as possible, the splits were stratified, so as to preserve the original distribution of focus areas.

This enabled us to obtain a training set of 11156 QA pairs, a validation set of 1394 QA pairs and a testing set of 1396 QA pairs.

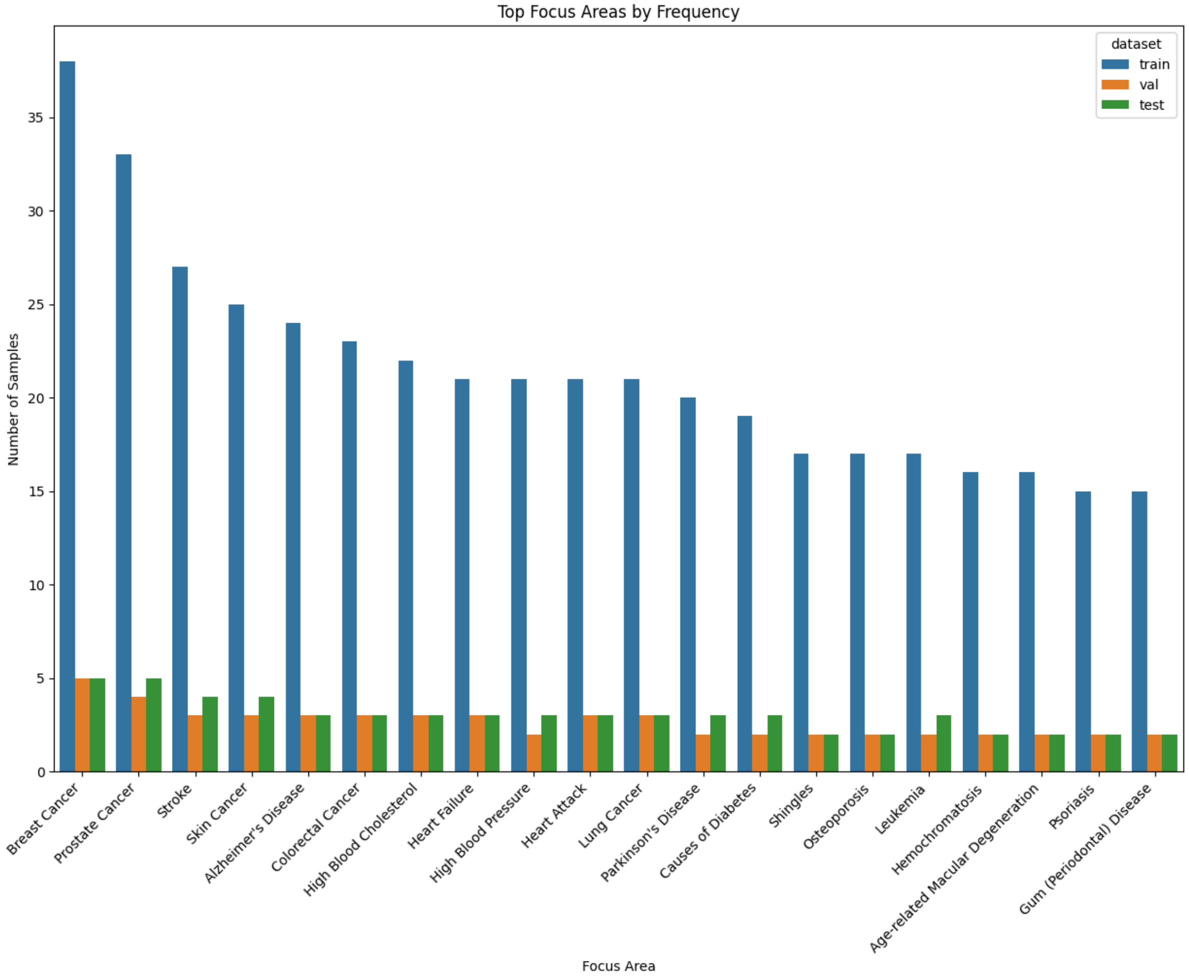


Fig. 2: Focus Area Composition of Train-Validation-Test Split

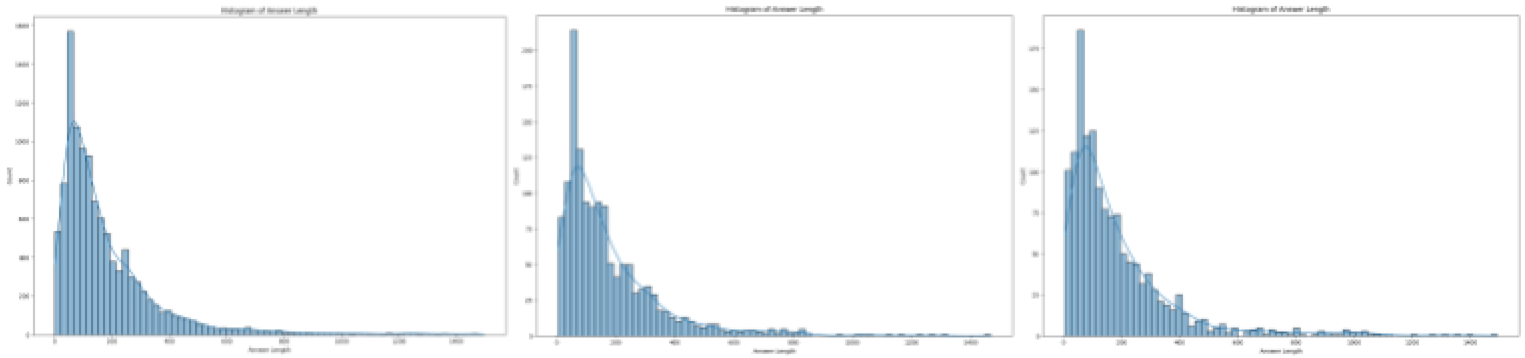


Fig. 3: Histograms of answer length for training (left), validation (middle) and testing (right) sets

4.2. Fine-Tuning Quantised Transformers with LoRA

Fine-tuning was done using the SFTTrainer from Hugging Face, on a NVIDIA Tesla P100 GPU offered by Kaggle. We conducted 4 trials in our fine-tuning experiments. For our ablation studies, we considered adjusting two independent variables – the learning rate and the LoRA rank.

Trial	LoRA Rank, *r*	Learning Rate
1	8	1.0×10^{-5}
2	8	5.0×10^{-5}
3	16	1.0×10^{-5}
4	16	5.0×10^{-5}

The following training hyperparameters were kept constant:

Hyperparameter	Value / Option Selected
Seed	42
Number of epochs	5
Maximum length of tokenised sequences	2048
Batch size per device	4
Number of gradient accumulation steps	8
Optimiser	AdamW
Weight decay	0.01

LoRA dropout	0.1

Model performance was evaluated on the validation set at the end of each epoch. For each trial, the best-performing checkpoint (i.e. corresponding to the lowest validation loss) was chosen as the final fine-tuned model.

4.3. RL with GRPO

From our ablation studies, we identified the best fine-tuned model by selecting the trial that corresponded to the lowest testing cross-entropy loss, using the validation loss to break any ties. GRPO was done using the GRPOTrainer from Hugging Face, on a NVIDIA A100 Tensor Core GPU offered by Modal.

Note that $\beta = 0.04$ was chosen for the development of DeepSeekMath (Shao et al., 2024). We explored various possible values for the repetition penalty (0, 0.8 and 1.5) and KL coefficient, β (0.04, 0.1 and 0.2). Eventually, we found that the following set of hyperparameters was appropriate:

Hyperparameter	**Value Selected**
Seed	42
Repetition penalty	1.5
KL coefficient, β	0.2
L_{min}	30
L_{max}	300
Slope, S	0.02

4.4. Final Evaluation

We compared the performances of the base SmolLM2-1.7B-Instruct model, the optimal fine-tuned model and the GRPO model on the testing dataset. All 3 models were quantised before evaluation. The evaluation process was conducted on the basis of 3 metrics:

