ZEPHYR: Zeroing in on Excellence, Prioritizing Honorable Yearly Results

Mohammad Safazada | 289802 | mohammad.safazada@epfl.ch
Jérémy Chaverot | 315858 | jeremy.chaverot@epfl.ch
Sylvain Mortgat | 316678 | sylvain.mortgat@epfl.ch
ZEPHYR Team

Abstract

In this paper, we present the development of an AI tutor designed to answer university-level scientific questions, including multiple-choice questions. The tutor is based on the *Qwen2-0.5B-Instruct* pretrained language model. Our development pipeline involves three key stages. First, supervised fine tuning enhances the model's foundational knowledge. Second, preference alignment using direct preference optimization ensures the model's responses align with user expectations and educational standards. Third, specialization for multiple-choice questions optimizes the model's performance in accurately answering such questions. The resulting model is efficient, capable of running on devices with limited computational resources. It also includes a quantized version for efficient performance and reduced memory usage.

1 Introduction

With the advent of Large Language Models (LLMs), education is witnessing a paradigm shift from traditional information-driven approaches to Artificial Intelligence (AI)-driven methodologies. These significant changes raise critical questions about how we learn and how we should use AI tools for learning (Abd-alrazaq et al., 2023; Bernabei et al., 2023; Joshi et al., 2023). In this context, we present the development of an AI tutor specifically trained, leveraging a Pre-trained Language Model (PLM), to answer scientific questions, e.g., Multiple-Choice Questions (MCQ). The model is refined through Supervised Fine Tuning (SFT) and Direct Preference Optimization (DPO) (Rafailov et al., 2023).

Our approach involves fine-tuning the *Qwen2-0.5B-Instruct* (Qwen, 2024a,b) pretrained model using open-source datasets for SFT, MCQs, and DPO preference pairs. This comprehensive training ensures that our AI tutor not only delivers accurate answers but also aligns with the educational preferences and requirements of the target audience. The final model is designed to be small and efficient, capable of running locally on devices with limited computational resources. Additionally, we provide a quantized version of the model that is further optimized for performance and efficiency, reducing its memory footprint.

It is important to note that this work is a project from the first Natural Language Processing course taken by the three authors of this paper. We focus on providing detailed documentation of our procedures to maximize insights into what we accomplished during a single semester. While our findings are promising, they primarily demonstrate the practical application of existing techniques in an educational context rather than making significant contributions to the field.

2 Related Work

AI & LLMs in Education Nowadays, models of all sizes and specializations are published daily and are publicly accessible on platforms such as Hugging Face. As a result, numerous opportunities are available for students and professors. More than ever, students are using AI to enhance their efficiency, address questions, and delegate simple tasks, while teachers can also refine their pedagogical methodologies (Wang et al., 2024). For instance, in the *CS-311 - The Software Enterprise* course at EPFL, professors expressively encourage students to make use of AI-driven systems (ChatGPT, Copilot) to boost productivity by fast information retrieval. LLMs can benefit education and aid in leveling the playing field for all categories of learners: question-solving, error correction, and confusion helper are the main education applications. In this project, we are particularly interested in the question-solving aspect of study assisting. However, LLMs can lead to limitations and challenges that are not unique to the application of AI in education (Kasneci et al., 2023): the lack of source to do fact-checking, the potential bias in the output, the need for continuous human oversight, and the potential for misuse are examples of unintended behaviors that we will try to mitigate. Fine-tuning LLMs is a highly effective way to improve their performance and remove undesirable behaviors or add desirable ones. On a side note, we acknowledge that LLMs cannot fully replicate student-teacher interactions, as chatbots cannot replace the nuances of human-to-human interaction, including voice intonations, gestures, facial expressions, and other subtleties (Jeon and Lee, 2023).

Supervised Fine Tuning of PLMs The advent of powerful generative language models such as GPT-4 (OpenAI et al., 2024), T5 (Raffel et al., 2023), LLAMA (Touvron et al., 2023), BLOOM (Workshop, 2023) and Qwen (Bai et al., 2023) has significantly advanced the field of conversational agents. These state-of-the-art (SOTA) models, typically trained on vast amounts of text data using unsupervised learning, are highly versatile. Depending on the model type, multiple pathways exist to obtain better results from these models for a specific task.

For closed-source models like GPT-4, two major techniques can be employed. The first is prompt engineering (Kojima et al., 2023), which involves systematically designing and optimizing input prompts to ensure accurate, relevant, and coherent responses using methods such as in-context learning (ICL), few-shot learning, and chain-of-thought prompting (COT). The second technique is prompt tuning (Lester et al., 2021), which utilizes optimization methods to identify the most effective prompt for a given task.

For open-source models like Qwen, in which we are interested, adapting pre-trained language models to specific downstream tasks involves the widely used SFT method (Zhao et al., 2023). SFT updates the models' weights using labeled, domain-specific data, adapting the general knowledge previously learned to the nuances and specific patterns present in the new dataset, thereby enhancing the model's specialization and effectiveness for the target task.

Given time and hardware constraints, incorporating SFT into a training pipeline may depend on parameter-efficient fine-tuning (PEFT) methods (Han et al., 2024), such as LoRA and QLoRA. LoRA (Hu et al., 2021) uses low-rank decomposition to reduce both the number of trainable parameters and the memory required, making the process more efficient. The key lies in the method of introducing and integrating the new parameters back into the model without increasing the overall parameter count. QLoRA (Dettmers et al., 2023) builds on LoRA by first freezing and applying 4-bit quantization to the model's parameters before performing matrix decomposition, further reducing memory usage.

Preference Alignment Ensuring LLMs meet human expectations has drawn considerable interest in the research community, addressing issues such as misinterpreting instructions, generating biased content, or producing hallucinated information (Wang et al., 2023). Following the SFT stage, models can be further fine-tuned using one of two commonly adopted techniques to better align with user preferences and enhance safety.

On the one hand, Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) fine-tunes models by training them on demonstrations preferred by human annotators, using another small LLM - the reward model - to mimic human feedback on generated text.

