Progress Report: Aligning Models for AI & STEM Question Answering

Andrea Miele | 302925 | andrea.miele@epfl.ch
Luca Mouchel | 324748 | luca.mouchel@epfl.ch
Elia Mounier-Poulat | 314771 | elia.mounier-poulat@epfl.ch
Frederik Gerard de Vries | 369939 | rik.devries@epfl.ch
REAL

1 Introduction

In this report, we describe our progress in training an AI assistant to answer questions related to EPFL courses. The pre-trained model we chose is **Google's Gemma 2B** (Mesnard et al., 2024). We decided to train Supervised Fine-Tuning (SFT) + Direct Preference Optimization (DPO) and Base Model + DPO and we found that both yield similar performance. We will discuss this and the various datasets we use. Additionally, we provide an overview of the preliminary results, the experimental setup, and future work.

2 Dataset

SFT Phase. For the SFT phase, we first collected data from the *EPFL preference dataset* by selecting only the chosen samples and discarding the rejected ones. This way, the SFT model is trained with what are supposed to be "favorable" answer explanations. Additionally, we added two new datasets: a *math preference dataset* (Argilla, 2023), where we also used only the chosen responses, and the *MMLU dataset* (Hendrycks et al., 2021). For the *MMLU dataset*, we extracted questions about math, physics, machine learning, etc., and used the course-provided GPT wrapper to generate answers. We prompted the wrapper by giving it the correct answer to the question, then asked it to provide explanations for why the correct answer was correct and why the others were incorrect. This approach ensures that the explanations are rational and similar to the distribution of the answers in our dataset. Our final SFT training dataset used 80% of the *EPFL dataset*, with the additional *MMLU* and *math preference data*, totaling around 28K samples.

DPO Phase. For the DPO phase, we used a subset of the *EPFL dataset*. By using a subset, we ensure that the external data has a larger weight in the training data distribution. Specifically, we first filtered answers shorter than 20 words and then used 6,000 samples from *EPFL*, selecting 4 answers per question, and generated 2,000 samples with the GPT wrapper from the *AllenAI Math/QA dataset* (Amini et al., 2019). Additionally, we included the *Argilla Math Preference dataset*, which contains over 2,000 questions with both preferred and dispreferred samples (Argilla, 2023). The train-test split used 84% (10.4K samples) of the data for training, with 16% (2K samples) held out for evaluation. Please note the test set contains only samples from the *EPFL dataset*.

3 Model

Base Model. The base model we are using is **Gemma 2B** (Mesnard et al., 2024), designed by Google for strong generalist capabilities in language understanding, making it a good candidate for our project. This model uses transformer decoder architectures and incorporates state-of-the-art deep learning techniques. It is pre-trained on a high-quality dataset of English text from web documents, mathematics, and code, totaling 3 trillion tokens of training data. The model also aligns with our ethical considerations as the original training dataset was filtered to ensure safety and minimize biases by removing personal information and sensitive data.

SFT. We found that using the SFT model and the base model as reference yielded very similar performance, as depicted in Fig. 1. As a result, we decided to simplify our training process and run DPO using the base model as the reference. SFT is primarily used to teach models how to integrate the distribution of the data and generate text accordingly. However, since our training data does not follow a specific distribution (e.g., a specific format for the answers), we decided that SFT was not necessary.

DPO Loss and Training. Building on this decision, we opted to optimize the model with DPO, using the base model directly as the reference model. To leverage the capabilities of a 2.5B model, we employed the Low-Rank Adaptation (LoRA) method (Hu et al., 2021), which creates an adapter and allows for fine-tuning on a subset of the parameters. By using this method, we significantly reduced the number of trainable parameters from 2.51 billion to 921,000, which represents 0.037% of the total parameters.

We trained the DPO model using the default loss function of the `DPOTrainer`, which is based on a log-sigmoid function applied to the difference in rewards between preferred and dispreferred responses.

4 Preliminary Training Results

4.1 Experimental Setup

SFT Phase. Since training the DPO with and without SFT led to similar performance (Fig. 1), we won't delve into the SFT evaluation further. However, for completeness, we provide the hyperparameters, we selected based on best practices, used for SFT in the Appendix 1.

DPO Phase. The hyperparameters (Table 2) for DPO were selected through empirical tuning and following best practices. A **batch size** of 4 was chosen to balance computational efficiency and the quality of gradient estimation, while a **learning rate** of 5×10^{-4} was set to achieve a good trade-off between convergence speed and stability. Training for 3 **epochs** proved sufficient for convergence without risking overfitting, as determined by validation performance. The **AdamW optimizer** was chosen for its effectiveness in handling sparse gradients and promoting better generalization. Additionally, a **cosine annealing scheduler** with 100 warm-up steps was implemented to stabilize early training and fine-tune the learning rate, thereby improving overall convergence and performance.