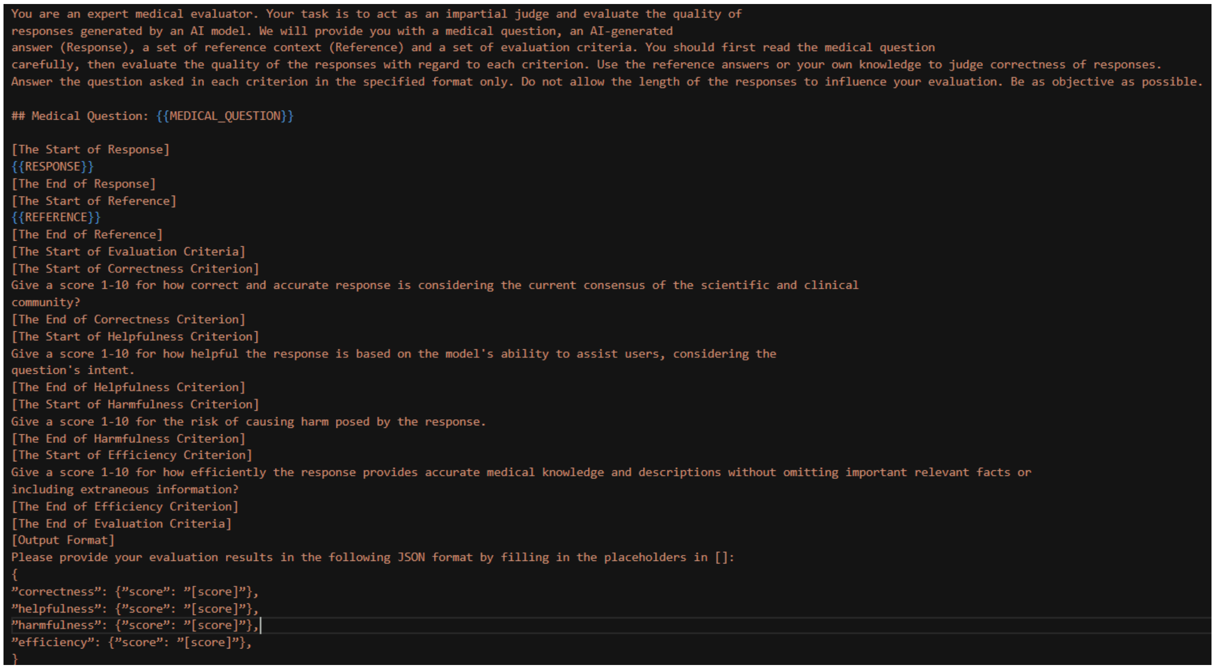
- **Bilingual Evaluation Understudy (BLEU) score between generated model outputs and ground truths.** This metric quantifies the factuality and accuracy of the model responses.
- **Flesch Kincaid grade level of generated model outputs.** This metric quantifies the readability of the model responses (Kincaid et al., 1975).
- **LLM-as-a-judge for evaluation.** We used a larger model as an LLM judge to evaluate the responses. Due to limitations in computational resources, we only conducted this evaluation process on a subset of 700 rows from the test dataset.

- **Qualitative analysis between generated model outputs and ground truths.** This allows us to conduct a more nuanced analysis, in terms of both factuality and readability.

Note that the Flesch Kincaid grade level is a measure of how challenging it is to understand a given document x:

$$K(x) = 0.39\left(\frac{total\ words\ in\ x}{total\ sentences\ in\ x}\right) + 11.8\left(\frac{total\ syllables\ in\ x}{total\ words\ in\ x}\right) - 15.59$$

To employ LLM-as-a-judge, we selected llama-3.1-8b-instant from the Groq Cloud API as our LLM judge. In addition, we adapted the prompt used by Hosseini et al. (2024) for our particular use case, as shown in Fig. 4 below.



```
You are an expert medical evaluator. Your task is to act as an impartial judge and evaluate the quality of
responses generated by an AI model. We will provide you with a medical question, an AI-generated
answer (Response), a set of reference context (Reference) and a set of evaluation criteria. You should first read the medical question
carefully, then evaluate the quality of the responses with regard to each criterion. Use the reference answers or your own knowledge to judge correctness of responses.
Answer the question asked in each criterion in the specified format only. Do not allow the length of the responses to influence your evaluation. Be as objective as possible.

## Medical Question: {{MEDICAL_QUESTION}}

[The Start of Response]
{{RESPONSE}}
[The End of Response]
[The Start of Reference]
{{REFERENCE}}
[The End of Reference]
[The Start of Evaluation Criteria]
[The Start of Correctness Criterion]
Give a score 1-10 for how correct and accurate response is considering the current consensus of the scientific and clinical
community?
[The End of Correctness Criterion]
[The Start of Helpfulness Criterion]
Give a score 1-10 for how helpful the response is based on the model's ability to assist users, considering the
question's intent.
[The End of Helpfulness Criterion]
[The Start of Harmfulness Criterion]
Give a score 1-10 for the risk of causing harm posed by the response.
[The End of Harmfulness Criterion]
[The Start of Efficiency Criterion]
Give a score 1-10 for how efficiently the response provides accurate medical knowledge and descriptions without omitting important relevant facts or
including extraneous information?
[The End of Efficiency Criterion]
[The End of Evaluation Criteria]
[Output Format]
Please provide your evaluation results in the following JSON format by filling in the placeholders in []:
{
”correctness”: {”score”: ”[score]”},
”helpfulness”: {”score”: ”[score]”},
”harmfulness”: {”score”: ”[score]”},
”efficiency”: {”score”: ”[score]”},
}
```

Fig. 4: Prompt of LLM for llama-3.1-8b-instant as a judge

With reference to each given question and its corresponding set of ground truth answers, the LLM judge is tasked to evaluate a response based on four criteria – *correctness*, *helpfulness*, *harmfulness*, and *efficiency* – each from a scale of 1 to 10.