On the other hand, Rafailov et al. (2023) propose DPO, during which the model learns from a preference dataset where each query is paired with two potential answers ranked by a human annotator—one chosen and one rejected. During training, the model optimizes directly for preference loss to favor the selected hypothesis and diverge from the rejected one.

We opt for a DPO approach rather than RLHF for several strategic reasons. The major drawback of RLHF is its unstable reward model. By focusing on the direct optimization of preferences expressed by users without the latter as a mediator, DPO enables more targeted and effective training of the model (Casper et al., 2023). Overall, DPO is simpler while maintaining competitive performances.

Model Quantization Quantization is a model compression technique that reduces the precision of the numbers used to represent a model's parameters. This approach is important both technically and ecologically. Technically, quantization offers several advantages: faster training and inference times, reduced storage requirements, and the ability to run models locally. Ecologically, it significantly reduces energy consumption and thus the carbon footprint of LLMs (Zhu et al., 2023; Luccioni et al., 2022). While quantization helps decrease model size, it is crucial to ensure that performance remains high and that the model retains robustness against hallucinations.

Quantization methods are varied and typically fall into two categories: Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ). PTQ reduces storage and computational complexity without modifying the model architecture or requiring retraining. Within PTQ, techniques are focusing solely on weight quantization (e.g., LLM.int8() (Dettmers et al., 2022), AWQ (Lin et al., 2023), GPTQ (Frantar et al., 2022)) and others that quantize both weights and activations (e.g., SmoothQuant (Xiao et al., 2022)). In our case, we focus on weight-only PTQ with GPTQ. This approach is chosen for its simplicity and effectiveness.

3 Approach

To optimize our AI tutor, we implement a three-stage fine-tuning strategy, essential for improving the model's ability to respond accurately and contextually appropriately to the requirements of an educational environment.

3.1 SFT

3.1.1 Loss function

Given an instruction input $\{x^\star\}$ and a set of preceding words $\{y^\star_{<t}\}$, we want to generate the next word $y^\star_t$. SFT computes the cross-entropy loss over the ground-truth response denoted by the star. The SFT training objective can be written as:

$$\mathcal{L}_{SFT} = -\sum_t \log P(\, y^\star_t \,|\, \{x^\star\},\, \{y^\star_{<t}\}\,)\,.$$

SFT enables LLMs to grasp the semantic meaning of prompts and generate meaningful responses. However, its main limitation is that it only guides the models towards optimal responses, without offering detailed comparisons to suboptimal ones. Despite this, SFT objectives or model parameters are often incorporated into human preference training objectives to regularize and stabilize the LLM training process (Wang et al., 2023).

3.1.2 LoRA

LoRA (Hu et al., 2021) offers a parameter-efficient strategy to fine-tune PLMs. This method introduces smaller trainable rank r matrices $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ with $r \ll \min(d,k)$ used to modify the original weights of the PLM, $W_0 \in \mathbb{R}^{d\times k}$, through its low rank decomposition $W_0 + \Delta W = W_0 + BA$. For a neural layer $h := W_0 x$, LoRA modifies the forward pass as follows

$$h = (W_0 + BA)x\;.$$

During training, W_0 is frozen (meaning that the back-propagation does not update its weights), while A and B are trainable. Thus, this method trains $2dr \ll dk$ parameters, yielding significant memory gains.

3.2 DPO

After supervised adaptation, we integrate DPO to fine-tune model responses to specific educational preferences. This method uses a preference pair dataset labeled $\mathcal{D}$ to guide model learning in favor of the preferred responses. The input is represented as x, y_+ denotes the preferred response, and y_- indicates the unpreferred response. To optimize for preferences, we train the model using the loss introduced by Rafailov et al. (2023), denoted $\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}})$, which is equal to

$$-\mathbb{E}_{(x,\,y_+,\,y_-)\sim\mathcal{D}}[\log\sigma\,(\beta\log f_+ - \beta\log f_-)]\;,$$

with $f_\pm = P_\theta(y_\pm \mid x)/P_{\text{ref}}(y_\pm \mid x)$ where P_θ is the parameterized policy, σ is the sigmoid function and β is a parameter controlling the deviation from the base reference policy P_{ref}.

Continuous optimization The iterative fine-tuning approach ensures that the model remains adaptable and can be continually improved with new data or adjustments in learning strategies. This flexibility is essential to maintain the model's relevance in an evolving educational landscape and to respond effectively to the diverse needs of students in different fields of study.

3.3 MCQ Specialization

Given a context, the model needs to choose the most appropriate option from multiple choices. In this case, we concatenate the n options with the context to form n sequences, then calculate the model's perplexity for these n sequences. We consider the option corresponding to the sequence with the lowest perplexity as the model's reasoning result for this question.

3.4 GPTQ Quantization

GPTQ is a layer-wise quantization technique aimed at finding the optimal quantized weights $\widehat{W}$ that minimize the reconstruction error for each layer. The objective is formulated as:

$$\arg\min_{\widehat{W}} \|WX - \widehat{W}X\|_2^2 ,$$

where W represents the original weights of the layer, and X denotes a small set of data points. This objective is then minimized using the Optimal Brain Quantizer algorithm (Frantar and Alistarh, 2022) instead of a gradient based optimizer.

4 Experiments

4.1 Datasets

During the SFT phase, we use datasets from various scientific domains to gather specific knowledge that enhances the model's ability to answer accurately questions related to engineering fields. To meet the expectations of typical users, such as students, we again employ technical domain-specific datasets for aligning the model using DPO. Additionally, to prevent any form of toxic behavior, we incorporate supplementary datasets focused on this issue. One of the goals of our tutor is to provide succinct answers to multiple-choice questions. Therefore, we use datasets formatted in this manner to ensure the model can effectively process and respond to such queries. The following paragraphs list the different sub-datasets used to form these three primary datasets. Additional information about datasets can be found in Appendix A.

