QwenSTEM: Specialized Fine-tuning of Small Language Models for University-Level STEM Education

Sayantan Biswas | SCIPER 388904 | sayantan.biswas@epfl.ch
Bich Ngoc Doan | SCIPER 395722 | bich.doan@epfl.ch
Khanh Nguyen | SCIPER 371125 | khanh.nguyen@epfl.ch
Reinatt Hansel Wijaya | SCIPER 394199 | reinatthansel.wijaya@epfl.ch

Abstract

This work explores optimization techniques for fine-tuning Qwen3-0.6B-Base on educational multiple-choice question answering using EPFL course materials. We evaluate Direct Preference Optimization (DPO), supervised fine-tuning with and without reasoning augmentation, quantization, and Retrieval-Augmented Generation (RAG). Our analysis reveals that supervised fine-tuning consistently improves performance, with the counterintuitive finding that non-reasoning variants achieve slightly higher accuracy despite reduced interpretability. DPO provides additional improvements depending on training data, while RAG effectiveness varies by subject. Quantization achieves significant compression while maintaining competitive performance, demonstrating compact model viability for educational applications.

1 Introduction

Multiple-choice question answering represents a fundamental assessment format across educational institutions, yet automated systems for this task face distinct challenges. Recent advances in language model optimization have opened new possibilities for creating specialized educational assistants, but questions remain about which techniques provide the most effective improvements.

This work explores the application of Qwen3-0.6B-Base for educational MCQA, with particular attention to understanding when and why different approaches succeed. We examine four key optimization strategies: Direct Preference Optimization (DPO) for aligning model outputs with educational preferences, supervised fine-tuning with domain-specific data, quantization for deployment efficiency, and Retrieval-Augmented Generation (RAG) for incorporating external knowledge sources.

Our investigation focuses on the comparison between reasoning and non-reasoning training approaches. We find that models trained without explicit reasoning steps achieve marginally higher accuracy on MCQA tasks, despite offering less insight into their decision-making process. Using EPFL course materials as our evaluation benchmark, our compact model approach also investigates whether significant performance gains can be achieved without the computational overhead of larger models. These findings contribute to the growing understanding of how to effectively adapt compact LLMs for specialized educational applications.

2 Approach

Qwen3-0.6B demonstrates strong performance across math, reasoning, and coding benchmarks while maintaining a compact 600M parameter size suitable for resource-constrained deployment (Yang et al., 2025). Full details of how we leverage this model for different approaches are illustrated in Figure 1.

DPO Our training is centered on DPO combined with a curriculum learning strategy. Given the varied difficulty of our datasets, we train the model progressively rather than on a single concatenated dataset. The curriculum begins with data exhibiting low cosine similarity (low information) and gradually introduces data with higher cosine similarity (high information), allowing the model to learn from simpler to more complex examples.

Recognizing that the crowd-sourced M1-Preference dataset has a higher potential for label noise, we also incorporated a ramp loss function into our DPO training. The ramp loss is known for its robustness on noisy datasets compared to standard loss functions. It is a non-convex function defined as $L(y, f(x)) = \max(0, 1 - yf(x)) - \max(0, -s - yf(x))$

MCQA To develop MCQA ability, we apply SFT to the base model data on various STEM-related MCQA datasets, experimenting with sev-

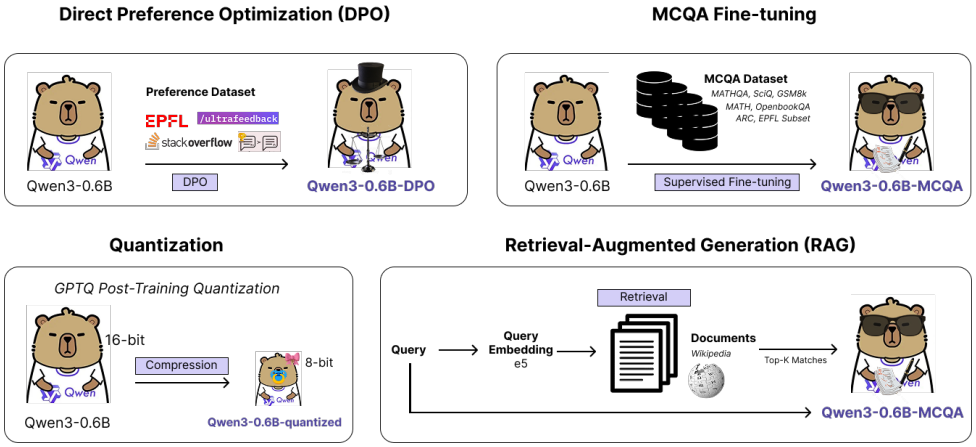

Figure 1: A summary of the methods used to create different variants of Qwen3-0.6B.