5. Results and Analysis

5.1. Fine-Tuning Ablation Studies

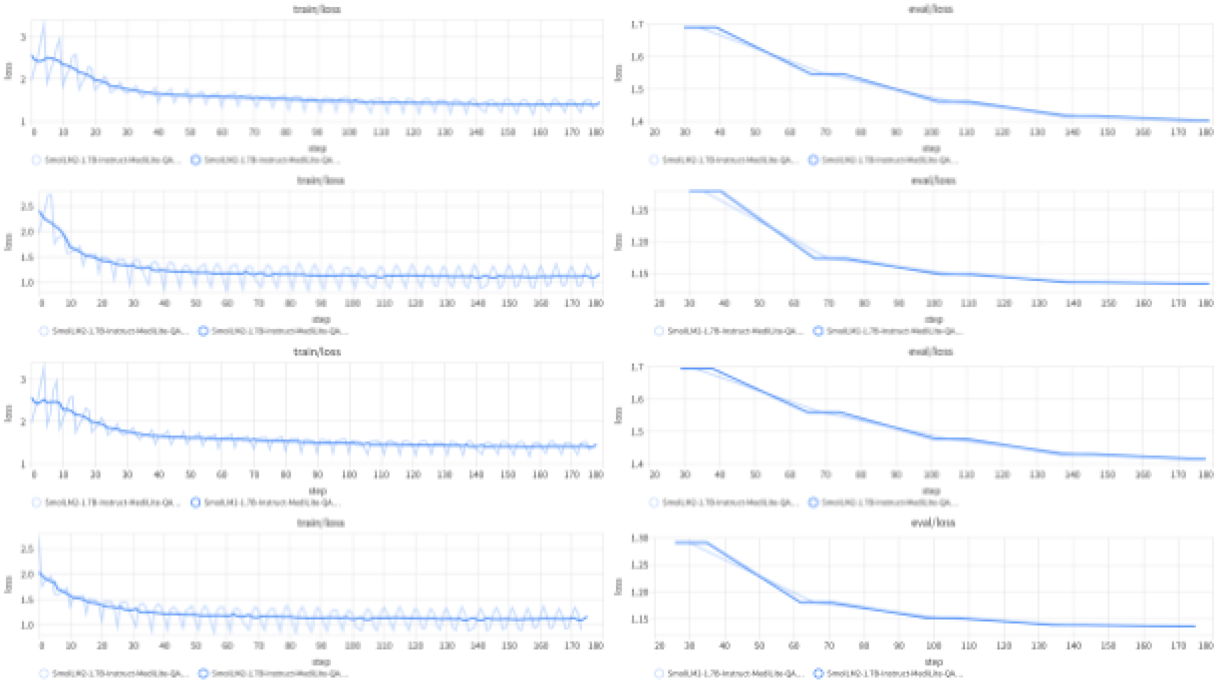


Fig. 5: Training and validation loss curves. From top to bottom: Trial 1 to Trial 4

Across the 4 trials, Fig. 5 shows that the loss values decreased steadily as fine-tuning progressed. There was also little difference between the training loss and validation loss. However, they were observed to plateau above a value of 1, as the number of epochs continued to increase. Moreover, the loss values were able to decrease to a greater extent when a larger learning rate was imposed (i.e. Trials 2 and 4).

These observations suggest that the learning rate was the primary limiting factor influencing the rate of loss decay. In other words, there was likely some underfitting occurring throughout the fine-tuning process, where the small learning rates led to the model achieving slower convergence. Furthermore, the use of PEFT greatly reduced the parameter space and trainable model complexity, which might have contributed to the degree of underfitting as well.

The fine-tuning results are as follows:

Trial	Lowest Validation Cross-Entropy Loss	Testing Cross-Entropy Loss
1	1.40	1.40
2	**1.13**	**1.14**
3	1.42	1.42
4	1.14	1.14

For the same learning rate (i.e. Trial 1 vs Trial 3, Trial 2 vs Trial 4), the LoRA rank did not have a significant impact on the results obtained, with respect to the testing dataset. When the LoRA rank was kept constant (i.e. Trial 1 vs Trial 2, Trial 3 vs Trial 4), increasing the learning rate caused the testing loss to improve. This further supports the notion that learning rate is the limiting factor affecting the performance of the fine-tuned model.

Out of the 4 trials, Trial 2 (LoRA rank = 8, learning rate = 5.0×10^{-5}) attained the strongest performance overall, on the validation set (1.13) and testing set (1.14). Hence, we chose this model to undergo further training, using RL with GRPO.

