# EduGPT - DPO is all you need

Ho Nam Wong | 383111 | ho.wong@epfl.ch
Yufan Deng | 383656 | yufan.deng@epfl.ch
Tytarenko Tymur | 375184 | tymur.tytarenko@epfl.ch
NLPG Team

## Abstract

In this project, we propose EduGPT to act as a helpful course assistant for EPFL. We achieve this by firstly doing Direct Preference Optimization (DPO) on the base model, OpenHermes 2.5 Mistral 7B (Ope, 2024), with EPFL preference pair dataset, and then improving MCQ answering performance by fine-tuning with various MCQ datasets. Finally, the model's memory cost is improved by Quantization.

## 1 Introduction

Large language models (LLMs) achieved great break-through recently, and its power of generating and understanding text is attested by many successful LLM. LLM has huge potential to be used for educational purpose such as provide answer and explanation to a academic questions. In this project, we aim to provide EduGPT targeting for academic questions answering.

Current open-source LLM is trained with general text, and thus often struggle with accuracy, and efficiency for academic questions. Methods like Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017) and Direct Preference Optimization (DPO) (Rafailov et al., 2023) help improving the model by teaching with human's preference. Many MCQ datasets are open-sourced, and can ensure the correctness. Quantization methods effectively help reduce the memory cost of LLM both in training stage and inference stage.

EduGPT improve the accuracy and efficiency by building on the base model, OpenHermes 2.5 Mistral 7B (Ope, 2024), and enhancing it through a series of specialized training processes. Firstly, we applied DPO using an EPFL-specific preference pair dataset to align the model's responses with the expectations of the EPFL academic environment. Following this, we fine-tuned the model with a variety of MCQA datasets to boost its performance on standardized testing formats. Finally, we tackled the issue of high memory usage by implementing advanced quantization techniques (Dettmers et al., 2024, 2022), reducing the model's footprint without significantly compromising its performance.

The results demonstrate that EduGPT achieves decent MCQA accuracy and overall response quality, while the quantization process ensures that these enhancements are achieved in an efficient manner.

## 2 Related Work

### 2.1 Human preference

Recent advancements have focused on aligning large language models (LLMs) with human preferences to improve their performance in specific applications. One effective method is Reinforcement Learning from Human Feedback (RLHF), which fine-tunes models using human feedback to enhance response quality and relevance. This method has been used successfully in educational tools to provide personalized assistance and feedback to students (Christiano et al., 2017). Direct Preference Optimization (DPO) is another approach that directly optimizes model parameters based on human preference data, streamlining the alignment process by bypassing intermediate reward modeling. This method can be particularly useful in educational settings where timely and contextually appropriate responses are crucial (Rafailov et al., 2023).

### 2.2 Multiple Choice Question Answering

Multiple choice question answering (MCQA) is similar to the task of question answering, in which the model needs to generate answer to a question but from a fixed set of options instead of any answer (hug). For question answering, the question will be input to the model as prompt and the generated output will be the answer, while the output of the MCQA can be fixed to options letter, e.g. "A,

B, C, D", so either the output can either be letter only or a repetition of the option. In (Mondal), the correct option letter followed with repetition of the correct option is concatenated to the question to form the text used to do supervised fine-tuning training (SFT). In this project, we also following this pipeline to fine-tune our model with MCQ datasets to improve its MCQA performance.

## 2.3 Quantization

Usually models are trained and stored with float 32 bit or float 16 bit. Converting the data type to 8-bit or 4-bit can greatly reduce the memory cost. In (Dettmers et al., 2022), vector-wise quantization to 8-bit with separate normalization constants is used for each inner product in the matrix multiplication, and 16-bit is used for the 0.1% outlier cases. Memory can be effectively reduced without losing much performance. Memroy can be further reduced by 4-bit NormalFloat proposed in (Dettmers et al., 2024). Apart from model's quantization, the optimizer and LoRA in training process also can be improved with quantization (Dettmers et al., 2021, 2024). In this project, we utilize the QLoRA and 4-bit quantization (Dettmers et al., 2024) to reduce the gpu memory cost of out model's training and inference.