For metrics requiring inference, such as *win-rate* and DiscoScore, the evaluation was conducted on 100 samples, due to compute time constraints. However, for other metrics, computed in the `evaluator` module, we used the whole 2000-sample test set from the *EPFL dataset*.

4.2 Preliminary Results

The DPO model demonstrated a significant improvement over the base pretrained model in its capacity to prioritize correct answers over incorrect ones, as indicated by the reward margins, reward accuracies, win rate metrics, and DiscoScore (Zhao et al., 2022) We perform evaluation on two datasets. The first dataset $\mathcal{D}_1$ consists of 100 questions, with generated responses with the DPO and the base model. $\mathcal{D}_2$ consists of 2000 samples from the *EPFL dataset*.

Reward Margins and Accuracies. The reward margins measure the difference between the reward values assigned to preferred responses and dispreferred responses and the reward accuracies count how many times the chosen rewards are higher with respect to the rejected rewards. The reward margins were 1.8 and 0.66 during training and evaluation respectively and the accuracies were 0.94 and 0.64 for training and evaluation (Figures 2a and 2b). These results are motivating, as it is indicative of efficient training and quite robust performance on unseen data. This was performed on $\mathcal{D}_2$.

Win Rate. We crafted our own win rate metric using an AI expert system (detailed in Appendix A) to compare the fine-tuned model's effectiveness with its pretrained version in generating preferred answers. Our DPO-trained model achieved a win rate of 67.3%, while the base model won 32.7% of the time, with the GPT wrapper consistently avoiding 'tie'. This highlights the DPO model's integration of high-quality answers to STEM questions, offering superior explanations, as supported by the prompt to select the clearest and most accurate answer for computing the win rate. This was performed on $\mathcal{D}_1$.

DiscoScore. Finally, we integrated DiscoScore (Zhao et al., 2022) into our evaluation framework. This metric assesses discourse coherence and logical reasoning in text compared to a reference. The entity graph metric ranged from 0.82 for the reference model to 1.31 for the DPO, indicating success in learning more logical answers. This evaluation was performed on $\mathcal{D}_1$.

5 Specializations

5.1 Retrieval-Augmented Generation (RAG)

5.1.1 Data Preparation

For integrating RAG, we will use a subset of curated documents from `wiki-dpr` (Karpukhin et al., 2020) that our retriever can access during inference, in order to fit the 5GB constraint. Additionally, we have set aside a portion of this data specifically for validation and testing to evaluate the effectiveness of the retrieval component.

5.1.2 Model Adaptation

We are considering the integration of a document retriever with our fine-tuned **Gemma 2B** model using the `wiki-dpr` database. We will preprocess the data to make it fit in 5GB, by querying relevant documents related to the questions. Our initial idea is to query 5-10 documents for each of the questions in the *EPFL dataset*, and store them into a separate database. This ensures the documents are exactly related to the topics treated in our data. From these documents, we will retrieve context from the question at inference time and add it to the prompt we feed the model.

5.1.3 Baselines and Evaluation Metrics

We will use as baselines the base model and DPO model and compare them to our DPO model with RAG. For evaluation, we will compare it using metrics such as relevance and coherence of generated responses like DiscoScore, BLEU score, and automatic evaluation with the `GPT wrapper` (e.g., computing the *win-rate*). These metrics will help us assess the improvements brought by the retrieval component in terms of accuracy and informativeness.

5.2 Model Quantization

We will evaluate the quantized model on a subset of the *EPFL dataset*.

5.2.1 Model Adaptation

We plan to quantize our model, after our MCQ specialization training. We will use the Hugging Face Transformers library, as well as the `‘bitsandbytes‘` library. This process involves converting the model weights to a lower precision format. **Gemma 2B** uses 16-bit precision by default, which can be lowered to 8 or 4 bits.

5.2.2 Baselines and Evaluation Metrics

Evaluation metrics will include model size, inference speed, and accuracy. Specifically, we will measure the reduction in model size and improvement in inference speed achieved through quantization, while monitoring any changes in model accuracy. This will help us quantify the efficiency gains from quantization and ensure that the model's performance remains acceptable.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Argilla. 2023. Argilla/distilabel-math-preference-dpo · datasets at hugging face.

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)*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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, and Douglas Eck. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*. Version 4.

Wei Zhao, Michael Strube, and Steffen Eger. 2022. Discoscore: Evaluating text generation with BERT and discourse coherence. *CoRR*, abs/2201.11176.

A Metrics definition

A.1 Win Rate Definition and Calculation

In our project, we define the win rate to evaluate the performance of two explanation generation policies. The win rate is based on comparing the quality and correctness of explanations generated by a reference policy and a policy of interest for a set of STEM-related questions (Computer Science, AI, Mathematics, Physics).

Process

1. **Data Collection**: We gathered a dataset of questions with explanations generated by both policies.

2. **Evaluation**: Each question and its explanations were evaluated by an expert system (GPT-based model) to choose the better explanation or indicate a tie.