eral setups. First, we compare incremental learning (grouping datasets by difficulty progression) with mixed training (random shuffling of all datasets). The incremental approach may improve learning efficiency (Soviany et al., 2022) but risks catastrophic forgetting (Kemker et al., 2018), while mixed training maintains diverse exposure throughout training. Secondly, for reasoning integration, we implement two variants: (1) with reasoning using "Let's think step-by-step" prompts, and (2) without reasoning. Complete prompt templates for both variants are in Appendix D. We also explored DPO followed by SFT, however this resulted in performance degradation, so we excluded it from final analysis.

Quantization We quantize our model using Post Training Quantization (PTQ) with different algorithms. We have GPTQ (Frantar et al., 2023), AWQ (Lin et al., 2024), GPTQ + SmoothQuant (Xiao et al., 2024), and bits&bytes. The GPTQ algorithm solves the quantization problem by iteratively selecting and quantizing weights in a fixed order for all rows of a matrix, regardless of the individual weights' impact on error. AWQ on the other hand search for the optimal per-channel scaling that protects the salient weights by observing the activation, not weights. Lastly, SmoothQuant smooths the activation outliers by offline migrating the quantization difficulty from activations to weights with a mathematically equivalent transformation.

RAG For our query encoder, we fine-tune the pre-trained multilingual E5 embedding base model (Wang et al., 2024). For our generator, we experiment with deploying directly the MCQA model to our pipeline, as well as fine-tune the base Qwen model. Our training data are synthesized from the comprehensive Wikipedia STEM english corpus, which is also chosen for our document chunks.

3 Experiments

3.1 Data

DPO For our experiments, we utilize three distinct datasets, each with unique characteristics regarding generation process and similarity with preference pairs. *Math Stack Exchange*: This is a well-cleaned dataset containing approximately 30,000 data points derived from naturally occurring human preferences on the platform. *UltraFeed*: We use a filtered subset of the UltraFeed dataset, which consists of AI-generated responses with human-labeled preferences. After selecting for STEM-style QA pairs, our subset contains around 10,000 examples. *M1-Preference*: The dataset was created as part of our coursework at EPFL and contains 1,200 unique questions corresponding to 24,000 preference pairs. To enhance learning robustness, we further augmented this dataset by paraphrasing the questions.

Following the method described by (Shen et al., 2024), we analyze the cosine distance between the sentence embeddings of the preference pairs and visualize the differences between datasets in Figure 2) as a proxy for the relative learning difficulty.

MCQA To develop a comprehensive STEM-focused MC knowledge base, we assembled a diverse dataset covering multiple aspects of scientific and mathematical reasoning. A full summary of the data mix with corresponding sizes and sources is included in Appendix C. *1. Foundational Knowledge*: SciQ, OpenBookQA, and ARC-Easy establish basic scientific literacy and factual understanding across physics, chemistry, biology, and earth sciences. *2. Mathematical Reasoning*: GSM8K and Math datasets provide graded difficulty in nu-

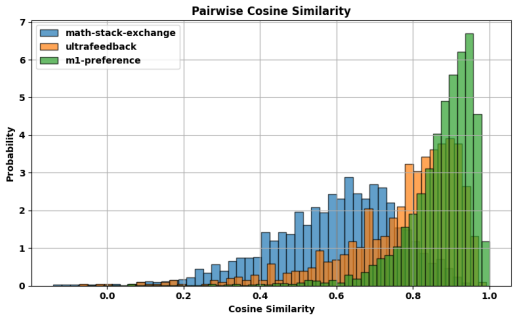

Figure 2: Distribution of cosine similarity of pairs from each dataset

merical problem-solving, from elementary arithmetic through advanced mathematical concepts. *3. Advanced Scientific Reasoning*: MathQA and ARC-Challenge present complex abstract reasoning challenges requiring multi-step logical inference and sophisticated mathematical manipulation. *4. EPFL-Specific Content*: A subset of MC questions from the DPO's provided EPFL-coursework.

These categories correspond to the incremental phases we experiment with in our SFT, which we refer as *"Four-stage Training"* (1→2→3→4). All datasets collected include accompanying explanations for the QAs, which serve as the optional "reasoning" component for training our models.