## 3 Approach

The overall pipeline of EduGPT is shown in Figure 1. For DPO, since completions in EPFL dataet are not sampled from the base model, we first apply Supervised Fine-tuning Training (SFT) with the chosen answers to base model to initialize $\pi_{\text{ref}}$ (Rafailov et al., 2024). Then, we do DPO training with EPFL preference pair dataset. After that, we improve the MCQ answering performance by fine-tuning the model after DPO with MCQA datasets mentioned in Sec. 4.1.2. Finally, the model will be quantized to 4-bit to reduce memory cost while preserving performance.

## 3.1 Models

We choose Mistral 7B (Jiang et al., 2023) as base model architecture for its efficiency and performance. Key features include Sliding Window Attention (SWA), which reduces the quadratic complexity of attention operations and improves throughput with a window size of $W = 4096$. The Rolling Buffer Cache further optimizes memory usage by maintaining a fixed cache size of $W$, overwriting old values as new ones are added. Addition-ally, the model utilizes pre-fill and chunking techniques to handle long prompts efficiently, breaking them into smaller chunks for processing.

In the preliminary experiment, we test the base model's capability of answering MCQ and how well can it fit DPO training. Mistral 7B is verified to perform better than other models with same size, e.g. Llama 3 8B, and and much better than those smaller models, e.g. TinyLlama or Gemma 2B (Zhang et al., 2024; Team et al., 2024). We also found that when doing DPO training, Chat Model format is more suitable for the given EPFL Course Dataset. Therefore, we choose the Open-Hermes 2.5 Mistral 7B (Team et al., 2024), a fine-tuned Mistral model following chat model template, as our reference model. The hugging face id is "teknium/OpenHermes-2.5-Mistral-7B".

## 3.2 Preference Pair Data collection

To make model learn as much knowledge as possible through DPO, we would like to instruct GPT-3 to generate lexically rich and complex answers. To achieve so, we set the instruction as

> You are a helpful assistant that answers question and gives explanation

To generate diverse responses, for each question, we use two chat instances to generate two answers. Then we manually annotate compare the two answers by Overall, correctness, relevance, clarity, completeness, and we further incorporate two important factors, conciseness and engagement, which is believed to be important for course assistant.

## 3.3 Direct Preference Optimization

As mentioned in (Rafailov et al., 2024), since completions in EPFL dataet are sampled from ChatGPT instead of the base model, we first apply Supervised Fine-tuning Training (SFT) with the chosen answers to base model to initialize $\pi_{\text{ref}}$, which is the module (a) in Fig. 1. Our SFT procedure follows the standard loss which aims at maximizing likelihood of preferred completions $(x, y_w)$ i.e.

$$\pi_{\text{ref}} = \arg\max_{\pi} \mathbb{E}_{x,y_w \sim D}[\log \pi(y_w|x)].$$

Then we do direct preference optimization (DPO), which is the module (b) in Fig. 1. For DPO, the loss $L_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}})$ is computed with $\beta = 0.1$
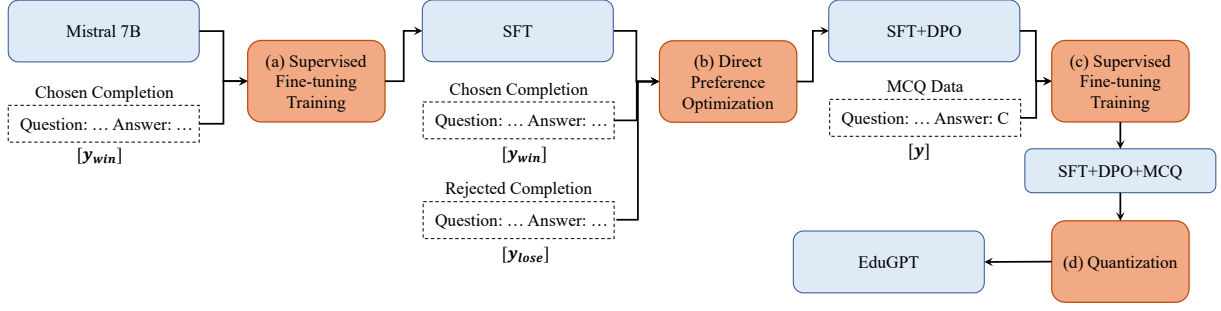
Figure 1: Pipeline of EduGPT. Module (a): do SFT on the base model Mistral 7B with chosen completion. Module (b): do DPO with preference pair. Module (c): do SFT with MCQ data. Module (d): do Quantization.

