Size Does Not Matter: A Data-Centric Approach to Fine-Tuning

Pierre Lardet | 376393 | pierre.lardet@epfl.ch
Mika Senghaas | 377332 | mika.senghaas@epfl.ch
Ludek Cizinsky | 377297 | ludek.cizinsky@epfl.ch
MLP

1 Introduction

This report details preliminary training results for fine-tuning **Phi-3-mini** (Abdin et al., 2024) for scientific question-answering. The goal for this milestone is to use annotated preference pairs to align the model using Direct Preference Optimisation (DPO) (Rafailov et al., 2023).

2 Dataset

The dataset used consists of 26,738 pairs of answers to scientific questions, where one answer is preferred over the other. The data comprises 1,522 unique questions from 24 different courses at EPFL in the fields of computer science, mathematics and physics. Each pair of answers was generated by prompting GPT-3.5 and annotated using various ranking criteria and an overall ranking by students from the CS-552 course. An example annotation can be found in the Appendix Section A.1. Before any further processing, we peform an 80-20 train-validation split, leaving 21,390 pairs for training and 5,348 pairs for validation. To help with efficiency, we remove pairs with a token count above the 95th percentile.

We suspect that the dataset may contain noisy examples, such as answer pairs that are too similar or too different. We developed two heuristics to remove such examples. Both are based on an *agreement score* which is computed as the number of specific ranking criteria that are equal to the overall ranking:

Global Threshold (GT). We remove pairs where the agreement score is below a threshold λ.

Local Tolerance (LT). We remove pairs with agreement score smaller than the maximum agreement score for the same question minus a tolerance value λ.

3 Model

We use the pre-trained *Phi-3-mini-4k-instruct* model (Abdin et al., 2024), a 3.8B parameter transformer decoder-only architecture with 32 heads, 32 layers, and a hidden dimension of 3072. It was pre-trained on 3.3T tokens of heavily filtered web data and synthetic data. The model has been further fine-tuned and aligned. As our fine-tuning objective, we use the standard DPO loss (Rafailov et al., 2023), and recently proposed variants such as RSO (Liu et al., 2024) and IPO (Azar et al., 2023), the results of which are compared in Section 4.1.

4 Preliminary Training Results

We follow a careful two-step strategy for hyperparameter tuning. Firstly, we tune selected hyperparameters on a subset of the data. For tuning, we use a base training configuration detailed in Table 2. Secondly, we combine the most promising training configurations from the previous step and train and evaluate on the entire dataset. Our final model is then trained on the combined training and validation sets using the best performing configuration.

For training, we use the `DPOTrainer` from the Huggingface TRL library (HuggingFace). All runs use 4-bit quantised QLoRa fine-tuning to keep the model size and checkpoints manageable. We employ the `Unsloth` library as we found a 4x training speedup and 50% reduction in memory usage. All experiments were performed on a single NVIDIA V100 GPU with 32GB of memory.

For this milestone, we evaluate using the DPO loss, margins and accuracy on the training and validation sets. We refrain from using common MCQ benchmarks for evaluation as no emphasis was placed on improving MCQ performance at this stage.

Table 1: Training Configuration and Results.

Model	Configuration						Results
	LR	Rank	Loss	Beta	LB Smoothing	Data Filt.	Val. Accuracy (%) ↓
H4	4e-5	32	IPO	0.1	0.1	None	**67.01%**
M2	2e-5	16	IPO	0.1	0.0	None	66.61%
H2	4e-5	32	DPO	0.05	0.1	None	65.97%
H1	2e-5	16	DPO	0.4	0.1	None	64.76%
M1	2e-5	16	IPO	0.1	0.0	LT (λ=0)	64.69%
H3	4e-5	32	DPO	0.4	0.0	None	63.89%
H5	4e-5	32	DPO	0.4	0.1	LT (λ=0)	62.07%

4.1 Initial Hyperparameter Tuning

For initial hyperparameter tuning we train on a subset of 4,000 examples and evaluate on 640 examples from the validation set. After reviewing the literature on DPO training, we chose to tune six hyperparameters through five experiments:

(H1) Learning Rate/ LoRa Rank (W&B). The learning rate and (Q)LoRa rank dimension are crucial for training stability. We jointly test the learning rate $\{1e{-}5, 2e{-}5, 4e{-}5, 8e{-}5\}$ and the LoRa rank dimension $\{16, 32, 64, 128\}$, resulting in a total of 16 experiments. We observe that smaller learning rates and LoRa ranks lead to higher evaluation accuracy.

(H2) DPO Beta (W&B). The DPO beta hyperparameter controls the influence of the reference model on the policy model. We test the range $[0.05, 0.5]$ in increments of 0.05 and observe that lower values lead to better evaluation accuracy.

(H3) Label Smoothing (W&B). Label smoothing (Chowdhury et al., 2024) promises to improve DPO training when using noisy preference data. We test label smoothing in the range $[0.0, 0.4]$ in increments of 0.1. We find that no label smoothing is the best choice.