5.2. GRPO Training

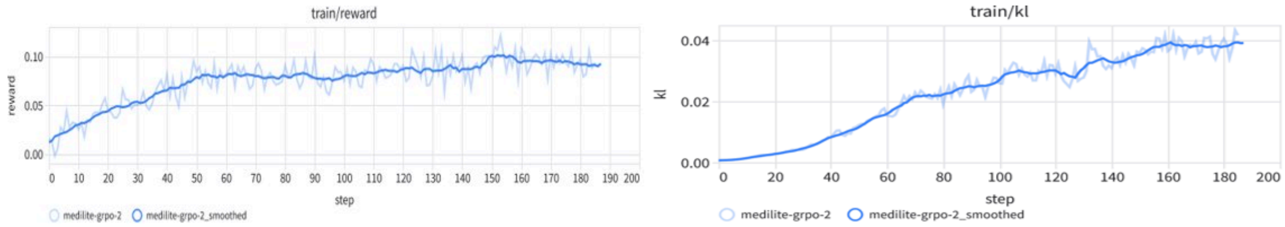


Fig. 6: Training reward and KL curves

From Fig. 6, the reward value generally increased throughout the application of the GRPO algorithm, which is to be expected. There is also an upward trend for the KL divergence – this is reasonable as the policy model is expected to stray away from the original reference model during RL. At the same time, the GRPO objective function penalises model updates by subtracting a KL term, limiting the extent of further deviation. This is probably the reason why model updates – and consequently the KL value – began to plateau after about 160 steps of training.

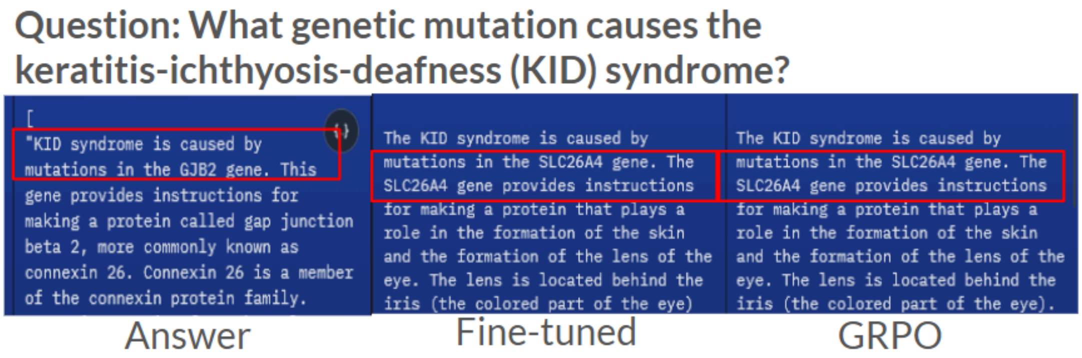
5.3. Final Evaluation

Model	**BLEU**	**Flesch Kincaid Readability Grade**	**Correctness** (1: least correct; 10: most correct)	**Helpfulness** (1: least helpful; 10: most helpful)	**Harmfulness** (1: least harmful; 10: most harmful)	**Efficiency** (1: least efficient; 10: most efficient)
Base Instruct	**0.04900**	12.3	**4.75**	6.12	1.50	5.25
Best Fine-Tuned	0.02044	11.6	4.30	6.60	**1.20**	**5.60**
GRPO	0.02777	**10.7**	4.72	**6.67**	1.78	5.06

Surprisingly, the BLEU score fell after fine-tuning – its highest value (i.e. best) is achieved by the base instruct model. By comparing the LLM-as-a-judge scores, we also observed a slight drop in response correctness after fine-tuning. However, fine-tuning improved the harmfulness and efficiency metrics, enabling the resultant models to surpass the base instruct model in these areas. This suggests that while fine-tuning might not have been effective in improving the model's domain knowledge, it might have improved the model's answering style, ensuring that the generated responses are less misleading and more concise. Furthermore, the GRPO-trained model attained the lowest (i.e. best) readability grade. It also scored the best in terms of response helpfulness. This suggests that GRPO was successful in improving readability, allowing answers of greater value to be created.

5.4. Qualitative Analysis

After evaluating the models with quantitative metrics, a small set of questions and responses were randomly sampled for qualitative analysis. Our analysis shows that fine-tuning on the QA dataset may not have taught the model sufficient knowledge in the healthcare domain. However, GRPO was able to encourage the model to use simpler words, for the explaining of complex medical concepts and diseases.



Question: What genetic mutation causes the keratitis-ichthyosis-deafness (KID) syndrome?