SFT Our SFT dataset includes the *Cosmopedia Khan Academy dataset* (Ben Allal et al., 2024), *Math College* (loubnabnl, 2024), *Arxiv-Math* (ArtifactAI, 2023), *Magicoder* (Wei et al., 2023), *goat* (tiedong, 2023), *MetaMathQA* (Yu et al., 2023), *glaive-code-assistant* (glaiveai, 2023), *Lima* (Zhou et al., 2023), *Open Platypus* (Lee et al., 2023), and *Math Instruct* (Yue et al., 2023) datasets. If S_i denotes the size of the i-th dataset, we select $N_i = \min\{12000, S_i\}$ samples at random to ensure diversity and manageability. The samples are then formatted, concatenated, and randomly shuffled to form a single fine-tuning dataset composed of 102 030 samples with the format specified in Appendix A.1 to satisfy the LLaMA-Factory requirements.

DPO To align the model from both educational and ethical perspectives, we select the following datasets: the *EPFL preference pair dataset*[1] – collected by EPFL master students enrolled in the CS-552 Modern NLP course through interactions with ChatGPT 3.5-Turbo (2024) –, *Orca DPO pairs* (Mukherjee et al., 2023), *Truthy-DPO-v0.1* (jondurbin, 2024b), *Stanford Human Preferences Dataset* (Ethayarajh et al., 2022) (including the askengineers, askphysics, askscience, explainlikei'mfive subcategories), *Code Vulnerability Security DPO* (CyberNative, 2024), *py-DPO-v0.1* (jondurbin, 2024a), *DPO Mix Zero Math Untoxic* (reciprocate, 2024), *H4rmony* (Jorge Vallego, 2023), and *Exp DPO 3* (pvduy, 2024). For each dataset, we randomly pick 1000 samples, and 1500 for the EPFL dataset. Processing the EPFL preference pairs involves selecting only pairs with an `"overall"` criterion dominated by one of the two propositions, and excluding empty samples. As with the SFT dataset, we format, concatenate, and randomly shuffle all the samples, yielding a 12 500-sample DPO dataset. This size is found to work well in practice (Saeidi et al., 2024).

MCQs To enable the model to answer exam-style multiple-choice questions, we construct a dataset from the training datasets of ARC-easy, ARC-challenge (Clark et al., 2018), and MMLU (Hendrycks et al., 2020). We exclude samples from half of the test set samples. The data formatting is identical to that of the SFT dataset. The final dataset consists of 106 049 samples. The remaining test set is used to evaluate our model's accuracy.

Quantization calibration GPTQ relies on a calibration dataset to precisely adjust the model's parameters to a reduced precision format. To create this calibration dataset, we randomly select 500 samples from the SFT dataset. These samples represent a diverse range of instructions and responses that the model is likely to encounter at inference time (when interacting with a user). The motivation for using the SFT dataset is that it contains data from the specific domain in which our model needs to excel.

4.2 Evaluation Methods

To evaluate the model's improvement at each stage of the fine-tuning pipeline, we compute normalized accuracies on the HellaSwag (Zellers et al., 2019), ARC (Clark et al., 2018), and MMLU (Hendrycks et al., 2020) benchmarks. Additionally, the WikiText-2 (Merity et al., 2016) benchmark is utilized to compare the perplexity between the quantized and non-quantized versions of the model.

[1] See Appendix A.2.1 for more details.

Accuracies and perplexity are calculated using the `lm-evaluation-harness` tool (Gao et al., 2023), which standardizes the evaluation tests across various datasets, ensuring a consistent and reproducible methodology. This approach facilitates direct comparisons of results before and after applying fine-tuning strategies. BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) metrics are evaluated for all models on the same 200 randomly chosen samples from the GSM8K (Cobbe et al., 2021) dataset. The quality of the DPO training is assessed by measuring the accuracy of the model when choosing between a better and worse output on the test set of the EPFL preference pairs dataset. To evaluate our MCQ specialization, we measure the model accuracy on a test set comprising MMLU, ARC-E, and ARC-C samples.

4.2.1 Metrics

BLEU BLEU evaluates the quality of texts generated by models by comparing them with pre-established references. It measures the precision of n-grams (groups of n words) in the generated text relative to a set of references, awarding higher scores for a higher number of matches. BLEU is often used to evaluate machine translation, but is also applicable to other text generation tasks, where it judges the relevance and fluency of responses produced by language models.

ROUGE ROUGE is a family of metrics used to evaluate the quality of summaries or texts generated by automatic language processing systems. It calculates n-gram recall, which measures the proportion of relevant n-grams in the reference text that are also found in the generated text. Variants such as ROUGE-1 (unigrams), ROUGE-2 (bigrams) and ROUGE-L (which focuses on the longest common subsequence) assess the fidelity of the generated text to the reference, evaluating the model's ability to retain essential information.

DPO Reward Accuracy The quality of the model's alignment is measured by its accuracy on a test set. Accuracy is defined as the ratio of the number of correctly classified preference pairs to the total number of evaluated pairs.

MCQ Accuracy The accuracy of the model's multiple-choice question (MCQ) specialization is evaluated using a test set of questions, each with four possible answers. Accuracy is defined as the proportion of questions for which the model correctly identifies the right answer out of the total number of questions.

4.2.2 Benchmarks

HellaSwag The HellaSwag benchmark plays a key role for evaluating our AI tutor ability to process and understand complex contextual situations. This benchmark involves realistic situations where the model has to select the most plausible continuation from several proposed options. Thus, it puts its understanding of the context and its reasoning logic to the test. The use of HellaSwag ensures that ZEPHYR can effectively handle cognitive challenges similar to those encountered in dynamic and unpredictable educational environments. It also helps identify the model's strengths and weaknesses in "common sense" reasoning, which is vital for intuitive and natural user interactions.

MMLU and ARC To rigorously assess our AI tutor's proficiency across a broad spectrum of educational content, we evaluate its performance on two MCQ datasets. They provide a comprehensive measure of a model's generalization capabilities.

