LiSoViMa - An Educational LLM Assistant

Lina SADGAL | 342075 | lina.sadgal@epfl.ch
Sofia TAOUHID | 339880 | sofia.taouhid@epfl.ch
Vincent FISZBIN | 394790 | vincent.fiszbin@epfl.ch
Matthias WYSS | 329884 | matthias.wyss@epfl.ch
LiSoViMa

Abstract

We present a systematic exploration of building an educational large language model (LLM) assistant specialized in STEM tasks. Starting from `Qwen/Qwen-0.6B-Base`, we fine-tune the model on domain-specific data and develop four specialized variants: a multiple-choice question answering (MCQA) model, a quantized MCQA model for efficient deployment, a retrieval-augmented generation (RAG) model leveraging external STEM documents, and a direct preference optimization (DPO) model for preference-based ranking. The quantized model achieves comparable accuracy to its full-precision counterpart while reducing inference cost. While the RAG model does not significantly outperform the MCQA baseline due to partially relevant retrieved content, the DPO model reliably aligns with human preferences, achieving over 81% accuracy on held-out comparisons. These results outline a practical path toward effective, efficient, and preference-aware educational LLMs for STEM domains.

1 Introduction

LLMs have demonstrated strong performance across many tasks, but their use in educational contexts, especially in STEM, demands fine-tuning, efficiency, and alignment with pedagogical goals. This work explores how to build a compact educational assistant that answers STEM-related questions with high reliability and efficiency.

We start from the open-source `Qwen/Qwen3-0.6B-Base` model (Qwen, 2025), fine-tune it on STEM data, and develop four variants: a MCQA model; a quantized MCQA model for efficient inference; RAG model accessing an external STEM corpus; and a DPO model trained on synthetic and human-labeled preference pairs. This setup enables comparisons across task-specific strategies and highlights trade-offs in accuracy, efficiency, and preference alignment.

Our results show that RAG provides limited gains over the base model, quantization maintains performance at lower cost, and DPO yields reliable preference predictions. We release our code and models to support further work on STEM-oriented educational LLMs (Appendix B).

2 Approach

To develop an effective LLM for STEM subjects, we first fine-tuned the `Qwen/Qwen3-0.6B-Base` model using question/open-answer data, aiming to familiarize it with domain-specific knowledge. We then train it for two key tasks: DPO and MCQA. For MCQA, we also experimented with a LoRA approach (Hu et al., 2022). In parallel, we ran a second experiment without the initial pre-training step, enabling a comparison of performance. Additionally, we trained a reasoning model using the SciQ support text, which, unlike other models that only output answer choices, provides both the answer and an explanation for direct assessment of reasoning.

We experimented with several quantization strategies to reduce the model's memory footprint. First, we applied Post-Training Weight Quantization using `bitsandbytes`, which compresses float16/bfloat16 weights to 8-bit or 4-bit precision. The 8-bit quantization uses a block-wise scheme inspired by LLM.int8 (Dettmers et al., 2022), where weights are divided into blocks and quantized separately to isolate outliers and preserve accuracy. For 4-bit quantization, we used Float4 and NormalFloat4, as introduced in QLoRA (Dettmers et al., 2023), which are designed to match typical weight distributions. We also applied 8-bit quantization to both weights and activations (W8A8) using `llmcompressor`, combining SmoothQuant (Xiao et al., 2024), which rescales activations to limit outliers impact during calibration, and GPTQ (Frantar et al., 2023) for linear layer quantization. We also tested weight-only 4-bit quantization (W4A16) with GPTQ. Finally, we experimented with QLoRA (Dettmers et al., 2023), which enables efficient fine-tuning by quantizing the base model to 4-bit and training only lightweight LoRA modules, offering a good trade-off between compression and accuracy.

For the RAG model, we constructed a dedicated corpus based on 28 authoritative STEM textbooks, converted into clean Markdown format, and built an embedding index using FAISS (Douze et al., 2025) with the `thenlper/gte-small` model (Xiao et al., 2023), chunking the texts into segments of 512 tokens. During training, we fine-tuned both `Qwen/Qwen3-0.6B-Base` and the SFT model with LoRA, retrieving the top $k = 5$ most relevant chunks for each query to

augment the generation process, thereby familiarizing the model with the RAG format (Lee et al., 2025).