(H4) DPO Loss (W&B). Other variants of the DPO (Rafailov et al., 2023) loss exist. We test on two variants DPO: RSO (Liu et al., 2024) which uses a Hinge loss and IPO (Azar et al., 2023). We find that IPO outperforms the other two.

(H5) Data Filtering (W&B). Finally, we test different filtering strategies described in Section 2. We use global threshold filtering with $\lambda = \{3, 4\}$ and local tolerance filtering with $\lambda = \{0, 1\}$. Evaluation is performed on the unfiltered validation set, so as not to bias the results. We observe little difference in the evaluation accuracy between the baseline and filtering data, despite training on fewer examples.

Overall, we find that low learning rates, LoRa rank dimensions, DPO beta values, no label smoothing, and the IPO loss function result in the best validation accuracy. We also find that the local tolerance filtering with $\lambda = 0$ is superior to the other filtering methods.

4.2 Full Hyperparameter Tuning

We now use the full dataset to train five models using optimal hyperparameters identified from the previous five experiments, the results of which can be see in (Table 1: H1 to H5). We then combine these optimal hyperparameters, to two mixed models (Table 1: M1 and M2), one with optimal data filtering and one with none, as there was little peformance difference. Each configuration is trained for one epoch and evaluated on 640 validation samples during training, and subsequently on the entire validation set. The experiment logs are accessible at W&B. We select the best peforming configuration (A4) for our final model which is re-trained on the combined training and validation sets.

5 MCQ Specialisation & Quantisation

To prepare our model for scientific question answering, we will finetune it on `SciQ` (Johannes Welbl, 2017). For generating single letter answers, we will include examples in the prompt to set the expected format. We will then follow the `EleutherAI` approach (Gao et al., 2023) by collecting the probability of each full answer based on next token logits.

To reduce the model size, we will merge the LoRA adapters with the base model and quantise the model using `GPTQ` (Frantar et al., 2023). We will then evaluate the model on a validation split of the `SciQ` dataset and standard benchmarks such as `HumanEval`, `MBPP` and `MMLU`, in order to assess the performance loss from quantisation.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, and Jyoti Aneja. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. 2024. Provably Robust DPO: Aligning Language Models with Noisy Feedback.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers.

Leo Gao, Jonathan Tow, Baber Abbasi, and Biderman. 2023. A Framework for Few-Shot Language Model Evaluation.

HuggingFace. HuggingFace TRL.

Matt Gardner Johannes Welbl, Nelson F. Liu. 2017. Crowdsourcing Multiple Choice Science Questions.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. 2024. Statistical Rejection Sampling Improves Preference Optimization.

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.

A Appendix

A.1 Annotated Example

```
{
    "course_id": 1,
    "question_id": 1,
    "question": "...",
    "question_options": "...",

    "A" : "...",
    "B" : "...",

    "ranking_criteria": {
        "overall": "A",
        "correctness": "B",
        "relevance": "AB",
        "clarity": "None",
        "completeness": "A",
        "other": "Conciseness: B"
    }
},
```

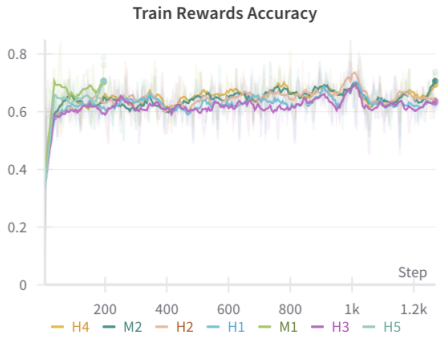
Listing 1: Annotated Example

A.2 Base Training Setup

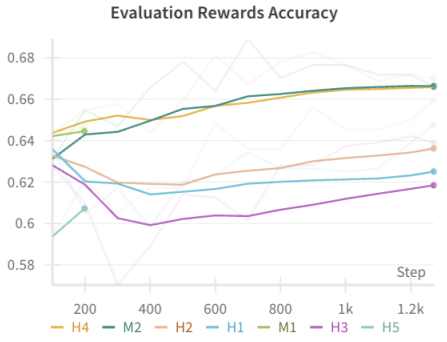
Table 2: **Base Training Setup.** The table details the default hyperparameters used for training the model. The hyperparameters marked in italics are ablated in our experiments, while the rest are kept constant.

Category	Hyperparameter	Default
Optimiser	Name	AdamW
	Learning Rate	$4e-5$
	Weight Decay	0.0
Scheduler	Strategy	Cosine
	Warmup Ratio	0.1
PEFT	Rank	32
	Alpha	16
Dataset	Batch Size	16
	Filtering Strategy	None
Objective	*Beta*	0.1
	Loss Type	Sigmoid
	Smoothing	0.1

A.3 Training and Evaluation Accuracies



(a) Training Accuracy



(b) Evaluation Accuracy

Figure 1: **Training and Evaluation Accuracies.** The figures show the training and evaluation accuracies over the course of training.