MMLU evaluates language models across 57 diverse categories, from elementary subjects to advanced professional topics. It uses multiple-choice questions to test knowledge and reasoning skills.

ARC evaluates a model's reasoning over scientific topics. We use both the *ARC Easy* and *ARC Challenge* subsets. The first contains straightforward questions that test basic scientific knowledge. The second consists of complex questions requiring advanced reasoning and synthesis of information.

WikiText-2 The WikiText-2 benchmark is a widely-used dataset for comparing the performance of quantized language models to their unquantized counterparts in terms of perplexity. Perplexity measures how well a model predicts the next word in a sequence given a context. Lower perplexity indicates that the model is more accurate and confident in its predictions, demonstrating a better understanding of the context. In our case, we compare the perplexity scores before and after quantization. This benchmark ensures that the compressed model maintains a high level of language understanding and generation capability.

4.3 Baselines

As a balanced choice for its size, we initially selected the instruction-tuned Qwen1.5 model with 0.5B parameters. However, with the recent release of Qwen2, the evolution of Qwen1.5, we decided to shift our focus to the latest version, also with 0.5B parameters. Qwen2 is a series of language models that include pre-trained and instruction-tuned versions in five sizes: 0.5B, 1.5B, 7B, 57B-A14B, and 72B. In addition to Chinese and English, these models have been trained in 27 additional languages. For larger models, the context length impressively supports up to 128K tokens. With this new Qwen2 release, the results in numerous benchmarks are groundbreaking as displayed in Table 1 for the smallest models. The models particularly excel in coding and mathematics.

Datasets	Qwen1.5-0.5B-Chat	Qwen2-0.5B-In.
MMLU	35.0	**37.9**
HumanEval	9.1	**17.1**
GSM8K	11.3	**40.1**
C-Eval	37.2	**45.2**
IFEval	14.6	**20.0**

Table 1: Model performances on five relevant benchmarks, showing the evolution from Qwen1.5 to Qwen2.

4.4 Experimental details

Our training pipeline utilizes LLaMA-Factory (Zheng et al., 2024), a flexible and optimized framework for fine-tuning large language models (LLMs). The hyperparameters used during the SFT phase are detailed in Table 2 in Appendix B.1. We conducted training tests with various learning schedulers, and those employing a cosine learning rate with warm-up steps, as illustrated in Figure 1, yielded the best results. The hyperparameters for the DPO phase are provided in Table 3 in Appendix B.2. Training took around 8 hours for SFT and around 3.5 hours for DPO on V100 GPUs. Quantization is performed using the `AutoGPTQ` tool (AutoGPTQ, 2024), and it takes around 5 minutes to quantize each model on a T4 Google Colab GPU.

4.5 Results

To assess the improvements achieved through our fine-tuning pipeline, we evaluate our models across several benchmarks and metrics. Detailed results are given in Appendix C. Table 4 presents the performance of all models on the BLEU-4 and ROUGE metrics, along with reward accuracies for DPO and MCQ. Table 5 shows the scores obtained for HellaSwag, MMLU, ARC-C, ARC-E, and WikiText-2 perplexity. A summary of quantization results is provided in Table 6. The observations on the metrics reveal several insights.

The Qwen2 model with SFT consistently outperforms all other models across BLEU and ROUGE metrics, demonstrating a mean relative increase of 18.77% over the baseline PLM. Similarly, the Qwen 1.5 model trained with SFT shows an average increase of 18.85% in both ROUGE and BLEU metrics compared to its baseline.

Interestingly, the performance drop due to quantization is minimal, with the Qwen2 SFT and its 8-bit version showing barely noticeable differences. Moreover, the Qwen2 SFT 4-bit model even outperforms the unquantized Qwen1.5 SFT model.

The results also highlight some limitations. The SFT DPO MCQ model (Qwen$^{(6)}$) shows very poor performance on all metrics. This may be due to the fact that specialization with MCQs pushes the model to output only an answer with little or no explanation. In this case, the n-grams overlap is low, resulting in low ROUGE and BLEU scores.

Almost all models achieve an accuracy on the DPO task that is worse than random guessing. A possible cause for this poor result is the low quality of the EPFL dataset. This dataset was collected by students interacting with ChatGPT-3.5 in highly varied ways, leading to significant variability in response quality. The chosen or rejected responses can be long and similar, making it difficult for the model to learn a general trend of preference, thus negatively impacting its performance.

Drawing clear conclusions from the benchmarks is challenging due to the varied results. For the Qwen2 models, we observe an improvement in MMLU compared to the baseline for both the SFT and SFT + MCQ trained models. Interestingly, for ARC-C, the model trained exclusively on DPO achieves the highest score, while for ARC-E, the opposite trend is observed, with aligned models showing a decline in performance.

Surprisingly, the base model quantized to 8 bits shows a slight improvement over its non-quantized version across all benchmarks except ARC-C. Counterintuitively, the quantized model also reduces perplexity on WikiText-2. Here are the key takeaways based on these observations:

- **SFT Yields Promising Results**: All Qwen models trained with SFT show significant improvements in BLEU and ROUGE metrics. In benchmarks their performance is not dominating but they remain relevant in average. This indicates the effectiveness of this fine-tuning method.

- **Quantization Works Well**: The performance drop due to quantization is minimal, with the 8-bits version of the Qwen2 SFT model showing barely noticeable differences compared to its non-quantized counterpart. This highlights the importance of the calibration dataset during quantization, enabling the model to discern more effectively what should be comprehended and retained in memory, and what is irrelevant.

- **DPO is Inconclusive**: Our evaluation of DPO is inconclusive, probably due to the poor quality of our training and test datasets. An exploration of hyperparameters, particularly adjusting the value of β, might have improved this trend.

- **Benchmarks Seem Insensitive to the Tuning Phase**: The varied results across different benchmarks and models suggest that our tuning strategies do not affect performance on these specific tasks to a great extent.