3 Experiments

- **Data:** For the domain knowledge SFT training, we randomly sampled respectively 50,000 and 100,000 examples from `PrimeIntellect/SYNTHETIC-1-SFT-Data` (Mattern et al., 2025b). We preprocessed the data by extracting question-answer pairs from the original conversational format and removing unnecessary columns. The resulting dataset was then split into training and evaluation subsets (90%/10%) to enable model validation during fine-tuning.
  Building upon this, we further enhanced the model by incorporating existing datasets in the MCQA domain for STEM subjects. Specifically, we merged the train splits of (D1) a filtered version of `cais/mmlu` restricted to STEM subjects; (D2) `allenai/ai2_arc` (Challenge and Easy subsets); (D3) `allenai/sciq`, a science-focused MCQA dataset; and (D4) `deepmind/aqua_rat`, which contains math word problems. We standardized the datasets to ensure consistency, into a unified structure of a question, four answer choices, and the correct answer. SciQ's answers were shuffled to remove positional bias, and we randomly removed one incorrect answer from AquaRat. For MMLU, we used `mistral-large-latest` to classify examples by STEM subject and filtered out non-STEM examples, retaining 22,500 relevant entries. To identify the best dataset combination for fine-tuning, we ran several SFT experiments on the Qwen base model using LoRa on each dataset individually (Appendix C). The final dataset mix consisted of 30,000 examples (10,000 from each of SciQ, MMLU, and AquaRat) for fine-tuning, with an additional 150,600 examples for full fine-tuning. Its detailed distribution can be found in Appendix G
  Our W8A8 and W8A16 quantization methods with SmoothQuant and GPTQ require calibration to estimate appropriate scaling factors for weights and activations to minimize accuracy loss. Following standard practice in literature (Williams and Aletras, 2024), we sampled calibration data from datasets similar to those used during training/fine-tuning, ensuring calibration statistics match the data distribution encountered at inference. We used the dataset used to fine-tune our MCQA model, as well as Ultrachat200k (Ding et al., 2023), a multi-turn dialogue dataset designed to train chat models. We randomly sampled 256/512/1,024 examples, and settled with 512 since prior work (Williams and Aletras, 2024), and our own experiments(Appx K), showed diminishing returns in accuracy for larger calibration sets. For QLoRA fine-tuning, we reused the full and balanced MCQA datasets, with sample sizes 512/4,096/30,000.
  For the RAG model, we focused on building a high-quality corpus based on authoritative STEM textbooks. To achieve this, we collected 28 reference books covering various STEM subjects. However, some of the books were not properly OCR-processed. To address this issue, we used the latest Mistral OCR model, processing each book by splitting it into partial PDFs of 10 pages to avoid issues with the API. This process resulted in 28 cleaned and structured Markdown files, one for each book. Compared to raw text extraction from PDFs, Markdown preserves semantic structure (e.g., headings, lists, formulas, code), which has been shown to improve the quality of data used for LLM pretraining and instruction tuning (Chen et al., 2025). This is particularly important in the context of STEM content, where accurate parsing of mathematical expressions is essential, something that Mistral OCR handled effectively in our pipeline.
  Some of the books sourced from unofficial channels, raising ethical concerns regarding copyright. While the project is strictly academic and non-commercial, we acknowledge the importance of respecting intellectual property. To limit exposure, the Hugging Face repository will be made private once the project is completed.
  Concerning the training data for the LoRA fine-tuning, we used the train split of `allenai/sciq` reformatted.
  In addition to the textbook corpus, we explored other data sources and filtering strategies to enrich the RAG corpus, including the Wiki STEM Corpus (conjuring92, 2024), a curated subset of H4StackExchange (Lambert et al., 2023), and the student-generated preference-pair dataset. Due to time and resource constraints, these alternative datasets were not fully integrated into the final RAG pipeline but are described and made available in Appendix A.
  For the reward model, we built three preference datasets, each filtered to include only STEM-related content and formatted to share a consistent schema: prompt, chosen and rejected. Our first dataset (DPO1) comes from `argilla/ultrafeedback-binarized-preferences` (arg, 2023), filtered by running each prompt through the Mistral model, which we prompted to classify whether the question was STEM-related. The second dataset (DPO2) is a merge of `PrimeIntellect/SYNTHETIC-1-Preference-Data` (Mat-

tern et al., 2025a), a synthetic (Llama-generated) set of pairs responses to STEM questions, and `prhegde/preference-data-math-stack-exchange` (Lambert et al., 2023), a real and human-annotated preference pair dataset drawn from Math Stack Exchange. The last one comes from `allenai/reward-bench` (Lambert et al., 2024) filtered to include only STEM-related questions, and served as our primary evaluation set. Using these datasets, we were able to conduct a controlled study of how different training-set combinations affect downstream reward-modeling performance.

On another hand, the evaluation data to compare our models is based on six MCQA datasets. We used the test splits of the datasets D1 to D4 described earlier. Additionally, we included (D5) a dataset derived from the preference-pair dataset constructed earlier this semester by all students. We filtered it to retain only multiple-choice questions with four answers. Furthermore, we incorporated (D6) a synthetic benchmark consisting of 500 generated questions by Mistral, which were then double-checked by GPT-4o to ensure they were correct, leaving a total of 462 valid questions. Together, these six datasets cover a broad range of STEM domains and reasoning skills. Links to all evaluation datasets are provided in Appendix B.

- **Evaluation method:** For the MCQA, Quantized, and RAG models, accuracy is computed by evaluating each multiple-choice question individually, with four answer choices labeled `A`, `B`, `C`, and `D`. The model receives the question along with all four options. To ensure robustness, a human analysis was conducted to check that the model doesn't repeatedly produce the same outputs or follow undetected patterns in the data. This evaluation examined the model's responses for variability and consistency. Additionally, we used Mistral to analyze the explanations from our reasoning MCQA model specifically on the SciQ evaluation set. This process allowed us to investigate the primary causes of errors and understand the weaknesses in the model's reasoning. These errors were classified into four categories: hallucination, where the model generates false or fabricated information; irrelevance, where the response doesn't address the question; misinterpretation, where the model misunderstands the input; and inconsistency, where the response contains contradictions or logical errors. These metrics were instrumental in understanding the model's performance and pinpointing areas for improvement.

For the RAG model specifically, a set of relevant documents is retrieved from the FAISS vector database corresponding to the RAG corpus dataset. These documents are initially divided into 512-token chunks. Retrieval is performed by the embedding model, which ranks the chunks according to their cosine similarity with the question embedding, selects the top-k most relevant chunks, and includes them as context before the question and its answer choices. The exact prompt format used for evaluation is provided in Appendix E.

Concerning the quantized model, we also derived others metrics to assess the memory and inference performance: the model size in VRAM, as estimated by the `calculate_maximum_sizes` function from the `Accelerate` utilities, the VRAM usage immediately after loading the model onto the GPU, the peak VRAM during evaluation, the disk size of the model, the total evaluation time and the number of tokens generated per second (Appx H).