by

$$-\mathbb{E}_{(x,y_w,y_l)\sim D}\big[\log\sigma(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}$$
$$-\beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\big].$$

### 3.4 MCQ Fine-tuning

MCQ answering fine-tuning process is similar to question answering. The correct option letter is concatenated to the question, and the concatenated text is used to fine-tune the model following the standard SFT training decribed in Section 3.3, which is the module (c) in Fig. 1.

### 3.5 Quantization

After DPO and MCQ training, we quantize our final model SFT+DPO+MCQ to 4-bit and also to 8-bit for comparison with methods in (Dettmers et al., 2022, 2024), which is the module (d) in Fig. 1.

## 4 Experiments

### 4.1 Dataset

#### 4.1.1 Preference Dataset

For DPO task, we only utilize the EPFL preferences data in order to maximize its performance on EPFL courses. After removing the dirty preference pair where the answer are '...', there are totally 26641 preference pairs from 1522 questions. The max length for prompt is within 1024, and only 18 answers exceed the length of 1024. Therefore, we use 1024 for both prompt max length and answer max length.

#### 4.1.2 MCQA Dataset

We select all questions from both EPFL-MCQ dataset and external datasets to form our MCQA dataset.

**EPFL-MCQ:** Among 1522 questions from EPFL preference pair dataset, 603 questions are 4-choice MCQ. We extract these 603 questions and use ChatGPT to generate the answer to become the EPFL-MCQ dataset. Details of ChatGPT usage is in the appendix.

**MMLU:** MMLU (MIT License) (Hendrycks et al., 2021) contains 115700 4-choice MCQ of various subjects.

**ScienceQA:** ScienceQA (CC BY-NC-SA 4.0) (Lu et al., 2022) contains many multimodal samples. Totally 1149 text 4-choice MCQ of various science subjects are selected.

**TruthfulQA:** TruthfulQA (Apache License, Version 2.0) (Lin et al., 2022) contains totally 234 4-choice MCQ of various topics.

#### 4.1.3 Data Preprocessing

For DPO and SFT before it, since the base model use chat model template, the EPFL preference pair data is pre-processed to the chat format —

<|im_start|> system

You are a helpful assistant that answers academic question and gives explanation.<|im_end|>

<|im_start|> user

Question<|im_end|>

<|im_start|> assistant

Answer<|im_end|>

The Question and Answer are the prompt and chosen / rejected answer in dataset.

For the MCQ fine-tuning, the same chat format is also applied with an extra instruction of single-choice question rule added before question —

<|im_start|> system

You are a helpful assistant that solve the multiple choice question and answer the correct option.<|im_end|>

<|im_start|> user

This question has a single-choice answer. Therefore, select only one answer that is true and none else. For example, if the options are A, B, C, and D, you may select only one of these (A, for example). Here is the question: Question<|im_end|>

<|im_start|> assistant

Answer:A/B/C/D<|im_end|>

## 4.2 Evaluation Metric

### 4.2.1 Basic Evaluation

Two evaluation metric inherited from DPO task and MCQA task are Reward Accuracy and MCQ Accuracy. For Quantization, the model's size and MCQ Accuracy will be evaluated.

**Reward ACC:** The reward is computed by $\log \pi_\theta(y|x) - \log \pi_{\text{ref}}(y|x)$. The Reward ACC is the accuracy that the predicted reward of chosen completion is bigger than the predicted reward of rejected completion.

**MCQ ACC:** The MCQ ACC is the rate of correct option prediction.

**Model Size:** The memory size of the model will be recorded against its performance of MCQ ACC.

### 4.2.2 Extra Evaluation

Besides, we would like to evaluate the generation ability of the model before MCQA, which can demonstrate whether it acquires the knowledge from the training data. For these metrics, we compute the "margin" score i.e. the score of (generation, chosen) - the score of (generation, rejected).