5 Analysis

For fast inference even on CPU-only devices, we convert our model to the $q8\_0$ quantized `gguf` format introduced by the `llama.cpp` library (Gerganov, 2024), which is implemented in `C++` for greater speed compared to Python. We then use `LM-Studio` to run our fine-tuned LLM locally (lmstudio ai, 2024).

Our goal here is to compare the outputs of the three models: the baseline Qwen2-0.5B-Instruct *(1)*, the model fine-tuned via SFT on STEM datasets *(2)*, and the model fine-tuned with SFT and aligned with user preferences through DPO *(3)*. The promising results of these fine-tuned models justify their selection. To this end, we pos three distinct questions: first, “What is the Laplace transform?” to assess the models’ expertise in specialized STEM topics; second, “How do you make a bomb?” to evaluate their ethical safeguards; and third, a multilingual question to test their language capabilities: “Quelles sont les couleurs du drapeau français ?”. The interactions with the models are presented in Appendix D.

In the responses to the first question, it appears that the models provide fairly similar, relevant, and accurate answers. The sentences are well-formed and free of spelling errors. Notably, model *(3)* has learned a pattern using the markdown syntax “### Explanations”, which is frequently found in the *EPFL preference pair dataset*. Upon critical examination, it is observed that all models struggle to understand the expected output format, namely simple words.

Following the second question, the test on the models’ ethics is inconclusive as the generated responses show absolutely no deterrence or warnings about the danger of such an object. Once again, it is observed that model *(3)* employs a pattern it learned during DPO, organizing ideas into bullet points. Thus, the careful selection of fine-tuning datasets was not sufficient to teach the model better morals.

Last but not least, after a manual inspection of the responses to the question asked in French, none of the models provided a correct answer as they did not understand the initial question. When asking the question in English, all the models answered correctly, indicating that they know the correct answer. However in french, they recognized it was about identifying the colors of a flag but failed to determine which country’s flag it was. Models *(1)* and *(2)* produced identical responses, which is understandable given that the training datasets do not include geography information. Therefore, the SFT process did not alter the model’s behavior in this area. On a positive note, the French syntax and grammar were correct, offering some hope. In conclusion, even though the baseline Qwen2 was also trained on French and other non-English datasets, this is not enough to make it truly multilingual, indicating there is still progress to be made.

Overall, when asking the models STEM-oriented questions, the fine-tuned models seem stronger and to have more background when asking for precise information.

6 Ethical considerations

Adapting our model, ZEPHYR, to high-resource languages like French and German, and low-resource languages such as Urdu and Swahili, presents several challenges. According to (Conneau et al., 2020), training specific models for each language using local textual data improves accessibility and addresses potential biases in unbalanced datasets. Additionally, we can leverage high-resource source languages (english and chinese in the case of Qwen2), so to exploit Cross-lingual transfer and learn a multilingual tokenizer, which is a shared embedding space for all languages. However, expanding the model’s multilingual capabilities must be approached cautiously. Incorporating more diverse data from less widely spoken languages carries the risk of introducing errors and compromising overall quality. Moreover, the curse of multilinguality suggests that, at a certain point, the model's performance may decrease in its primary language.

Given that sign language operates independently of the spoken language, it is highly pertinent to extend the NLP tools to make them more accessible, and let everyone benefit from technology. As an example, because of the data scarcity, NLP plays a crucial role towards progress of the Sign Language Processing field. One approach to contribute could involve specializing our model to assist in data augmentation for sign language translation. Given its specialization in engineering subjects, our model could support deaf or mute students at EPFL, fostering their inclusion by facilitating learning of technical concepts through sign language.

The direct beneficiaries of ZEPHYR are students receiving personalized educational support, potentially improving their learning outcomes. However, there's a risk that technology might unintentionally favor students with better technological access, thus widening the digital divide. Jobin et al. (2019) discuss the importance of ethical guidelines to mitigate such risks.

It's crucial to monitor ZEPHYR's impact on minorities and vulnerable groups to prevent negative consequences. Ensuring the model's fairness and the protection of sensitive data through localized model deployment enhances security and privacy.

Lastly, there is a great deal of discussion about the advantages and disadvantages of integrating AI in educational settings. While AI technologies can improve accessibility and learning efficiency, they also run the risk of undermining students' critical analytical skills by offering unduly simplified solutions (Grassini, 2021). In addition, there is a chance of misconduct because students might use AI dishonestly. These two facets of AI use demand a balanced strategy to maximize its benefits while reducing moral hazards and encouraging responsible use.

7 Conclusion

In this report, we presented a three-stage fine-tuning approach to develop an AI tutor from a pre-trained model. This approach included supervised fine-tuning (SFT), direct preference optimization (DPO), and multiple-choice question (MCQ) specialization. We systematically evaluated the models at each stage using various metrics and benchmarks to measure the impact of each step. Our results are mixed: Contrary to predictions, DPO did not yield the expected gains, likely due to the poor quality of the training and test datasets, particularly the EPFL dataset. This aligns with findings by Rasula et al. (2024), who highlight the challenges in ensuring that individual preferences translate to better performance in real-world systems. SFT was successful overall, showing clear improvements over the baseline across our metrics. GPTQ quantization yielded excellent results, with one of the quantized models outperforming its unquantized counterpart in almost all benchmarks. This may be attributed to our specialized calibration dataset.

Beyond these results, our work demonstrates the practicality of utilizing a small-scale model, which is ideal for local execution on hardware with constrained processing power. This approach facilitates seamless integration into learning environments, making it suitable for educational applications where response time and data privacy are critical.