To compare the different quantization approaches, we defined a score that captures the trade-off between accuracy loss and VRAM usage for each evaluation set i, as shown in Equation 1:

$$\text{Score}^{(i)}_{\text{quant}} = \frac{\text{Accuracy}^{(i)}}{\text{ModelSize}_{\text{VRAM}}} \times 100 \quad (1)$$

We additionally compute the average score across all evaluation sets to summarize overall performance. Regarding the reward model measure of quality, we computed accuracy on held-out preference pairs drawn from our STEM-filtered evaluation dataset described earlier. We used the base model `Qwen/Qwen-0.6B-Base` as a reference model for scoring as long as the reward model accuracy. This evaluation protocol ensures that we compare models by their ability to predict preferences on new STEM questions, and how they are compared to the base model.

- **Baselines:** The baselines are the results obtained with `Qwen/Qwen3-0.6B-Base`.
- **Experimental details:** We fine-tuned the STEM-knowledge SFT model with a train batch size of 2, gradient accumulation steps of 8 (effective batch size 16), learning rate of 2e-5, and trained for 3 epochs. Training took approximately 29 hours.

For MCQA, we explored two main approaches: **full fine-tuning** and **LoRA-based fine-tuning**, the latter reducing computational cost by updating only a small portion of the model parameters.

LoRA Fine-Tuning (M1 model): LoRA was used to fine-tune the model on individual datasets and a merged set of around 30k examples, balancing performance and efficiency. We experimented

with learning rates of 5e-4, 1e-5, 5e-5, and 2e-6.

LoRA on Domain-Knowledge Pre-fine-Tuned Models (M2 model): LoRA was applied to a pre-fine-tuned model, rich in domain knowledge, to further optimize it for MCQA tasks.

Full Fine-Tuning (M3 model): Full fine-tuning was performed on a 150k dataset, comparing the impact of large-scale training with the efficiency of LoRA.

The dataset formatting strategy followed the same approach as in the LightEval evaluation, except for the reasoning model that received explanations alongside the correct letter. Training parameters were optimized to avoid overfitting, using evaluation loss to select the best model. We employed the AdamW optimizer with a learning rate of 5e-5, batch size of 4, gradient accumulation of 4, and 3 epochs. Training times varied: LoRA on SciQ took about 1 hour, while full fine-tuning took around 25 hours.

SmoothQuant + GPTQ (W8A8): The calibration set of 512 samples was drawn either from UltraChat200k or from the full MCQA dataset. In the latter case, samples were formatted as MCQA prompts in the same way as during `lighteval` evaluation. Each prompt was tokenized with a maximum sequence length of 2,048 tokens (as recommended in (Williams and Aletras, 2024)) and passed through the quantization pipeline, which first applied activation smoothing using SmoothQuant, followed by 8-bit quantization of linear layers using GPTQ (W8A8).

GPTQ (W4A16) - Weight-only quantization using GPTQ: The model was calibrated with `llmcompressor` on 512 random samples from UltraChat200k, with a maximum sequence length of 2,048 tokens. Linear layers were quantized to 4-bits.

QLoRA (4-bit base + LoRA adapters): We fine-tuned Qwen/Qwen3-0.6B-Base model using QLoRA, applying 4-bit quantization (NF4 + double quantization, bfloat16 compute) to the frozen base model. LoRA adapters were trained on MCQA-style prompts formatted similarly to `lighteval`. We experimented with both the full and balanced MCQA datasets, using 512, 4,096, 30,000 samples. The fine-tuning used a LoRA configuration with $r = 8$, $\alpha = 16$, and dropout $= 0.05$, targeting all linear layers (as recommended for smaller models in (Dettmers et al., 2023)). Training was performed for 1 epoch with a learning rate of 2e-4 and a batch size of 4. After training, the LoRA adapters were merged into the base model for evaluation.

For the RAG model, we fine-tuned both `Qwen/Qwen3-0.6B-Base` and the SFT model using LoRA. The LoRA configuration used a rank $r = 16$, `lora\_alpha=32` and a dropout of 0.05. During training, we retrieved the top $k = 5$ most relevant 512-token chunks for each question, based on cosine similarity in the embedding space, and prepended them to the prompt before the question and its answer choices. Training was done with a batch size of 4, a learning rate of 5e-5, 3 epochs, and gradient accumulation steps of 4. Each training run took approximately 7 hours. Regarding the reward model, we explored several training strategies, varying both the loss formulation and the choice of training data. We tried **ORPO** (Hong et al., 2024), a new fine-tuning technique that combines the traditional supervised fine-tuning and preference alignment stages into single process. This reduces the computational resources and time for training. In parallel, we also trained two close settings : **traditional DPO** that directly optimizes a pairwise preference ranking loss without requiring a separate policy gradient or reinforcement learning steps, and **DPO following a SFT**.

We tried training using ORPO by varying the learning rate and the training set. However, when comparing to our DPO results, ORPO underperformed relative to it. To improve upon this, we ran additional DPO experiments on the `argilla/ultrafeedback-binarized-preferences` dataset, in which we varied the learning rate (2e-6, 1e-5 and 5e-5) and whether we use SFT before or not. This yields six distinct training runs. By evaluating all six, we aim to understand both the impact of the learning rate itself and the effect of performing an SFT before DPO.

Across all runs, we used a train-validation split, AdamW for optimization, model checkpoints and the loading of the model based on the best performing model on the validation set.

- **Results:**

  We evaluated all of our models on the 6 evaluation datasets (D1 to D6) mentioned earlier. Starting with the balanced 30k dataset mentioned before, we fine-tuned the model using LoRa while varying the learning rate. This process was carried out across all the datasets, and we present their resulting average accuracy in 1 and details for each dataset individually are provided in Appendix G.