**Bertscore:** We use the F1 score of BERTScore (Zhang et al., 2019), a metric based on BERT embedding, to evaluate the similarity between candidate and reference sentences. BERTScore utilizes pre-trained BERT model to compute the cosine similarity between token embedding of the candidate and reference sentences. It considers precision, recall, and F1 score, where F1 score is the harmonic mean of precision and recall, providing a balanced measure of similarity. This approach captures both lexical and semantic similarity, making it suitable for text generation tasks.

**BLEU-1 (Bilingual Evaluation Understudy):** BLEU-1 (Papineni et al., 2002) is a metric for evaluating the quality of text which has been machine-translated from one natural language to another. It compares the 1-grams of the candidate translation with the 1-grams of the reference translations and counts the number of matches. These matches are then weighted and averaged to produce a final score that ranges from 0 to 1, where a higher score indicates a closer match to the reference translations.

**RougeL (Recall-Oriented Understudy for Gisting Evaluation) :** ROUGE-L score (Lin, 2004) measures the longest common subsequence between the candidate and reference texts. This metric is particularly effective for evaluating the quality of text generation tasks like summarization. A higher ROUGE-L score indicates better overlap and alignment with the reference text.

## 4.3 Baselines

For DPO, the optimized model first go through SFT to align with the completion in dataset and then go through the DPO process. It is compared with baseline of model that directly do DPO, model that only have SFT, and the base model Mistral 7B without any training.

For MCQA, the baseline is the model after SFT and DPO from previous stage.

For Quantization, the baseline is the original checkout stored after MCQ fine-tuning, data type of which is float 16.

## 4.4 Experimental details

In the experiments, all trainings are done on a single A100 or RTX4090, with global batch size of 32 and seed 3407. After testing between learning rate of 2e-5 and 5e-5, we discover that both SFT, DPO, and MCQ fine-tuning have better performance with learning rate 5e-5 and 100 warm up steps. Both the SFT and DPO went through 2 epoch of EPFL dataset. Both of them use the whole EPFL preference pair dataset as the training set. For DPO we randomly pick 10% of thre EPFL dataset questions as validation set. MCQ fine-tuning went through 2 epoch of all MCQA datasets mentioned in Sec. 4.1.2.

To accelerate our trainings in both SFT, DPO, and MCQ fine-tuning, we use QLoRA (Dettmers et al., 2024) with rank 16. We have tried rank 8 and discover that rank 16 is better.

After all the three training, the checkpoint is originally stored in float 16. For the quantization

experiments, the original checkpoint is quantized to 8-bit and 4-bit for evaluation.
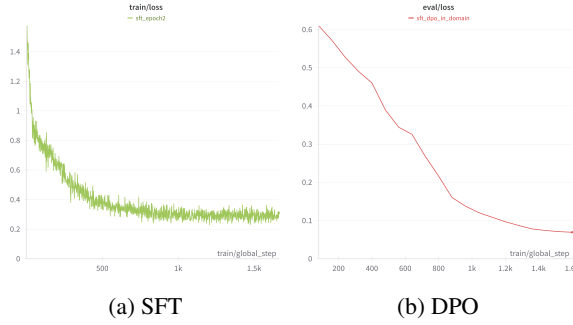


(a) SFT                    (b) DPO

Figure 2: Loss of SFT and DPO training



(a) Reward Accuracy        (b) Reward Margin

Figure 3: Evaluation of DPO. (a) shows the Reward ACC of each training step. (b) shows the margin between the reward of chosen completion and reward of rejected completion.

## 4.5  Results

### 4.5.1  Basic Metrics

Training loss of SFT and DPO are presented in Fig. 2. Evaluation results of DPO are shown in Fig. 3. DPO achieve nearly perfect Reward ACC (around 95%) on EPFL dataset after two epoch of training.