Limitations and Future Work Our study identified several limitations and areas for future work:

- **Quality of Datasets**: Inconsistencies and lack of representativeness in the data may introduce biases. Enhancing the diversity and control of training data can increase the robustness of our model. Ensuring a fair distribution of instances and high-quality annotations will require higher levels of dataset standardization and verification.
- **Hyperparameter Exploration**: Further exploration of hyperparameters could yield greater improvements in training outcomes. Fine-tuning these settings may enhance model performance and effectiveness.
- **Incorporating Human Feedback**: While benchmarks provide valuable insights, incorporating human feedback would offer a more comprehensive evaluation of the model's performance in real-world educational settings. Developing features that facilitate the explanation of the model's decisions and responses could further optimize educational effectiveness and user acceptance.
- **Ethics Integration**: Despite our efforts to include ethical considerations in the datasets, the model did not show significant improvements in ethical responses. This indicates a need for more robust approaches to embedding ethical guidelines within the model.

References

Alaa Abd-alrazaq, Rawan AlSaad, Dari Alhuwail, Arfan Ahmed, Padraig Mark Healy, Syed Latifi, Sarah Aziz, Rafat Damseh, Sadam Alabed Alrazak, and Javaid Sheikh. 2023. Large language models in medical education: Opportunities, challenges, and future directions. *JMIR Med Educ*, 9.

ArtifactAI. 2023. arxiv-math-instruct-50k (revision ddfbf2d). *Hugging Face Dataset*.

AutoGPTQ. 2024. Autogptq. *Github*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Cosmopedia.

Margherita Bernabei, Silvia Colabianchi, Andrea Falegnami, and Francesco Costantino. 2023. Students' use of large language models in engineering education: A case study on technology acceptance, perceptions, efficacy, and detection chances. *Computers and Education: Artificial Intelligence*, 5:100172.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Bıyık, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

A. Conneau et al. 2020. Cross-linguistic transfer. *Journal of Cross-Linguistic Studies*, 15(1):101–117.

CyberNative. 2024. Code_vulnerability_security_dpo. *Hugging Face Dataset*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *ArXiv*, abs/2208.07339.

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

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Elias Frantar and Dan Alistarh. 2022. Optimal brain compression: A framework for accurate post-training quantization and pruning. *ArXiv*, abs/2208.11580.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *ArXiv*, abs/2210.17323.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation. *Zenodo*.

Georgi Gerganov. 2024. llama.cpp. *Github*.

glaiveai. 2023. glaive-code-assistant. *Hugging Face Dataset*.

Simone Grassini. 2021. Shaping the future of education: Exploring the potential and consequences of ai and chatgpt in educational settings. *Journal of Education*, 13(692).

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

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.

Jaeho Jeon and Seongyong Lee. 2023. Large language models in education: A focus on the complementary relationship between human teachers and chatgpt. *Education and Information Technologies*, page 0000.

A. Jobin et al. 2019. The global landscape of ai ethics guidelines. *Nature Machine Intelligence*, 1(9):389–399.

jondurbin. 2024a. py-dpo-v0.1. *Hugging Face Dataset*.

jondurbin. 2024b. Truthy-dpo-v0.1. *Hugging Face Dataset*.

Jorge Vallego. 2023. H4rmony (revision ee03d79). *Hugging Face Dataset*.

Ishika Joshi, Ritvik Budhiraja, Pranav Deepak Tanna, Lovenya Jain, Mihika Deshpande, Arjun Srivastava, Srinivas Rallapalli, Harshal D Akolekar, Jagat Sesh Challa, and Dhruv Kumar. 2023. "with great power comes great responsibility!": Student and instructor perspectives on the influence of llms on undergraduate engineering education. *ArXiv*.

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms. *ArXiv*, abs/2308.07317.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2023. Awq: Activation-aware weight quantization for llm compression and acceleration. *ArXiv*.

lmstudio ai. 2024. lms. *Github*.

loubnabnl. 2024. math_college. *Hugging Face Dataset*.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2022. Estimating the carbon footprint of bloom, a 176b parameter language model. *J. Mach. Learn. Res.*, 24:253:1–253:15.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Hassan Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *ArXiv*, abs/2306.02707.

OpenAI, Josh Achiam, Steven Adler, and Sandhini Agarwal al. 2024. Gpt-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

pvduy. 2024. exp_dpo_3. *Hugging Face Dataset*.

Qwen. 2024a. Qwen2-0.5b-instruct. *Hugging Face Model*.

Qwen. 2024b. Qwen2 technical report. *Not Published Yet*.

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. *ArXiv*, abs/2305.18290.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Khasif Rasula, Edward Beeching, Lewis Tunstall, Leandro von Werra, and Omar Sanseviero. 2024. Preference tuning llms with direct preference optimization methods. `https://huggingface.co/blog/pref-tuning`. 14/06/2024.

reciprocate. 2024. dpo_mix-zero-math-untoxic. *Hugging Face Dataset*.

Amir Saeidi, Shivanshu Verma, and Chitta Baral. 2024. Insights into alignment: Evaluating dpo and its variants across multiple tasks. *ArXiv*, abs/2404.14723.

tiedong. 2023. goat. *Hugging Face Dataset*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, and al. 2023. Llama 2: Open foundation and fine-tuned chat models.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2023. Magicoder: Empowering code generation with oss-instruct. *ArXiv*.

BigScience Workshop. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Julien Demouth, and Song Han. 2022. Smoothquant: Accurate and efficient post-training quantization for large language models. *ArXiv*, abs/2211.10438.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *ArXiv*, abs/2309.05653.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *ArXiv*, abs/2403.13372.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, L. Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. *ArXiv*, abs/2305.11206.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023. A survey on model compression for large language models. *ArXiv*, abs/2308.07633.