3. **Response Parsing**: The system's output was parsed to identify the selected answer: '1' (reference policy), '2' (policy of interest), or '3' (tie).

4. **Result Aggregation**: The results were aggregated into 'reference', 'policy', and 'tie'.

Formula

The win rate for each policy is calculated as:

$$\text{Win Rate} = \frac{\text{Number of Wins}}{\text{Total Evaluations} - \text{Number of Ties}} \tag{1}$$

For the reference policy $W_{\text{reference}}$:

$$W_{\text{reference}} = \frac{N_{\text{reference}}}{N_{\text{total}} - N_{\text{tie}}} \tag{2}$$

For the policy of interest W_{policy}:

$$W_{\text{policy}} = \frac{N_{\text{policy}}}{N_{\text{total}} - N_{\text{tie}}} \tag{3}$$

where $N_{\text{reference}}$ and N_{policy} are the counts of times each policy's explanation was chosen, N_{tie} is the number of ties, and N_{total} is the total number of evaluations.

This method allows us to objectively compare the performance of the policies in generating high-quality and correct explanations.

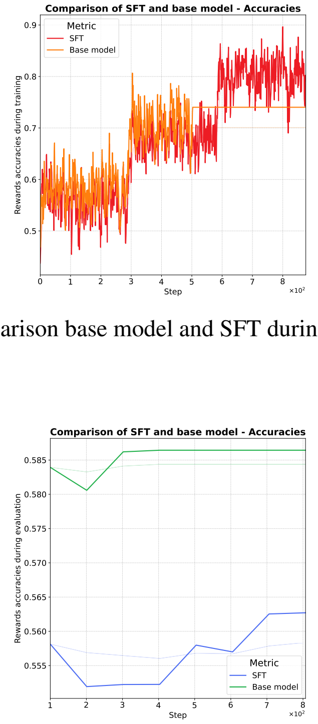
B Training details

B.1 SFT Hyperparameters

Table 1: Hyperparameters for Supervised Fine-Tuning (SFT).

Training	
Batch Size	8
Learning Rate	2e-4
Number of Epochs	4
Optimizer	Adam
Scheduler	Annealing linearly

B.2 SFT vs. Base Mode



(a) Comparison base model and SFT during training

(b) Comparison base model and SFT during evaluation

Figure 1: Performance differences between base model and SFT while training DPO.

B.3 Data Generation for DPO Using GPT Wrapper

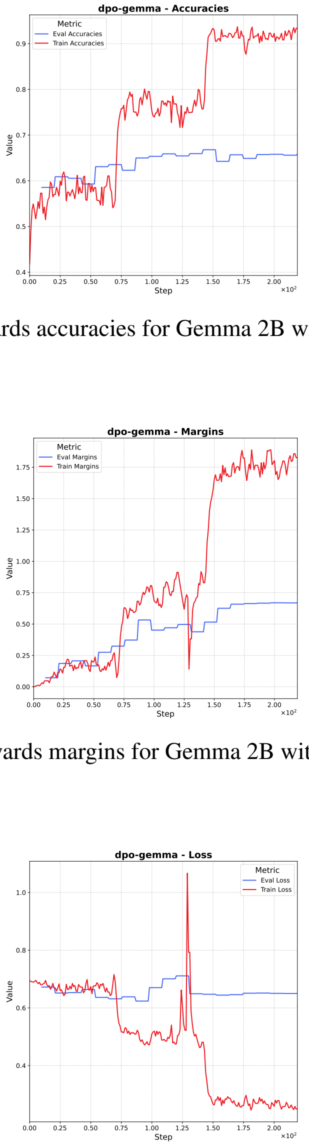
In this appendix, we describe the methodology used to generate our dataset leveraging the GPT wrapper in Python.

The dataset used for this step is `MathQA`. Each question is prepared with a prompt instructing the GPT model to act as a specialist in STEM fields, particularly in computer science, AI, mathematics, or physics. The model is tasked with evaluating each possible answer, determining its correctness, and explaining the reasoning behind its evaluation.

For each question, two separate GPT chats are created, and the prepared prompt is used to generate two distinct responses. To compare and rank these responses, we use a secondary prompt that evaluates the answers based on correctness, relevance, clarity, completeness, conciseness, and engagement. The ranking prompt instructs the GPT model to provide a JSON-formatted ranking result.

Based on the ranking results, we determine the chosen and rejected answers.

B.4 DPO training plots



(a) Rewards accuracies for Gemma 2B with DPO

(b) Rewards margins for Gemma 2B with DPO

(c) Loss plots for Gemma 2B with DPO

Figure 2: Metrics plots for Gemma 2B with DPO

B.5 DPO Hyperparameters

Table 2: Hyperparameters for Direct Preference Optimization (DPO).

	Training
Batch Size	4
Learning Rate	5×10^{-4}
Number of Epochs	3
Optimizer	AdamW
Scheduler	Cosine with 100 warm-up steps