Lr	5e-4	1e-5	5e-5	2e-6
Acc.	51.60%	52.96%	55.13%	51.56%

Table 1: Average Accuracy vs Learning Rate

Then, we evaluated models M1 (LoRa-only with the balanced 30k dataset), M2 (LoRa on a model trained on knowledge domain) and M3 (Full-finetuning on the 150k dataset) using the best-

performing learning rate, presenting the results for the main models 1.

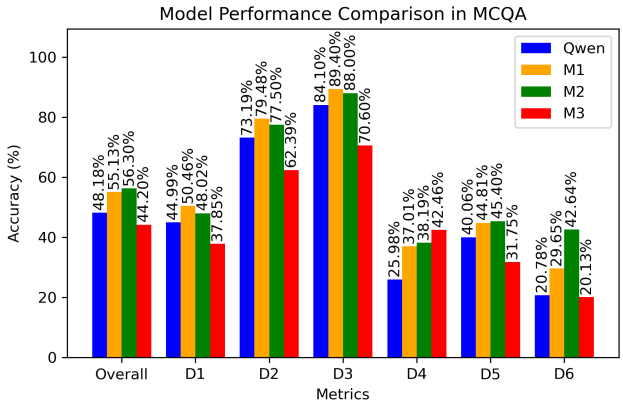

Figure 1: Comparison of MCQA models' accuracy

Therefore, we selected M1 as the best model for MCQA, as it strikes the best balance of accuracy across all our benchmarks.

For the reasoning model, a zero-shot prompt achieves an accuracy of 70.25%, while a one-shot strategy slightly improves this to 71.90% on the SciQ test set. Regarding the error types, we observe the following 4:

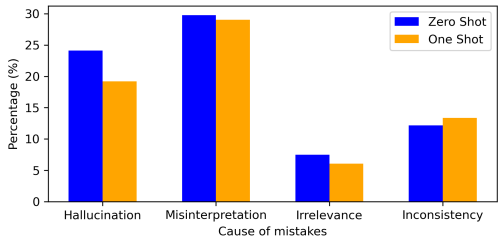

Figure 2: Comparison of errors for the reasoning model

Quantized approaches abbreviated as follows: Q1: Qwen3-0.6B-Base, Q2: 8-bit bitsandbytes, Q3: 4-bit bitsandbytes, Q4: SmoothQuant + GPTQ W8A8 calibrated on UltraChat (512 samples), Q5: GPTQ W4A16 calibrated on UltraChat (512 samples) and Q6: QLoRA on MCQA balanced (30k samples)

Model	Acc.	Size	Tokens/s	Score
Q1	48.40	1136.64	33.82	4.26
Q2	47.95	716.88	17.76	6.69
Q3	46.00	506.88	32.64	9.08
Q4	48.10	617.35	17.36	7.79
Q5	46.35	513.44	20.92	9.03
Q6	50.91	506.88	65.44	10.04

Table 2: Comparison of quantized models: average accuracy (%), VRAM model size (MB), tokens generated/s, and score (see Equation 1).

Q6 has the smallest model size (4-bit quantized weights) yet achieves the highest overall accuracy, outperforming all other quantized models and even the unquantized base model on most evaluation sets (Appx I). This makes it the best model in terms of accuracy–memory trade-off score. Although inference speed is hardware-dependent, GPTQ models tend to be slower due to runtime dequantization (vLLM Team, 2024). In contrast, QLoRA avoids these costs when LoRA weights are merged prior to evaluation.

For the RAG, we evaluated different models on the RAG corpus based on the 28 STEM books described earlier, with the `BAAI/bge-small` embedding model. The models evaluated are: (R1) `Qwen/Qwen3-0.6B-Base`; (R2) `NFX74/Qwen3-0.6B-Base-LoRA-SciQ-RAG` the fine-tuned model with LoRA adapter; (R3) `NFX74/Qwen3-0.6B-Base-SFT-STEM` the SFT model; (R4) `NFX74/Qwen3-0.6B-Base-SFT-STEM-LoRA-SciQ-RAG` the fine-tuned model with LoRA adapter on the SFT model; and (R5) `LinaSad/MNLP_M3_mcqa_model` the best MCQA model. The results are showed on 3:

Dataset	R1	R2	R3	R4	R5
D1	44.99	44.99	45.43	45.43	50.95
D2	73.19	73.19	71.41	71.41	79.19
D3	84.10	84.10	83.60	83.60	88.90
D4	25.98	25.98	23.62	23.62	28.74
D5	40.06	40.06	38.58	38.58	49.26
D6	20.78	20.78	33.55	33.55	29.87

Table 3: RAG accuracy for different models on selected datasets (values in %)

For the reward model, varying learning rate and training strategy gave us the results in figure 3 on the evaluation dataset.

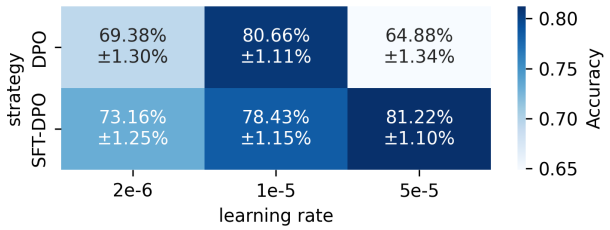

Figure 3: Performance comparison between different DPO training strategies.