Quantization Calibration dataset used in post training quantization is important in determining LLM performance (Williams and Aletras, 2024). Our main GPTQ model was trained on 3 datasets: C4 (Raffel et al., 2023), chosen based on the suggestion from the GPTQ paper (Frantar et al., 2023) with excerpts from randomly crawled websites representing generic text data; Wikitext2 (Merity et al., 2016), extracted from the set of verified good and featured articles on Wikipedia; and our own sample from the MCQA datasets.

RAG We use the latest STEM Wikipedia corpus for ease of implementation, since they do not impose any license restriction. To post-train our encoder, we used the triplet loss, first proposed to train FaceNet (Schroff et al., 2015). The corresponding data was constructed by first synthesizing triplet data of the form (anchor, positive, negative); then, for every chosen pair of (positive, negative) documents, we used GPT4o-mini (OpenAI, 2024) to generate MCQA questions based on the positive documents. We noticed one of the limitations of using 4o-mini is that the majority of the time it only created fact-based or summarizing questions but not inferential questions. For fine-tuning the generator, we trained following the technique proposed in (Zhang et al., 2024a). Using the previously synthesized data, we created a training dataset as follows: for $P\%$ of the data, we put the positive document plus $k-1$ random documents in the prompt, and for $(1-P)\%$ of the data, we only included k random documents.

3.2 Evaluation Method

DPO For benchmarking, we create test splits for all three datasets mentioned, along with one additional dataset, STEM selected version of reward-bench which consist of 1.4K preference pairs. Specifically for the M1-Preference dataset, we split it by `question_id` to ensure that model is evaluated only on novel preference pairs, without access to any training information of the answer correctness.

MCQA We employ a dual evaluation framework to assess both model performance and generalization capabilities. Our primary evaluation uses the held-out *test split* from our collected data mixture to measure in-distribution performance on trained content types. For out-of-distribution robustness testing, we evaluate on *MMLU-STEM*, a standardized benchmark subset covering mathematics, physics, chemistry, biology, and computer science that was not included in our training mixture (Hendrycks et al., 2020). This independent evaluation assesses the model's ability to transfer learned STEM reasoning to novel question formats and topics. We report accuracy across both evaluation sets, further complemented by a subject-specific performance analysis to examine strengths and weaknesses across STEM domains.

Quantization The performance evaluation is the same as MCQA model, with additional measurements of average peak VRAM usage and tokens/s. To observe the average peak VRAM usage, we measure cuda max memory allocated during inference process. We use vLLM (Kwon et al., 2023) for throughput benchmarking. To assess efficiency, we introduce a metric balancing accuracy loss and VRAM usage reduction relative to the non-quantized model: $\text{Score} = (1 - \Delta\text{Acc})w_{\text{acc}} + \Delta\text{vram} \cdot w_{\text{vram}}$, where $\Delta\text{Acc} = \frac{\text{Acc}_{\text{nonq}} - \text{Acc}_{\text{q}}}{\text{Acc}_{\text{nonq}}}$ and $\Delta\text{vram} = \frac{\text{vram}_{\text{nonq}} - \text{vram}_{\text{q}}}{\text{vram}_{\text{nonq}}}$. and weights are $w_{\text{acc}} = 0.75$ and $w_{\text{vram}} = 0.25$. We put more weight on accuracy given the small model size.

RAG To evaluate the overall performance of our pipeline, we will evaluate it following our method

for MCQA models.

3.3 Baselines

DPO For evaluation, we conduct a series of experiments using two base models: (1) a MCQA model trained specifically for STEM multiple-choice question answering as part of this project, and (2) the Qwen3 0.6 Base model, a strong general-purpose language model. All the reward results are calculated with Qwen3 0.6 Base as the reference model.

MCQA We establish two primary baselines for comparison: (1) the original Qwen3-0.6B base model without any fine-tuning, and (2) the same base model prompted without explicit reasoning steps, to measure both the model's improvement in STEM capabilities and its reasoning effectiveness.

Quantization Our baseline model will be the fine-tuned MCQA model, noting performance loss and VRAM usage to measure the quantization effectiveness in preserving model quality.

RAG With the Wikipedia documents, we propose two baselines: (1) Base Qwen3-0.6B model for evaluation generator, using multilingual-e5-base encoder with $k = 5$ (2) Our MCQA model for evaluating encoder, using base encoder with $k = 5$.