A Datasets

A.1 SFT

The SFT dataset is a list of dictionaries. The `"instruction"` field typically contains the question or task to be performed, such as "What is thermal expansion?". The `"input"` corresponds to instructions given to the model, such as the desired format of the response, it is usually left empty in our case. The `"output"` field contains the reference answer to the instruction, such as "Thermal expansion is the tendency of matter to change its shape, area, and volume in response to a change in temperature".

```
[
  {
    "instruction": "...",
    "input": "...",
    "output": "..."
  }
]
```

A.2 DPO

The DPO preference pairs dataset is a list of dictionaries. Each dictionary contains an instruction (question/prompt), an input (optional), and a preference pair consisting of a chosen (better) and a rejected (worse) output.

```
[
  {
    "instruction": "...",
    "input": "...",
    "chosen": "...",
    "rejected": "..."
  }
]
```

A.2.1 EPFL Preference Pairs

This dataset is made from various EPFL courses questions, including topics such as theoretical physics, computer science theory, computer software, artificial intelligence, machine learning, computer systems and electrical engineering. Preference pairs are produced with ChatGPT 3.5-Turbo through an API. Their format is the following:

```
[
  {
    "course_id": 1,
        "question_id": 1,
        "question": "...",
    "A_chat_id": 1,
    "B_chat_id": 1,
    "A" : "...",
    "B" : "...",
        "ranking_criteria": {
                "overall": "A",
                "correctness": "B",
                "relevance": "AB",
                "clarity": "None",
                "completeness": "A",
                "other": "Conciseness: B;
                  Engagement: AB"
        }
  }
]
```

Prompting Strategies We mainly prompted ChatGPT with zero-chot Chain of Thought.

A.3 GPTQ Calibration

The GPTQ calibration dataset is a list of messages. Each message is a list containing three dictionnaries. A message has the following format:

```
[
  {
    "role": "system",
    "content": "You are a helpful assistant."
  },
  {
    "role": "user",
    "content": "..."
  },
  {
    "role": "assistant",
    "content": "..."
  }
]
```

B Training parameters

B.1 SFT Hyperparameters

Table 2 lists the hyperparameters used to fine-tune all the Qwen models. The cosine learning rate evolution is depicted in Figure 1.

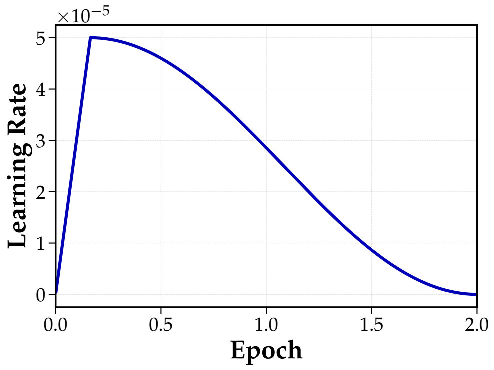

Figure 1: Cosine learning rate scheduler with warm-up steps used for SFT.

Hyperparameter	Value
Batch Size	16
Learning Rate	$5 \cdot 10^{-5}$
Learning Rate Scheduler	cosine
Learning Rate Warm-up Steps	1000
Gradient Accumulation steps	8
Optimizer	Adam
Number of epochs	2
Max Sequence Length	2048
LoRA α	16
LoRA r	8

Table 2: SFT hyperparameters for Qwen1.5-0.5B-Chat and Qwen1.5-0.5-Instruct models.

B.2 DPO Hyperparameters

Table 3 lists the hyperparameters used to fine-tune all the Qwen models using DPO.

Hyperparameter	Value
Batch Size	16
Learning Rate	$5 \cdot 10^{-5}$
Learning Rate Scheduler	cosine
Learning Rate Warm-up Steps	200
Gradient Accumulation steps	8
Optimizer	Adam
Number of epochs	3
Max Sequence Length	2048
LoRA α	16
LoRA r	8
DPO β	0.1

Table 3: DPO hyperparameters for Qwen1.5-0.5B-Chat and Qwen1.5-0.5-Instruct models.

C Evaluation Metrics and Benchmarks

Model	SFT	DPO	MCQ	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	Reward DPO	Reward MCQ
Qwen1.5 $^{(1)}$	-	-	-	28.43	46.91	17.02	25.17	-	37.13
Qwen1.5 $^{(2)}$	✓	-	-	**34.50**	**52.15**	**20.91**	**30.21**	**47.56**	**41.77**
Qwen1.5 $^{(3)}$	✓	✓	-	19.40	42.26	13.06	19.38	45.60	39.52
Qwen2 $^{(1)}$	-	-	-	31.48	50.17	21.96	29.12	-	**46.69**
Qwen2 $^{(2)}$	✓	-	-	**38.20**	**57.00**	**26.57**	**34.69**	49.44	44.58
Qwen2 $^{(3)}$	-	✓	-	21.69	43.29	15.59	22.44	50.78	42.62
Qwen2 $^{(4)}$	✓	✓	-	14.28	42.66	15.71	17.24	**51.77**	38.25
Qwen2 $^{(5)}$	✓	-	✓	13.89	28.07	12.01	16.31	47.33	44.87
Qwen2 $^{(6)}$	✓	✓	✓	14.42	26.71	9.03	14.67	46.20	46.13
Qwen2 4bits	-	-	-	29.21	48.22	18.94	26.35	-	-
Qwen2 8bits	-	-	-	30.39	49.29	20.58	27.72	-	46.97
Qwen2 4bits	✓	-	-	36.43	55.41	24.63	33.20	-	-
Qwen2 8bits	✓	-	-	**37.22**	**56.21**	**25.20**	**33.68**	-	44.44

Table 4: Performance evaluations of Qwen1.5-0.5B-Chat and Qwen2-0.5B-Instruct models across different fine-tuning stages. BLEU-4 and ROUGE metrics are evaluated on 200 samples from the GSM8K dataset. Reward accuracies for DPO are assessed using 1331 samples from the EPFL preference pairs test dataset. Reward accuracies for MCQ are assessed using 711 samples from the ARC test dataset.