4 Analysis

An unexpected finding, diverging from existing literature, is that Qwen fine-tuned with LoRa outperformed full fine-tuning with a larger dataset (Anyscale, 2023; Teknoloji, 2023). This can be attributed to LoRa's efficiency, which fine-tunes only a subset of parameters, allowing for faster convergence and less overfitting. The full fine-tuning dataset, dominated by AquaRat (about 80%), introduced bias towards it. Additionally, using LoRa with SFT for domain knowledge sometimes improved performance on certain benchmarks, but LoRa-only fine-tuning was more balanced across all benchmarks. The domain knowledge dataset, mainly consisting of Math and coding tasks, led to superior performance on domain-specific benchmarks like D4 and D5, but less so on more diverse bench-

marks. Its advantage was particularly noticeable in our synthetic benchmark, which includes a wider range of STEM subjects. However, since this benchmark is synthetic, and despite being evaluated by two advanced LLMs, we are cautious about fully generalizing these results. In terms of reasoning, the one-shot strategy slightly outperformed zero-shot, as previous research (Vinyals et al., 2016) suggests one-shot learning provides more context for better decision-making. However, both strategies faced issues with misinterpretation and hallucination, especially with the MCQA dataset's complex, context-dependent questions, which likely contributed to inaccuracies.

The fact that QLoRA outperforms the unquantized base model is expected: fine-tuning on domain-relevant MCQA prompts enables the model to better align with the structure and reasoning patterns required by the task. Furthermore, accuracy increases with the size of the fine-tuning dataset. Q6, trained on 30,000 domain-relevant examples, outperforms variants trained on smaller 512/4,096 subsets (Appx J), confirming the benefit of larger, high-quality data. Regarding calibration, models calibrated on UltraChat achieve comparable or superior accuracy to those calibrated on MCQA data (Appx K). This may be explained by UltraChat's greater diversity and broader coverage of question types, making it more representative of inference-time inputs and potentially leading to better calibration statistics. We also examined the effect of calibration set size. Increasing the number of calibration samples from 256 to 512 yields a measurable improvement, but gains diminish beyond that. This observation aligns with prior findings (Williams and Aletras, 2024), which suggest that a small but diverse calibration set is sufficient to capture representative activation statistics.

While RAG is generally expected to enhance performance by leveraging external knowledge, our experiments show limited benefits in the STEM multiple-choice setting: the RAG model underperforms the best MCQA model. This likely stems from partially relevant or noisy retrieved chunks that misalign with the question context, causing confusion rather than clarification. Surprisingly, the fine-tuned LoRA models (M2 and M4) perform identically to their base versions (M1 and M3) across all datasets, suggesting a possible error during fine-tuning that warrants further investigation. RAG's limited gains may also be due to suboptimal retrieval parameters, such as chunk size and the number of retrieved passages (k). Too many or overly long chunks can dilute relevance and lower accuracy. Similar concerns have been raised in prior work (Joren et al., 2024), emphasizing that retrieval quality, driven by retriever effectiveness and corpus structure, remains a key challenge for applying RAG to fine-grained STEM questions.

Initializing the reward model from an SFT-fine-tuned checkpoint consistently outperforms a "DPO-only" setup. At a learning rate of 5e-5, "DPO post SFT" reaches ≈81% held-out accuracy, whereas DPO-only peaks near 80 %. Lower learning rates (2e-6) lead to underfitting for both variants, confirming that warm-starting on domain knowledge places the model in a better parameter region and a relatively high learning rate is necessary for DPO's pairwise loss to move away from the SFT minimum.

5 Ethical considerations

Our paper focuses on fine-tuning Qwen for STEM subjects using English-language data. While this ensures consistency and high-quality datasets, it limits accessibility for non-English speakers, marginalizing learners proficient in other languages. Future work should consider multilingual support to promote more equitable access and bridge the digital divide. Another key concern is that students may become overly reliant on the model for answers, undermining engagement with course material and encouraging academic dishonesty. This could be mitigated by providing incremental hints rather than immediate answers, fostering active learning. Finally, the environmental impact of training large-scale models should not be overlooked. Optimizing efficiency and using renewable energy for deployment can help minimize the carbon footprint associated with this technology. By addressing these concerns, we can maximize the educational benefits of fine-tuning Qwen while minimizing its potential harms.

6 Conclusion

In this project, we developed a compact, specialized educational assistant using a 600M-parameter open-source LLM, fine-tuned on STEM synthetic question-answering data. The MCQA model with LoRa achieved an average accuracy of 55.1% on our benchmarks, while the quantized version maintained or exceeded base accuracy with reduced model size, balancing performance and efficiency. The RAG model, which retrieves information from STEM textbooks, showed limited improvement due to partially relevant retrievals, highlighting the importance of corpus quality and retrieval alignment. Our DPO-based reward model achieved over 80% accuracy on STEM preference data, outperforming ORPO, which was less effective. Experiments showed that starting from SFT and using smaller, filtered datasets with dropout regularization led to better results. In summary, lightweight fine-tuning, preference alignment, and careful dataset design can create strong STEM-specific assistants. Future work could extend this to broader educational tasks, explore retrieval-enhanced preference optimization, and address RAG limitations.

7 Contribution

Lina Sadgal was responsible for the MCQA fine-tuning and evaluation, Vincent Fiszbin handled the quantization experiments and analysis, Matthias Wyss developed the RAG pipeline and retrieval evaluation, and Sofia Taouhid worked on the DPO training and preference modeling. All team members collaborated on designing the SFT process and preparing the evaluation datasets.

References

2023. Huggingface argilla ulitrafeedback binarized preferences dataset.

Anyscale. 2023. Lora vs full fine-tuning: A comprehensive comparison. Accessed: 2025-05-14.

Zhongpu Chen, Yinfeng Liu, Long Shi, Zhi-Jie Wang, Xingyan Chen, Yu Zhao, and Fuji Ren. 2025. Mdeval: Evaluating and enhancing markdown awareness in large language models.

conjuring92. 2024. Wiki stem corpus. https://www.kaggle.com/datasets/conjuring92/wiki-stem-corpus. Accessed: May 21, 2025.

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

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The faiss library.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model.

Edward Hu, Xuezhi Shen, Xuehai Lin, and et al. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*. PMLR.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. 2024. Sufficient context: A new lens on retrieval augmented generation systems. ArXiv preprint arXiv:2411.06037.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. https://huggingface.co/spaces/allenai/reward-bench.