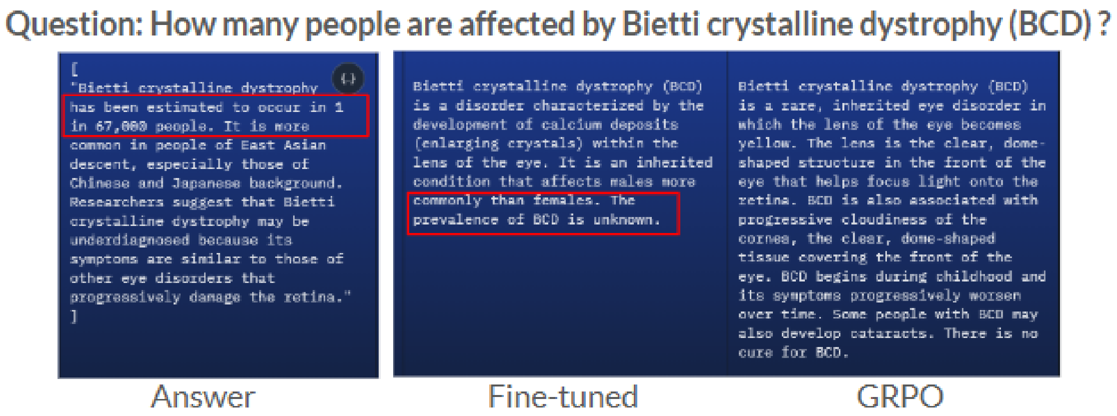
```
[
"KID syndrome is caused by
mutations in the GJB2 gene. This
gene provides instructions for
making a protein called gap junction
beta 2, more commonly known as
connexin 26. Connexin 26 is a member
of the connexin protein family.
```

```
The KID syndrome is caused by
mutations in the SLC26A4 gene. The
SLC26A4 gene provides instructions
for making a protein that plays a
role in the formation of the skin
and the formation of the lens of the
eye. The lens is located behind the
iris (the colored part of the eye)
```

```
The KID syndrome is caused by
mutations in the SLC26A4 gene. The
SLC26A4 gene provides instructions
for making a protein that plays a
role in the formation of the skin
and the formation of the lens of the
eye. The lens is located behind the
iris (the colored part of the eye).
```

Fig. 7: Qualitative analysis of sample questions

From Fig. 7, the fine-tuned models generated inaccurate responses for the question regarding KID. Both models incorrectly identified SLC26A4 as the causative gene for KID, whereas the correct gene mutation responsible for KID is in GJB2. While mutations in either SLC26A4 or GJB2 can lead to deafness, only mutations in GJB2 cause KID. This example illustrates that even after fine-tuning, the models were still confused when conditions share similar symptoms.



Question: How many people are affected by Bietti crystalline dystrophy (BCD) ?

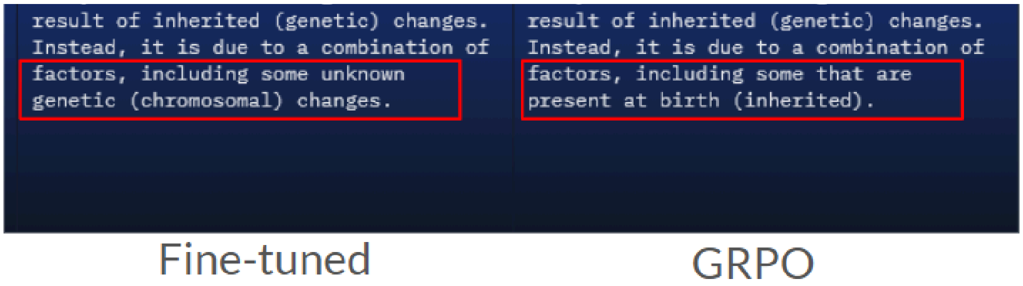
```
[
"Bietti crystalline dystrophy
has been estimated to occur in 1
in 67,000 people. It is more
common in people of East Asian
descent, especially those of
Chinese and Japanese background.
Researchers suggest that Bietti
crystalline dystrophy may be
underdiagnosed because its
symptoms are similar to those of
other eye disorders that
progressively damage the retina."
]
```

```
Bietti crystalline dystrophy (BCD)
is a disorder characterized by the
development of calcium deposits
(enlarging crystals) within the
lens of the eye. It is an inherited
condition that affects males more
commonly than females. The
prevalence of BCD is unknown.
```

```
Bietti crystalline dystrophy (BCD)
is a rare, inherited eye disorder in
which the lens of the eye becomes
yellow. The lens is the clear, dome-
shaped structure in the front of the
eye that helps focus light onto the
retina. BCD is also associated with
progressive cloudiness of the
cornea, the clear, dome-shaped
tissue covering the front of the
eye. BCD begins during childhood and
its symptoms progressively worsen
over time. Some people with BCD may
also develop cataracts. There is no
cure for BCD.
```

Fig. 8: Qualitative analysis of sample questions

From Fig. 8, we can observe that the fine-tuned models failed to learn some knowledge from the dataset. The fine-tuned model did not know the prevalence of BCD, even though it was present in the training dataset.



```
result of inherited (genetic) changes.
Instead, it is due to a combination of
factors, including some unknown
genetic (chromosomal) changes.
```

```
result of inherited (genetic) changes.
Instead, it is due to a combination of
factors, including some that are
present at birth (inherited).
```

Fig. 9: Qualitative analysis of sample questions

From Fig. 9, we can see that GRPO correctly guided the model to use simpler words, replacing “chromosomal” and “genetic” with “inherited” to simplify the answer without altering the meaning of the generated text.