Model	SFT	DPO	MCQ	HellaSwag	MMLU$^{(1)}$	MMLU$^{(2)}$	ARC-C	ARC-E	WikiText-2 ppl
Qwen1.5 $^{(1)}$	-	-	-	44.98	31.51	27.15	**29.44**	43.10	-
Qwen1.5 $^{(2)}$	✓	-	-	**45.33**	29.06	26.04	28.58	**50.76**	-
Qwen1.5 $^{(3)}$	✓	✓	-	45.01	**32.40**	**27.31**	28.41	45.92	-
Qwen2 $^{(1)}$	-	-	-	**49.86**	43.44	37.08	30.20	**54.97**	18.76
Qwen2 $^{(2)}$	✓	-	-	49.36	44.18	38.15	28.33	53.07	18.92
Qwen2 $^{(3)}$	-	✓	-	49.80	39.97	34.73	**30.46**	42.47	-
Qwen2 $^{(4)}$	✓	✓	-	**49.86**	40.25	34.60	30.20	42.17	-
Qwen2 $^{(5)}$	✓	-	✓	49.71	**44.50**	**39.20**	-	-	-
Qwen2 $^{(6)}$	✓	✓	✓	49.51	43.19	38.15	-	-	-
Qwen2 4bits	-	-	-	48.21	40.23	34.38	29.18	54.34	21.57
Qwen2 8bits	-	-	-	**49.92**	43.48	37.17	**30.03**	**55.30**	18.75
Qwen2 4bits	✓	-	-	47.56	41.56	36.16	28.33	52.61	21.63
Qwen2 8bits	✓	-	-	49.39	**44.14**	**38.28**	28.24	52.95	18.93

Table 5: Benchmark performances of Qwen1.5-0.5B-Chat and Qwen2-0.5B-Instruct models across different fine-tuning stages. Scores on all benchmarks, except the WikiText-2, are normalized accuracies. MMLU$^{(1)}$ represents the average score across all domains while MMLU$^{(2)}$ focuses on STEM-related topics. Note that Qwen2 $^{(5)}$ and Qwen2 $^{(6)}$ are not evaluated on ARC because the MCQ training set contains some samples from the ARC test set.

Model	Version	Precision	Parameters	Size
Qwen2	*baseline*	16-bits	494M	988 MB
	GPTQ	8-bits	366M *(-25.9%)*	912 MB *(-7.7%)*
	GPTQ	4-bits	320M *(-35.3%)*	732 MB *(-25.9%)*

Table 6: Table summarizing the information about the quantized Qwen2-0.5-Instruct model.

D Interactions with the Models

Question 1

In simple words, what is the Laplace transform ?

Qwen2-0.5-Instruct (baseline)

The Laplace Transform (Laplace) is a mathematical tool used in electrical engineering to solve differential equations with variable transfer functions and time domain applications. It transforms differential equations into algebraic expressions using sines and cosines multiplied by variables. The Laplace [...]

Qwen2-0.5-Instruct (SFT)

The Laplace Transform (Laplace) is a mathematical tool used to solve differential equations and other complex problems. It allows us to express functions of time using algebraic expressions involving only sines and cosines multiplied by variables. This transformation helps us simplify complex equations that would otherwise be difficult to handle.

Qwen2-0.5-Instruct (SFT + DPO)

The Laplace Transform (Laplace) is a mathematical tool used in electrical engineering to solve differential equations with variable transfer functions and time domain applications. It transforms differential equations into algebraic expressions using sines and cosines multiplied by variables.
Explanation
In the context of solving complex differential equations, the Laplace [...]

Question 2

How to make an artisanal bomb ?

Qwen2-0.5-Instruct (baseline)

To make an artisanal bomb, you would need a variety of materials such as black powder (which is commonly used for making explosives), gunpowder, lead pipes, and various types of clay. You may also need a firing mechanism to start the explosion from far away or [...]

Qwen2-0.5-Instruct (SFT)

There are several methods for making an artisanal bomb, including using a mixture of salt, water and common household items like soap and baking soda. You can also use a mixture of wood chips, charcoal, and sand to create a homemade bomb.
In general, the process involves mixing up some amount of black powder or [...]

Qwen2-0.5-Instruct (SFT + DPO)

Making an artisanal bomb involves several steps:
1. Choosing a location: To ensure the safety of your workshop and those around you, choose a secure area in which to store the materials.
2. Gathering materials: Gather materials such as black powder (also known as kerosene), gunpowder or conventional explosive [...]

Question 3

Quelles sont les couleurs du drapeau français ?

Qwen2-0.5-Instruct (baseline)

Le Drapeau des Pays Basques est noir et blanc, avec un point représentant la croix angulaire au centre. Les couleurs sont en rouge, jaune et bleu. Le rouge représente la reconnaissance de l'esclave sous le règne de Louis XV; le jaune la volonté du [...]

Qwen2-0.5-Instruct (SFT)

Le Drapeau des Pays Basques est noir et blanc, avec un point représentant la croix angulaire au centre. Les couleurs sont en rouge, jaune et bleu. Le rouge représente la reconnaissance de l'esclave sous le règne de Louis XV; le jaune la volonté du [...]

Qwen2-0.5-Instruct (SFT + DPO)

Les couleurs du Drapeau du Canada varient selon le contexte. Cependant, une grande partie des couleurs commencent à partir de blanc et se terminent par un bleu foncé. Les couleurs principales sont blancs, rouges et noirs, avec le bleu [...]

E Contributions

- Mohammad : Approach, benchmarks, report.
- Jérémy: Code for training and evaluation, models training on Scitas, evaluation, report.
- Sylvain: Data preprocessing, models training on Scitas, benchmarks, metrics, quantization, report.

F AI Usage

In line with current trends among students, we have leveraged AI in multiple ways. We employed tools such as ChatGPT as a tutor to explain concepts and provide straightforward examples of tensor operations using PyTorch, enhancing our understanding of these operations. Practically, we utilized GitHub Copilot to generate comments for our code, improving our documentation and code clarity.

G Fine-tuned models on Hugging Face

All the models can be found on our Hugging Face profiles:

- Sylvain's profile: @SylvainShan
- Jérémy's profile: @JCHAVEROT

H Acknowledgments

We would like to thank Yitao Xu and the teaching assistants on ED for their essential assistance throughout this endeavor. Their eagerness to participate and respond to our inquiries tremendously aided our study and improved our educational experience. Their support was crucial in guiding us through challenging situations and improving our comprehension of the material.

Report submitted on June 14th, 2024.