Nathan Lambert, Lewis Tunstall, Nazneen Rajani, and Tristan Thrush. 2023. Huggingface h4 stack exchange preference dataset.

Zhan Peng Lee, Andre Lin, and Calvin Tan. 2025. Finetune-rag: Fine-tuning language models to resist hallucination in retrieval-augmented generation.

Justus Mattern, Sami Jaghouar, Manveer Basra, Jannik Straube, Matthew Di Ferrante, Felix Gabriel, Jack Min Ong, Vincent Weisser, and Johannes Hagemann. 2025a. Huggingface primeintellect preferences dataset.

Justus Mattern, Sami Jaghouar, Manveer Basra, Jannik Straube, Matthew Di Ferrante, Felix Gabriel, Jack Min Ong, Vincent Weisser, and Johannes Hagemann. 2025b. Synthetic-1: Two million collaboratively generated reasoning traces from deepseek-r1.

Team Qwen. 2025. Qwen3 technical report.

Garantibbva Teknoloji. 2023. Full fine-tuning, lora, and qlora: A comparison of techniques in model optimization. Accessed: 2025-05-14.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3637–3645.

vLLM Team. 2024. Llm compressor documentation – vllm. https://blog.vllm.ai/llm-compressor/main/getting-started/compress/.

Miles Williams and Nikolaos Aletras. 2024. On the impact of calibration data in post-training quantization and pruning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 10100–10118. Association for Computational Linguistics.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2024. Smoothquant: Accurate and efficient post-training quantization for large language models.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

A Alternative RAG Data Sources

In addition to the textbook corpus, we explored several alternative data sources to diversify the RAG corpus. These include the Wiki STEM Corpus (conjuring92, 2024), which consists of Wikipedia articles filtered for STEM-related topics, as well as multiple subsets of the H4StackExchange dataset (Lambert et al., 2023). One subset includes only high-quality answers under 3,000 characters; another contains question–answer pairs where the question is under 2,000

characters and the answer under 3,000; a third version retains accepted or highly rated answers (score above 3), chunked into 1,000-character segments with 150-character overlap. We also generated a dataset from the student-generated preference-pair dataset by prompting `mistral-large-latest` with the question, its options, and the correct answer to obtain concise responses without chunking. These datasets were not integrated into the final RAG pipeline due to time and resource constraints, but they are publicly available for further exploration (see Appendix B).

B Resources

Code repository: https://github.com/matthias-wyss/LiSoViMa

Hugging face datasets:

- SFT training dataset: https://huggingface.co/NFX74/SFT_STEM_100k
- MCQA final training dataset: https://huggingface.co/datasets/LinaSad/MNLP_M3_mcqa_dataset
- MCQA large dataset: https://huggingface.co/datasets/LinaSad/LVSM_final
- MCQA synthetic evaluation set: https://huggingface.co/datasets/LinaSad/Synth_mistral
- MCQA preference data formatted evaluation set: https://huggingface.co/datasets/LinaSad/firstm_test_set
- Quantized models UltraChat200k calibration dataset: https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k
- Final Quantized model QLoRa fine-tuning dataset: https://huggingface.co/datasets/mkartofel/MNLP_M3_quantized_dataset
- RAG SciQ training dataset: https://huggingface.co/LiSoViMa/SCiQ_formatted
- RAG corpus dataset from 28 STEM books: https://huggingface.co/NFX74/rag_corpus_stem_books
- RAG corpus dataset from Wikipedia STEM: https://huggingface.co/NFX74/Wiki_STEM_Corpus
- RAG corpus dataset from H4 answers under 3,000 characters: https://huggingface.co/LiSoViMa/H4StackExchange_STEM_small_answers
- RAG corpus dataset from H4 questions under 2,000 characters and answers under 3,000: https://huggingface.co/LiSoViMa/H4StackExchange_STEM_small_questions_answers
- RAG corpus dataset from H4 selected or score above 3 answers chunked: https://huggingface.co/LiSoViMa/H4_STEM_selected_or_pm_score_over_3_chunked
- RAG corpus dataset from student dataset answers generated: https://huggingface.co/LiSoViMa/M1_MCQ_generated_with_questions
- DPO training dataset: https://huggingface.co/datasets/thdsofia/DPO_STEM_training

Hugging Face models:

- SFT model: https://huggingface.co/NFX74/Qwen3-0.6B-Base-SFT-STEM
- Final MCQA model (with LoRa): https://huggingface.co/LinaSad/MNLP_M3_mcqa_model
- MCQA fully finetuned model https://huggingface.co/LinaSad/mcqa_full_finetune
- MCQA with LoRa on top of domain knowledge https://huggingface.co/LinaSad/mcqa_lora_sft50k_final
- MCQA with learning rate 5e-4: https://huggingface.co/LinaSad/mcqa_lora_30k_5e_4
- MCQA with learning rate 1e-5: https://huggingface.co/LinaSad/mcqa_lora_30k_1e_5
- MCQA with learning rate 2e-6: https://huggingface.co/LinaSad/mcqa_lora_30k_2e_6
- MCQA model giving explanations: https://huggingface.co/LinaSad/mcqa_sciq_merged_reason_1
- Quantized model SmoothQuant + GPTQ W8A8 calibrated on UltraChat200k with 512 samples https://huggingface.co/mkartofel/Qwen3-0.6B-llmcompressor-W8A8-calib_ultrachat_512
- Quantized model SmoothQuant + GPTQ W8A8 calibrated on MCQA large dataset with 512 samples https://huggingface.co/mkartofel/Qwen3-0.6B-llmcompressor-W8A8-calib_LVSM_final_512
- Quantized model GPTQ W4A16 calibrated on UltraChat200k with 512 samples https://huggingface.co/mkartofel/Qwen3-0.6B-llmcompressor-W4A16-calib_ultrachat_512
- Quantized model QLoRa on MCQA large dataset with 1,024 samples https://huggingface.co/mkartofel/Qwen3-0.6B-qlora-LVSM_final_1024
- Quantized model QLoRa on MCQA large dataset with 512 samples https://huggingface.co/mkartofel/Qwen3-0.6B-qlora-LVSM_final_512_new_params
- Quantized model QLoRa on MCQA balanced dataset with 512 samples https://huggingface.co/mkartofel/