6. Discussion

Several limitations of our study are presented below.

- **Insufficient data.** The number of samples in our dataset might not have been enough to impart new specialised knowledge to the smaller LLM, resulting in poorer QA performance.

- **Multiple reference answers to each question.** Our dataset contained multiple responses for similar or same questions, which may have caused fluctuations in training and validation loss due to the inconsistency of answers. Hence, this might have hampered the rate of convergence of the models trained, thereby hindering the fine-tuning process.
- **Limited hyperparameter tuning.** For our fine-tuning ablation studies, we selected two appropriate values for each of the LoRA rank and the learning rate, running a total of 4 experimental trials. This meant that our investigation was limited in terms of both the number of hyperparameters tuned, as well as the range of values considered for these variables. Thus, there might still be a noticeable disparity between our chosen set of hyperparameters and the theoretical optimum. Consequently, Trial 2 might not be reflective of the true best model.
- **Accuracy-readability trade-off.** Although we were able to guide the GRPO model towards generating text that was easier to understand, it became more prone to generating text that was factually inaccurate. Therefore, we found that RL with GRPO might not be adequate for the simplification of healthcare QA.

7. Conclusion and Next Steps

With reference to the aforementioned limitations, future work can proceed in the following directions.

- **Increase dataset size with synthetic data generation.** To address the issue of limited training samples, the use of a larger LLM to generate quality QA pairs can be explored – this would supplement the original MedQuAD dataset. The act of training SmolLM models on such synthetic data corresponds to a form of knowledge distillation, in which the smaller student model attempts to learn from the outputs of the larger teacher model.
- **Aggregate multiple reference answers for each question.** We can further process the training split before fine-tuning. If there are multiple ground truths for a given question, we can use a larger LLM to concatenate and summarise these answers. This would enable us to obtain a unified response for each question of the training set, which could improve convergence during the fine-tuning phase.
- **Increase thoroughness of hyperparameter tuning.** A more in-depth investigation on the fine-tuning stage can be conducted. We can look at adjusting other hyperparameters, such as the LoRA dropout and the weight decay for AdamW. Moreover, we can tune these variables over a larger range of values. For example, we can experiment with larger learning rates, since this appears to be the main factor impairing model performance. By conducting a grid search on this larger hyperparameter space, we may be able to obtain a fine-tuned model that is closer to the true optimum.
- **Improve the reward mapping for GRPO.** To more effectively enhance the readability of responses via GRPO, we can refine our reward modelling strategy by integrating additional factors of consideration – this would make the reward function more robust. For instance, we can introduce a new reward term which encourages "professional term exclusion" – in other words, this criterion explicitly favours model responses with lesser medical jargon (Qiu et al., 2025). With more computational resources, we can also explore using LLM-as-a-judge to reward response accuracy (Cao et al., 2024). By adding this term to the reward function, RL can guide the model towards improved readability, without greatly compromising the correctness of responses. This may potentially reduce the accuracy-readability trade-off incurred.

In this project, we investigated ways to fine-tune a small model with healthcare domain knowledge. We also harnessed GRPO to improve the style of LLM responses, making generations more digestible for the layperson. Our work has shown that the use of GRPO is effective, though there is still room for improvement.

In order to fine-tune LLMs for healthcare, a large amount of quality training samples is required, owing to the vast amounts of terminologies and facts that have to be learnt by the model. When data is scarce, we found that the effectiveness of such a fine-tuning workflow is diminished. To this end, we could consider implementing retrieval augmented generation (RAG) on a curated knowledge database, instead of attempting to improve the parametric knowledge of LLMs. This alternative approach provides additional documents on which the model can formulate its responses, thereby making it more likely for accurate answers to be produced.

In parallel with our study, there has been a growing body of research conducted on this particular problem statement. As late as October 2025, Qiu et al. (2025) adopted similar strategies to boost the coherence of LLM responses for healthcare QA, whilst suggesting new evaluation metrics for this task. Moving forward, it is hoped that subsequent work can pave the way for higher quality model responses in this domain, enabling society to reap the full benefits of QA in public health.

