# Progress Report: Finetuning the model

Malak Lahlou Nabil | 329571 | malak.lahlounabil@epfl.ch
Sara Anejjar | 329905 | sara.anejjar@epfl.ch
Frederic Khayat | 326634 | frederic.khayat@epfl.ch
Peter Abdel Massih | 325664 | peter.abdelmassih@epfl.ch
4-pack

## 1 Introduction

In this deliverable, we employed Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) techniques to finetune Meta-Llama-3-8B (AI@Meta, 2024). As a result, the model now aligns more effectively with the task of answering EPFL exam questions.

## 2 Dataset

### 2.1 Training Datasets

The Table 2 shows a summary of the SFT and DPO datasets that we used with their corresponding purposes.

### 2.2 MCQA

To extract the response letter from our model's initial output, we decided to implement a post-processing function instead of directly training on an MCQA dataset. This will help the model generate better answers since it will be able to perform chain of thought. However, we will still evaluate our model's MCQA performance on the following datasets:

- **Computer Science MCQA (3K)** (Lab, 2023)
- **Physics MCQA (600)**: We only kept physics questions that don't require images as additional information (Thomas, 2023).

## 3 Model

Based on findings from the DPO paper (Rafailov et al., 2023), we have determined that fine-tuning our model with SFT before employing DPO is crucial to maximize performance.

### 3.1 Supervised Finetuning

We fine-tuned our model using the SFTTrainer from the trl package (trl, 2024). The parameters were selected to balance speed and memory usage, to prevent out-of-memory (OOM) errors while maximizing efficiency. Below, we detail these parameters and provide justifications for our choices.

- From the **LoRA** study (Hu et al., 2021), we set alpha = 128 and rank = 64, based on the recommendation that alpha should be twice the rank. We also used Rank-Stabilized Adapters (rsLoRA) to adjust adapter weights. They will be multiplied by $\gamma_r$ where $\gamma_r = \frac{\alpha}{\sqrt{rank}}$ (Kalajdzievski, 2023).
- We attach LoRA adapters on critical linear layers involved in data transformation and attention mechanisms.
- **Batch sizes** of 4 for SFT and 2 for DPO were chosen to prevent out-of-memory issues.
- We employed **gradient accumulation** over four steps and **gradient checkpointing** every 4 steps to save memory (Chen et al., 2016).
- **Mixed precision training** used BF16 for weights and TF32 for matrix multiplications (Micikevicius et al., 2018).
- **Flash Attention 2** was implemented to speed up attention computation and reduce memory use (Dao, 2023).
- **DeepSpeed ZeRO** with CPU Offload was integrated for efficient memory management (Rajbhandari et al., 2020).
- Backward processing was managed by **Accelerate** (Gugger et al., 2022).

These choices reflect our commitment to maximizing efficiency and stability throughout the training process.

We loaded the datasets using the huggingface Datasets library and then converted them into the instruction format : `{"prompt": "...", "completion": "..."}`

### 3.2 Direct Preference Optimization

We maintained consistent general parameters for both SFT and DPO, and used the `DPOTrainer` from trl to align our model with the preference data (Rafailov et al., 2023). We used the SFT checkpoint as the reference model, along with a new set of LoRA layers configured with similar parameters. According to the Hugging Face alignment handbook, DPO needs a significantly smaller learning rate about 10-100 times less than SFT (Face, 2023). The learning rate went from 2e-5 for SFT to 1e-6 for DPO, a 20-fold decrease. We also chose a beta parameter of 0.1 to manage the strength of the alignment, ensuring minimal divergence from the reference model. We used a sigmoid loss function as in the original paper (Rafailov et al., 2023).

We reformatted our datasets into triplets: `{"prompt": "...", "chosen": "...", "rejected": "..."}`.

# 4 Results

## 4.1 Training Results

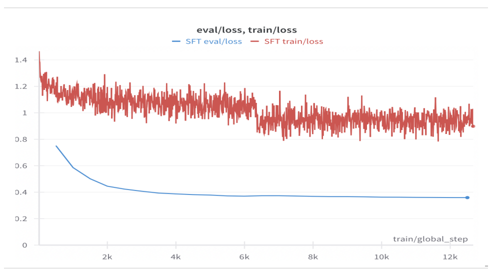


Figure 1: SFT training and evaluation loss for 2 epochs. The plot shows the significant decrease in both training and evaluation losses during the first epoch, followed by stabilization with minor decrease in second epochs.