3.4 Experimental details

DPO We perform a Bayesian hyperparameter sweep using Weights & Biases (`wandb`), varying parameters such as `learning_rate` and `beta`. The configuration that yields the lowest evaluation loss is selected for final training B. For efficiency, we conduct the sweep on a 20% stratified sample of the M1-Preference dataset, ensuring an equal distribution across questions.

MCQA Our SFT uses the AdamW optimizer with a learning rate of 2e-5. Training employed a per-device batch size of 4 with gradient accumulation steps of 16, across 2 epochs with weight decay of 0.01. To focus learning on response generation, we masked the question portions during training so that loss calculation applied only to the answer and explanation components.

Quantization We applied GPTQ, bits&bytes , SmoothQuant + GPTQ, and AWQ for quantization and experimented with different bit sizes (mainly 4 and 8). Each quantization approximately takes around 15-20 minutes to run.

RAG We trained the encoder with the triplet loss using a learning rate of 2e-5. For the generator model, we experimented with different $k = 3, 4, 5$ and different numbers of fine-tuning data from 0 to 20000. We deployed our MCQA models directly into our RAG pipeline without fine-tuning.

3.5 Results

DPO We present the phased training results in Table 1. We observe that the Qwen base model shows a steady improvement on the Preference Data Benchmark across training phases. In contrast, the MCQ model's performance is less consistent, which may be due to its training objective of generating only a small number of tokens.

On the Reward Benchmark, we see an initial significant improvement, followed by a decline, and then a further increase as training on preference data progresses. Notably, direct training on the UltraFeedback dataset using ramp loss achieves an impressive accuracy of 90% on the Reward Benchmark, highlighting the effectiveness of ramp loss in handling noisy preference data.

MCQA Table 2 presents our quantitative results across different training configurations. It's clear that the progressive four-stage training approach does not outperform single-stage mixed training. It also exhibits a general performance decline towards subsequent phases, indicating a strong effect of *catastrophic forgetting*. Surprisingly, training without explicit reasoning steps consistently outperforms reasoning-based training. The no-reasoning approach achieves 50.0% accuracy compared to 47.0% for reasoning-based training in the optimal mixed setting. This suggests that for this model scale and task complexity, step-by-step reasoning may introduce unnecessary complexity that hampers rather than helps performance. Overall, results remain consistent between our test split (in-distribution) and MMLU-STEM (out-of-distribution), with MMLU-STEM scores typically 1-3% lower. This suggests our training approach develops genuine STEM reasoning capabilities rather than dataset-specific overfitting.

Quantization Table 3 presents the effect of the calibration dataset on GPTQ quantization, while Table 4 summarizes the quantization results on MMLU-STEM. All quantized models significantly reduce VRAM usage and model size. Notably, GPTQ W8A16 achieves a 72.9% reduction in VRAM usage with only a 0.36% accuracy drop, resulting in the highest overall score of 0.9296. This indicates that GPTQ W8A16 offers the most balanced trade-off between accuracy and resource efficiency. Interestingly, AWQ demonstrates a higher

Training Setting	Qwen3 0.6 Base		MCQA Model	
	M1 preference	Reward Bench	M1 preference	Reward Bench
No training (fine-tuning)	0.50	0.5	0.492	0.674
Three-stage Training				
Phase 1: Math-Stack-Exchange	0.491	**0.844**	0.488	0.742
Phase 2: UltraFeed	0.537	0.712	0.531	0.721
Phase 3: EPFL-M1-preference	**0.549**	0.748	0.524	0.741

Table 1: Evaluation results for different settings under DPO training.

Training Setting	With reason		Without reason	
	Test Split	MMLU STEM	Test Split	MMLU STEM
No fine-tuning	-	-	0.454	0.458
Four-stage Training				
Phase 1: SciQ, OpenbookQA, ARC Easy	0.474	0.465	0.474	0.465
Phase 2: GSM8K, MATH	0.474	0.462	0.475	0.463
Phase 3: MathQA, ARC Challenge	0.46	0.449	0.458	0.450
Phase 4: EPFL Subset	0.462	0.450	0.460	0.452
One-stage Training				
Data mix	0.47	0.461	**0.500**	**0.472**

Table 2: Evaluation results for different settings under MCQA training.

Dataset	**Accuracy (%)**
C4	0.4703
Wikitext2	0.4613
Sample of MCQA dataset (Table 5)	0.4701

Table 3: Accuracy per dataset for GPTQ 8 bit

accuracy than MCQA model. We hypothesize that this is due to optimal per-channel activation scaling with quantization. It is however important to note that VRAM usage for AWQ and SmoothQuant could not be accurately measured, and token/s performance is unavailable due to lack of support in bitsandbytes for vLLM. Overall, the results demonstrate that quantized models can retain strong performance, with some configurations, such as AWQ, even outperforming the original in accuracy.