Qwen3-0.6B-qlora-MCQA_lora_final_512

- Final Quantized model QLoRa on MCQA balanced dataset with 30k samples https://huggingface.co/mkartofel/MNLP_M3_quantized_model
- RAG model with LoRA fine-tuning: https://huggingface.co/NFX74/Qwen3-0.6B-Base-LoRA-SciQ-RAG
- RAG model with LoRA fine-tuning from SFT model: https://huggingface.co/NFX74/Qwen3-0.6B-Base-SFT-STEM-LoRA-SciQ-RAG
- DPO model with learning rate 5e-5: https://huggingface.co/thdsofia/DPO_model_lr5e-5
- DPO model with learning rate 1e-5: https://huggingface.co/thdsofia/DPO_model_lr1e-5
- DPO model with learning rate 2e-6: https://huggingface.co/DPO_model_lr2e-6
- DPO model with learning rate 2e-6, post SFT: https://huggingface.co/DPO_model_lr2e-6_postSFT
- DPO model with learning rate 5e-5, post SFT: https://huggingface.co/thdsofia/DPO_model_lr5e-5_postSFT
- DPO model with learning rate 1e-5, post SFT: https://huggingface.co/thdsofia/DPO_model_lr1e-5_postSFT
- Final DPO model: https://huggingface.co/MNLP_M3_dpo_model

C MCQA - Models' result depending on dataset

- M1 : LoRa with MMLU trainig set
- M2 : LoRa with SciQ trainig set
- M3 : LoRa with 10k of AquaRat trainig set
- M4 : LoRa with 50k of AquaRat trainig set
- M5 : LoRa with all of AquaRat trainig set

Dataset	M1	M2	M3	M4	M5
All	52.97	54.54	52.04	53.10	53.62
D1	50.55	44.50	49.54	49.89	48.85
D2	79.50	77.98	77.16	77.59	77.70
D3	83.70	89.60	85.60	85.10	85.50
D4	31.10	33.46	35.83	41.34	46.06
D5	48.07	48.37	48.07	46.29	46.29
D6	24.89	28.35	16.02	18.40	17.32

Table 4: MCQA accuracy for models on selected datasets (values in %)

D MCQA - Dataset distribution for full-finetuning:

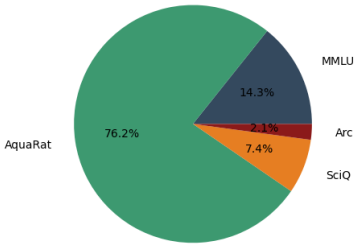

Figure 4: Dataset distribution for mcqa full-finetuning

E Evaluation Prompt Format

For evaluating multiple-choice questions (MCQA and quantized models), we used the following prompt format:

```
The following are multiple choice
questions (with answers) about
knowledge and skills in advanced
master-level STEM courses.

<Question text>

A. <Choice A>
B. <Choice B>
C. <Choice C>
D. <Choice D>

Answer:
```

For the RAG model, the prompt also includes top-k chunks retrieved from a FAISS vector database. These chunks are prepended before the question to provide additional context:

```
Relevant Documents:
Document 0:::
<retrieved_chunk_0>

Document 1:::
<retrieved_chunk_1>

...

Document k-1:::
<retrieved_chunk_k-1>

The following are multiple choice
questions (with answers) about
knowledge and skills in advanced
```

```
master-level STEM courses.


<Question text>


A. <Choice A>
B. <Choice B>
C. <Choice C>
D. <Choice D>


Answer:
```

The model evaluates all answer choices ('A', 'B', 'C', 'D') by computing their conditional log-likelihoods and selects the one with the highest score as its final answer.

F MCQA Training Code

The full training configuration used for all fine-tuning experiments is provided below.

```
training_args = SFTConfig(
   output_dir="./configs/qwen-general-full",
    per_device_train_batch_size=4,
    per_device_eval_batch_size=4,
    num_train_epochs=3,
    save_strategy="steps",
    logging_dir="./logs",
    logging_steps=100,
    learning_rate=5e-5,
    weight_decay=0.01,
    gradient_checkpointing=True,
    gradient_accumulation_steps=4,
    eval_strategy="steps",
    eval_steps=100,
    save_safetensors=False,
    load_best_model_at_end=True,
    metric_for_best_model="eval_loss",
    greater_is_better=False,
    completion_only_loss=True,
)
```

G MCQA - detailed models results with variation of learning rate

These models were all run using LoRa and the 30k balanced dataset.