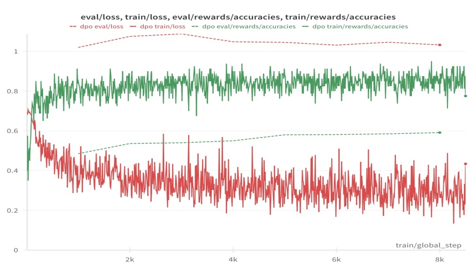


Figure 2: DPO training and evaluation loss, training and evaluation rewards accuracies. The DPO evaluation loss shows an increase at first and then a decrease with more steps, while the accuracies increase both for the evaluation and training

## 4.2 Benchmark

To quantify the improvements of DPO, we used the following benchmark datasets:

- Orca Math - Math questions (ibivibiv, 2024)
- DPO-datascience - Data science questions (Khan, 2024)
- M1 dataset - A subset of the M1 deliverable data
- dpo_preference_example.jsonl - The file that was present in the code template

We collected three metrics on our benchmarks:

1. **Accuracy:** Percentage of pairs where $\pi_\theta(y_w|x)/\pi_{\text{ref}}(y_w|x) > \pi_\theta(y_l|x)/\pi_{\text{ref}}(y_l|x)$.
2. **DPO Model Alignment:** Percentage of pairs where $\pi_\theta(y_w|x) > \pi_\theta(y_l|x)$.
3. **Ref. Model Alignment:** Percentage of pairs where $\pi_{\text{ref}}(y_w|x) > \pi_{\text{ref}}(y_l|x)$.

The results are summarized in the table below:

| | Accuracy | DPO Align. | Ref. Align. |
|---|---|---|---|
| **Example** | 0.897 | 0.848 | 0.692 |
| **M1** | 0.640 | 0.633 | 0.554 |
| **Orca-Math** | 0.777 | 0.978 | 0.973 |
| **Datascience** | 0.705 | 0.847 | 0.839 |

Table 1: Benchmark scores. The policy model is the fine-tuned Llama3 (SFT+DPO) and the reference model is Llama3 base (without any fine-tuning)

# 5 Quantization Specialization

We plan to refine our model by integrating QLoRA (Dettmers et al., 2023), which uses LoRA with a quantized version of our model. By leveraging the bitsandbytes library (Dettmers et al., 2021), we plan to quantize our model multiple times to gain a good understanding of the consequences of quantization on performance.

# 6 RAG Specialization

We plan to adapt our trained LLama-3 model for Retrieval-Augmented Generation (RAG) (Lewis et al., 2021) by setting a user query prompt for generating responses. We'll create embeddings for document chunks to enhance information retrieval. Furthermore, we'll configure a service context for managing large text segments and index documents, including EPFL course books, to enable efficient querying and retrieval of relevant information.

Table 2: Summary of SFT and DPO datasets

| Dataset | Purpose | SFT Size | DPO Size |
|---|---|---|---|
| M1 preference dataset | Questions from various EPFL courses | 25.8K | 25.8K |
| Stack Exchange (Internet Archive, 2024) | Dataset from multiple stack exchange forums. | 43K | 33.4K |
| Mathematical Reasoning (Hendrycks et al., 2021) | Complex mathematical problems | 10K | 2.4K |
| Programming Feedback (M-A-P et al., 2022) | Multiple programming tasks | 10K | - |
| Physics (Alves et al., 2021) | Physics questions with generated answers | 10K | - |
| Math DPO (argilla, 2024) | Math questions and answers | - | 2.4K |
| Python DPO (jondurbin, 2024) | Multiple Python programming tasks | - | 9K |
| STEM DPO (thewordsmiths, 2024) | Multiple questions from STEM fields | - | 30K |

We opted for these datasets since they were similar to courses at EPFL in maths, computer science, and physics.

## References


AI@Meta. 2024. Llama 3 model card.

L. G. A. Alves et al. 2021. Camel-ai-physics dataset. https://huggingface.co/datasets/lgaalves/camel-ai-physics.

argilla. 2024. Math dpo dataset. https://huggingface.co/datasets/argilla/distilabel-math-preference-dpo.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning.

Tim Dettmers, Mike Lewis, and Sam Shleifer. 2021. Bitsandbytes: 8-bit optimizers and matrix multiplication routines. https://github.com/facebookresearch/bitsandbytes. Accessed: 2024-06-03.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Hugging Face. 2023. The alignment handbook. Accessed: 2024-06-03.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

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.

ibivibiv. 2024. Orca math. Hugging Face Dataset Hub. Accessed: 2024-06-03.

Internet Archive. 2024. Stackexchange data dump.

jondurbin. 2024. Python dpo dataset. https://huggingface.co/datasets/jondurbin/py-dpo-v0.1.

Damjan Kalajdzievski. 2023. A rank stabilization scaling factor for fine-tuning with lora.

Anas Khan. 2024. Dpo-datascience. Hugging Face Dataset Hub. Accessed: 2024-06-03.

TIGER Lab. 2023. Mmlu-stem dataset. Hugging Face Dataset Hub.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

M-A-P et al. 2022. Codefeedback-filtered-instruction dataset. https://huggingface.co/datasets/m-a-p/CodeFeedback-Filtered-Instruction.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models.

thewordsmiths. 2024. Stem dpo dataset. https://huggingface.co/datasets/thewordsmiths/stem_dpo.

Derek Thomas. 2023. Scienceqa dataset. Hugging Face Dataset Hub.

trl. 2024. trl.