The MCQ ACC results are shown in Table 1. We evaluate the model on four data source seperately, in which 'epfl' is the full EPFL-MCQ dataset, 'sci' is the full ScienceQA dataset, 'truth' is the full TruthfulQA dataset, and 'mmlu' is the 500 samples from MMLU dataset. From the result, we can conclude that the 2-epoch MCQ fine-tuning do help increase the MCQA performance of the baseline model (sft-dpo) with an 11% to 27% increase of MCQ ACC.

For quantization, the model size vs MCQ ACC result is shown in Figure 4 and the exact MCQ ACC is shown in Table 1. The MCQ ACC is also evaluated for four data source separately. After quantized to 4-bit, the MCQ ACC drops less than

6% for all data source, and only drops 1% on mmlu dataset, while the model size is 0.3 times of the original model size. The 8-bit quantization result is also reasonable, with MCQ ACC higher than 4-bit and lower than 16-bit and model size also in between.

|         | epfl  | sci   | truth | mmlu  |
|---------|-------|-------|-------|-------|
| sft-dpo | 0.525 | 0.79  | 0.59  | 0.758 |
| mcq     | **0.638** | **0.962** | **0.863** | **0.862** |
| mcq 8bit| 0.625 | 0.95  | 0.861 | 0.86  |
| mcq 4bit| 0.60  | 0.931 | 0.795 | 0.85  |

Table 1: Result of MCQ fine-tuning. The MCQ ACC from each dataset. The 'sft-dpo' is the baseline model after finishing DPO training. 'mcq' is the model after MCQ fine-tuning, and 'mcq 8bit' and 'mcq 4bit' are the quantized version of 'mcq'.
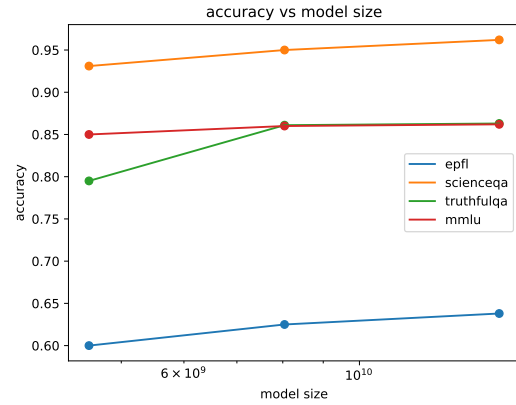


Figure 4: Quantization results. The 4-bit quantization model has size of 4.5e9 (left points), the 8-bit quantization model has size of 8e9 (middle points), and the 16-bit original model has size of 1.5e10 (right points). The accuracy is the MCQ ACC on each dataset.

### 4.5.2  Extra Metrics

We also use Bertscore, BLEU-1 and RougeL to evaluate our models (check section 4.2.2). We choose 40 batches ($40 \times 8 = 320$ questions) to evaluate and compare the generating quality of DPO+SFT model, SFT model, DPO model, base model. As shown in Table 2, we discover that there is increase in all metrics compared with the base model.

## 5  Analysis

### 5.1  Selected example

The selected question is:

|          | bleu   | F1     | RougeL |
|----------|--------|--------|--------|
| mistral7b | 0.0495 | 0.0009 | 0.0108 |
| sft      | 0.0499 | 0.0016 | 0.0145 |
| dpo      | 0.0463 | 0.0015 | 0.0125 |
| sft-dpo  | **0.0527** | **0.0029** | **0.0192** |

Table 2: Averaged score over batches, The "sft-dpo" is the model after finishing SFT and DPO training, the "dpo" is the model directly finetuned by DPO without SFT, the "sft" is the model after SFT without DPO, and the "mistral7b" is the base model.

> Question: If process i fails, then eventually all processes j≠i fail Is the following true? If a process j≠i fails, then process i has not failed.

The model after SFT+DPO generate the following answer:

> No, the second statement is not necessarily true. The given statements suggest a causal relationship between the failure of process i and the failure of all other processes (j≠i). However, the failure of another process j≠i does not imply that process i has not failed.
>
> Processes i and j could be independent, meaning the failure of one does not necessarily cause the other to fail. In other words, the failure of j≠i does not provide evidence that process i has not failed.
>
> To better understand the relationship between these processes, more information about the system or the nature of the processes would be needed. Without that context, we cannot assume a causal relationship or make definitive claims about the system's behavior."