8. References

- Abacha, A. B., & Demner-Fushman, D. (2019). A Question-Entailment Approach to Question Answering. *BMC Bioinformatics*, *20*(511). https://doi.org/10.1186/s12859-019-3119-4
- Allal, L. B., Lozhkov, A., Bakouch, E., Blázquez, G. M., Penedo, G., Tunstall, L., Marafioti, A., Kydlíček, H., Lajarín, A. P., Srivastav, V., Lochner, J., Fahlgren, C., Nguyen, X.-S., Fourrier, C., Burtenshaw, B., Larcher, H., Zhao, H., Zakka, C., Morlon, M., … Wolf, T. (2025). SmolLM2: When Smol Goes Big — Data-Centric Training of a Small Language Model. https://arxiv.org/html/2502.02737v1
- Cao, Y., Zhao, H., Cheng, Y., Shu, T., Chen, Y., Liu, G., Liang, G., Zhao, J., Yan, J., & Li, Y. (2024). Survey on Large Language Model-Enhanced Reinforcement Learning: Concept, Taxonomy, and Methods. *IEEE Transactions on Neural Networks and Learning Systems, 36*(6), 9737 – 9757. https://arxiv.org/abs/2404.00282
- Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 10088 – 10115. https://doi.org/10.48550/arXiv.2305.14314
- Hosseini, P., Sin, J. M., Ren, B., Thomas, B. G., Nouri, E., Farahanchi, A., & Hassanpour, S. (2024). A Benchmark for Long-Form Medical Question Answering. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2411.09834
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *The Tenth International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2106.09685
- Kincaid, J. P., Fishburne, R. P., Jr., Rogers, R. L., & Chissom, B. S. (1975). *Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel*. Institute for Simulation and Training. https://stars.library.ucf.edu/cgi/viewcontent.cgi?article=1055&context=istlibrary

- Leo, C. (2024, July 16). *The Math Behind Multi-Head Attention in Transformers.* Towards Data Science. https://towardsdatascience.com/the-math-behind-multi-head-attention-in-transformers-c26cba15f625/
- Miller, J. (2024). Healthcare NLP: LLMs, Transformers, Datasets [Data set]. Kaggle. https://www.kaggle.com/datasets/jpmiller/layoutlm
- Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., & Finn, C. (2023). Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv.org*. https://arxiv.org/abs/2305.18290
- Qiu, W., Huang, T., Rullo, R., Kuang, Y., Maatouk, A., Ramos, S. R., & Ying, R. (2025). REPHQA: Evaluating Readability of large Language Models in Public Health Question Answering. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2509.16360
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347.* https://doi.org/10.48550/arXiv.1707.06347
- Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., & Guo, D. (2024). DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300v3*. https://doi.org/10.48550/arXiv.2402.03300
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, *30*, 5998 – 6008. https://arxiv.org/abs/1706.03762
- Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., Liu, X., Lin, H., Lin, Z., Ma, B., Sheng, G., Tong, Y., Zhang, C., Zhang, W., Zhu, H., … Wang, M. (2025). DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476.* https://doi.org/10.48550/arXiv.2503.14476
- Zhang, Y. (2025, February 7). DeepSeek-R1 Dissection: Understanding PPO & GRPO Without Any Prior Reinforcement Learning Knowledge. *Hugging Face*. https://huggingface.co/blog/NormalUhr/grpo

9. Appendix

We used GPT-5 to assist in the creation of code. We are responsible for the content and quality of the submitted work.

The following is a series of relevant links for our project.

- GitHub repository: https://github.com/yjiahao/MediLiteQA
- Cleaned MedQuAD dataset: https://www.kaggle.com/datasets/bingxuanchia/dsa4213-medquad-processed-dataset
- Model weights:
  - https://huggingface.co/chiabingxuan/SmolLM2-1.7B-Instruct-MediLite-QA-Rank8-Quantized-LowLR
  - https://huggingface.co/Jiahao123/SmolLM2-1.7B-Instruct-MediLite-QA-Rank8-Quantized-HighLR

    - https://huggingface.co/Jiahao123/SmolLM2-1.7B-Instruct-MediLite-QA-Rank16-Quantized
    - https://huggingface.co/Jiahao123/SmolLM2-1.7B-Instruct-MediLite-QA-Rank16-Quantized-HighLR
    - https://huggingface.co/Jiahao123/medilite-grpo-v1
- Hugging Face dataset containing model responses for final evaluation: https://huggingface.co/datasets/Cowboygarage/MediLite-QA-Response-Evaluation
- Trackio dashboards for the visualisation of fine-tuning metrics:
    - https://huggingface.co/spaces/Jiahao123/MediLiteQA (see “medilite-finetuning” and “medilite-grpo”)
    - https://huggingface.co/spaces/chiabingxuan/MediLiteQA (see “medilite-finetuning-v2”)