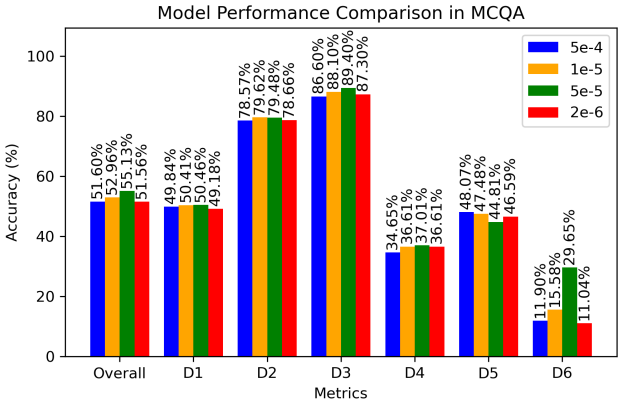

Figure 5: Models accuracies on MCQA while varying the learning rate

H Quantization - Quantization models performance metrics

Quantized approaches abbreviated as follows:

- **Q1**: Unquantized base model (Qwen3-0.6B-Base)
- **Q2**: 8-bit quantization using `bitsandbytes`
- **Q3**: 4-bit quantization using `bitsandbytes`
- **Q4**: SmoothQuant + GPTQ (W8A8), calibrated on UltraChat (512 samples)
- **Q5**: GPTQ (W4A16), calibrated on UltraChat (512 samples)
- **Q6**: QLoRA fine-tuned on 30k samples from the MCQA balanced dataset

Performance metrics computed:

- Model size in VRAM, as estimated by the `calculate_maximum_sizes` function from the `Accelerate` utilities (GB)
- VRAM usage immediately after loading the model onto the GPU (GB)
- Peak VRAM during evaluation (GB)
- Disk size of the model (GB)
- Total evaluation time (minutes)
- Number of tokens generated per second (tokens/second)

Metric	Q1	Q2	Q3	Q4	Q5	Q6
Model size	1.11	0.70	0.49	0.70	0.50	0.49
VRAM load	1.40	0.73	0.70	0.99	0.87	0.87
Peak VRAM	18.56	19.10	18.50	19.35	19.25	19.62
Disk size	2.26	2.26	2.26	2.03	1.62	1.62
Eval time	26.94	46.31	27.88	46.93	40.14	27.61
Tokens/s	33.82	17.76	32.64	17.36	20.92	65.44

Table 5: Performance and memory metrics for each quantized model.

I Quantization - Quantization models accuracy on all evaluations sets

Quantized approaches abbreviated as follows:

- **Q1**: Unquantized base model (Qwen3-0.6B-Base)
- **Q2**: 8-bit quantization using `bitsandbytes`
- **Q3**: 4-bit quantization using `bitsandbytes`
- **Q4**: SmoothQuant + GPTQ (W8A8), calibrated

on UltraChat (512 samples)

- **Q5**: GPTQ (W4A16), calibrated on UltraChat (512 samples)
- **Q6**: QLoRA fine-tuned on 30k samples from the MCQA balanced dataset

Eval set	Q1	Q2	Q3	Q4	Q5	Q6
All	48.40	47.95	46.00	48.10	46.35	50.91
D1	48.85	48.49	40.41	48.09	44.64	48.14
D2	76.35	75.87	59.94	76.32	70.90	75.70
D3	84.20	83.40	75.20	83.90	82.80	86.10
D4	31.50	29.92	24.80	30.71	28.74	40.55
D5	36.50	38.58	30.86	36.80	29.38	41.54
D6	12.99	11.47	44.81	12.77	21.65	13.42

Table 6: Accuracy (%) of each quantized model on each evaluation set.

J Quantization - Impact of sample size for QLoRa fine-tuning

To assess how fine-tuning sample size affects accuracy, we trained QLoRA models on randomly sampled subsets of the MCQA 30k balanced dataset, using 512, 4,096, and 30,000 examples.

Eval set	512	4,096	30,000
All	42.44	44.18	50.91
D1	40.83	42.58	48.14
D2	65.49	67.66	75.70
D3	78.10	80.80	86.10
D4	27.95	28.35	40.55
D5	31.45	34.42	41.54
D6	10.82	11.26	13.42

Table 7: Accuracy (%) of QLoRA models on each evaluation dataset with varying fine-tuning set size.

K Quantization - SmoothQuant + GPTQ (W8A8) models, calibrated on different datasets and with various calibration set sizes

The models are quantized using llmcompressor, with the SmoothQuant + GPTQ pipeline defined in Section 2. At this point, the demo MCQA evaluation set was used to compare the models' performance. The different models are abbreviated as follows:

- **QA**: Calibrated on SciQ (512 samples)
- **QB**: Calibrated on MCQA full (512 samples)
- **QC**: Calibrated on UltraChat (256 samples)
- **QD**: Calibrated on UltraChat (512 samples)
- **QE**: Calibrated on UltraChat (1024 samples)

Eval dataset	QA	QB	QC	QD	QE
All	55.60	56.48	54.94	56.52	55.97
D1	48.49	48.83	48.02	48.49	48.21
D2	76.01	76.46	75.84	76.29	75.90
D3	84.00	84.00	84.10	84.30	84.20
D4	31.50	31.10	28.74	29.53	29.53
Demo	38.00	42.00	38.00	44.00	42.00

Table 8: Accuracy (%) of SmoothQuant + GPTQ (W8A8) models calibrated on different datasets and sample sizes.

L Reward Model - ORPO results

To assess how both the choice of training data and the optimization hyperparameters affect ORPO's performance, we carried out two experiments. Table 9 examines the impact of dataset composition by reporting the accuracy achieved by models trained on DPO1 alone, DPO2 alone, and the combined DPO1 + DPO2 datasets. This comparison lets us isolate the contribution of each data source and evaluate whether simply merging them yields further gains. Table 10 explores the sensitivity of ORPO to the learning rate. We retrained identical model architectures using three different step sizes 1e-5, 2e-5 and 5e-5, reporting the resulting accuracies.

Train dataset	DPO1	DPO2	DPO1+DPO2
Accuracy	42.62	57.93	57

Table 9: Accuracy (%) of ORPO models trained on different datasets.

Learning rate	1e-5	2e-5	5e-5
Accuracy	57.14	52.41	57.93

Table 10: Accuracy (%) of ORPO models trained using different learning rates.