The base model Mistral 7B generate the following answer:

> No, the statement is not necessarily true. In the given situation, it states that if process i fails, then eventually all processes j≠i fail. However, this does not imply that if a process j≠i fails, then process i has not failed.
>
> For example, imagine we have two processes, A (i) and B (j≠i). The statement says that if A fails (process i), then eventually B will fail as well, since it's different from A. However, B failing does not

mean that A has not failed or that A is still functioning. The failure of process B is a separate event and is not related to whether process A has failed or not. Each process operates independently and its failure does not determine the success or failure of another independent process.

We discover that our trained model tends to provide a little more than the question after completely explained the question, such as "more information would be needed" to understand the system. This is favorable because students may met a situation when more context is provided, which making this answer more helpful.



(a) Chosen reward     (b) Rejected reward

Figure 5: Reward of chosen completion and reward of rejected completion of each training step.

## 5.2 Chosen and Rejected rewards decrease at the same time

We discover an interesting counter-intuitive phenomenon that both chosen reward and rejected reward decreases at the same time as shown in figure 5 [c,d], this is also observed by other groups of researchers in this field recently (Guo and Xiong, 2024). To further analyze this, we recall the gradient computation as demonstrated in figure 6. Given that during the training process, the possibility of rejected samples $\pi_\theta(y_l|x)$ decreases, the coefficient of $\nabla_\theta \pi_\theta(y_l|x)$ will dominate the gradient, when in extreme cases (it approaches zero), it will explode to infinity. This might explain the reason of $\pi_\theta(y_w|x)$ term is not optimized (maximized), and keep decreasing in observations.

We suspect that controlling $\pi_\theta(y_l|x)$ can avoid overoptimization on the rejected samples as said in (Guo and Xiong, 2024). However, this is still an open question, and we leave it for future research.

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{DPO}} \left( \pi_{\boldsymbol{\theta}}; \pi_{\mathrm{ref}} \right)$$

$$= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[ \left( 1 - \sigma \left( \beta \log \frac{\pi_{\boldsymbol{\theta}} \left( y_w \mid x \right)}{\pi_{\mathrm{ref}} \left( y_w \mid x \right)} - \beta \log \frac{\pi_{\boldsymbol{\theta}} \left( y_l \mid x \right)}{\pi_{\mathrm{ref}} \left( y_l \mid x \right)} \right) \right) \right.$$

$$\left. \times \left( \beta \frac{\pi_{\mathrm{ref}} \left( y_w \mid x \right)}{\pi_{\boldsymbol{\theta}} \left( y_w \mid x \right)} \nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}} \left( y_w \mid x \right) - \beta \frac{\pi_{\mathrm{ref}} \left( y_l \mid x \right)}{\pi_{\boldsymbol{\theta}} \left( y_l \mid x \right)} \nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}} \left( y_l \mid x \right) \right) \right].$$

Figure 6: Gradient of DPO objective

## 6 Ethical considerations

Our model could be adapted to handle other high-resource languages such as French, German, Spanish, and Chinese by incorporating extensive multilingual datasets (de Gibert et al., 2024), utilizing transfer learning, and implementing cross-lingual training techniques. We could use data augmentation techniques, such as back-translation, to generate synthetic data and collaborating with local communities and linguistic experts to gather and validate language data.

Adapting EduGPT to interact with users in signed languages, such as American Sign Language (ASL), involves integrating video data of signed language, visual recognition and gesture generation techniques. We should choose a base model with multimodal functionalities such as LLaVa (Liu et al., 2023), and integrate it with devices capable of capturing and interpreting sign language, similar to advancements in gesture recognition technology (Camgoz et al., 2020). Collaboration with sign language experts and the deaf community is crucial to ensure accuracy and cultural appropriateness.

EduGPT provides personalized learning experiences since it allows excessive interactions. With multilingual abilities, EduGPT could also interact in French or German, tailoring the need for local students at EPFL. However, the model may perpetuate biases present in the training data, leading to unfair or inaccurate responses, particularly concerning gender, race, or socioeconomic status.

To mitigate this harm, we will implement robust mechanisms to detect and correct bias (Thomas Ph.D, 2023), regularly update training data to include diverse and representative samples (Dixon et al., 2018), and conduct ongoing assessments to identify and correct bias.

Besides, students might exploit EduGPT for cheating in homework and exams at EPFL, resulting in unfair cases. And they might also be overly dependent on EduGPT to answer academic questions without further thinking, posing harm to their problem-solving skills.

## 7 Conclusion

This project introduce EduGPT, a chat-LM targeting at answering academic questions. Based on Mistral 7B, EduGPT is firstly trained with SFT, DPO, and MCQ fine-tuning to increase answering ability, and then compressed with quantization to increase efficiency. The experiment results show that EduGPT can achieve high MCQ ACC and Reward ACC, and the quantization can significantly reduce memory cost without losing performance.

However, there are limitations present in this work. First, it takes excessive time to finetune the model even with quantization and QLoRA, it might be interesting to investigate a sample-efficient strategy to train the model. Moreover, as demonstrated in Section 5.2, it is required to figure out a way to make chosen rewards increasing and rejected rewards decreasing for more efficient training. Lastly, we are still using Vanilla DPO to train our model while we can use several variants (Pal et al., 2024), (Zeng et al., 2024) to improve the training stability and performance of the model.

## 8 Team Contributions

**Yufan Deng**: DPO experiment and report, MCQ dataset preparing, MCQ fine-tuning experiment and report, Quantization experiment and report
**Ho Nam Wong**: DPO and SFT training, evaluation experiment, investigate phenomenon on decrease of both chosen / rejected rewards, report writing
**Tytarenko Tymur**: Evaluation Metric initial design, investigate ethical issue, DPO dataset preparing, report writing

# References

Multiple choice — huggingface.co. `https://huggingface.co/docs/transformers/en/tasks/multiple_choice`. [Accessed 13-06-2024].

2024. Openhermes-2.5-mistral-7b. `https://huggingface.co/teknium/OpenHermes-2.5-Mistral-7B`. Accessed: 2024-06-02.

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. 2020. Sign language transformers: Improving accuracy of sign language production. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Paul F Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307.

Ona de Gibert, Graeme Nail, Nikolay Arefyev, Marta Bañón, Jelmer van der Linde, Shaoxiong Ji, Jaume Zaragoza-Bernabeu, Mikko Aulamo, Gema Ramírez-Sánchez, Andrey Kutuzov, Sampo Pyysalo, Stephan Oepen, and Jörg Tiedemann. 2024. A new massive multilingual dataset for high-performance language technologies.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2021. 8-bit optimizers via block-wise quantization. *arXiv preprint arXiv:2110.02861*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. pages 67–73.

Shangmin Guo and Wei Xiong. 2024. Alignment guidebook. `https://www.notion.so/Alignment-Guidebook-e5c64df77c0a4b528b7951e879397fa78#a9a397ec7bbf4c4486334383f9087468`. Accessed: 2024-06-02.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Saankhya Mondal. Mistral Instruct 7B Finetuning on MedMCQA Dataset — saankhya.medium.com. `https://saankhya.medium.com/mistral-instruct-7b-finetuning-on-medmcqa-dataset-6ec25` [Accessed 13-06-2024].

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Ryan Rafailov, Uri Nisan, Ruiqi Zhang, and Daniel Ho. 2023. Direct preference optimization: Your language model is secretly a reward model. In *arXiv preprint arXiv:2305.18290*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker,

George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology.

Josh Thomas Ph.D. 2023. Bias in ai models and a hybrid human-ai framework for mitigation.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. 2024. Token-level direct preference optimization.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

## A  AI Usage Appendix

The completions of preference pair data in Sec. 4.1.1 is generated by the ChatGPT API provided by CS552. The usage details are written in Sec. 3.2.

The answer of EPFL-MCQ dataset in Sec. 4.1.2 is also generated by the ChatGPT API provided by CS552. The instruction used for generation is

> You are a helpful assistant that choose the correct options and only answer in "A", "B", "C", or "D"

We generate the answer multiple times and choose the majority answer to ensure the correctness.