RAG From initial experiments, the results showed that the best k is 4, but performance gap was negligible. Our finetuning results showed that under our choice of documents and trained encoder with $k = 4$, the more samples we used to fine-tune our MCQA model, the worse the generator became. With 100 samples, our model became 1% worse compared to simply deploying our MCQA model into the RAG pipeline, and with 20000 samples, the performance became roughly the base MCQA model. In contrast, with the base model and the base encoder as well as $k = 5$, the RAG pipeline increased the performance to 0.404 on the evaluation data mix. Switching the base encoder to a trained encoder increased the score to 0.421 and with our MCQA model it became 0.490.

4 Analysis

Qualitative Result To compare reasoning and non-reasoning MCQA models, we qualitatively analyzed outputs from random test samples. For instance, when presented with a chi-squared statistics interpretation question (Appendix E, the non-reasoning model produced only the final answer choice ("B"), while the reasoning model generated a complete step-by-step explanation, leading to the correct answer ("A"). The analysis showed that, despite higher accuracy in some cases by the non-reasoning model, the reasoning model's transparent process enables verification of its conclusions. With regard to DPO, its generation (Appendix E.1.2) reveals that, although the model generates comprehensive answers, it sometimes makes logical errors due to the lack of training on STEM data. This suggests that model selection should balance accuracy against the need for interpretable reasoning in educational contexts.

Quantization Setting	Accuracy	Peak VRAM (MB)	Token/s	Score
MCQA model	0.472	3814.86	257.17	-
GPTQ W8A16	0.4703	1032.9	299.4	0.9296
GPTQ W4A16	0.3923	821.26	294.39	0.8195
GPTQ + SmoothQuant W8A8	0.471	1447.66	295.1	0.9035
GPTQ + SmoothQuant W4A8	0.4355	1433.23	298.3	0.8298
bits&bytes W8A16	0.4684	1024.57	-	0.9271
bits&bytes W4A16	0.4152	819.85	-	0.8560
AWQ W4A16	0.4735	2288.36	260.31	0.8524

Table 4: Evaluation results for different settings after Quantization

MCQA and RAG We compare performance patterns of the fine-tuned MCQA model and its RAG-enhanced counterpart across STEM domains. Figure 3 shows strengths diverge across different subjects. The RAG approach demonstrates superior performance in physics-related tasks, likely benefiting from the comprehensive contextual information for physics concepts and problem-solving. Conversely, MCQA model shows advantages in biological sciences, suggesting that the curated MCQA training data may be more aligned with biological question formats and terminology. The RAG model's ability to leverage external knowledge appears beneficial for subjects requiring extensive factual recall and conceptual understanding, such as medicine and mathematics, while the fine-tuned model's internalized knowledge proves more effective for domains where the training distribution closely matches the evaluation tasks. These findings suggest that the optimal approach may be domain-dependent, with RAG providing value when external knowledge retrieval complements the model's reasoning capabilities.

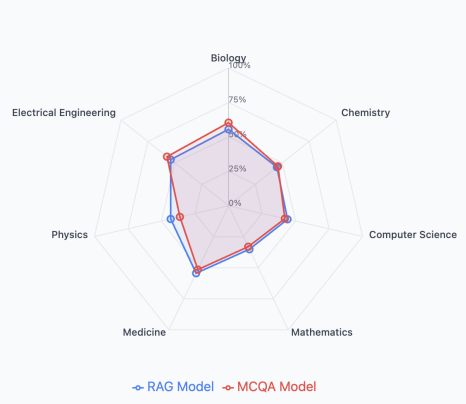

Figure 3: Performance comparison of MCQA model versus RAG-enhanced model across STEM subjects.

5 Ethical considerations

When functioning as intended, our model could democratize access to educational assistance, benefiting students in resource-constrained environments who lack access to human tutors. However, the model's reliance on existing datasets may perpetuate biases present in educational materials, potentially providing incorrect information or reinforcing cultural stereotypes. Additionally, while the reasoning variant offers interpretability, both model versions could be misused to facilitate academic dishonesty if deployed without proper safeguards.

To address these concerns, we recommend implementing robust content filtering, regular bias auditing, and clear usage guidelines that emphasize the tool's role as an educational aid rather than an authoritative source. Furthermore, developing multilingual versions should prioritize collaboration with native speakers and educators from target language communities to ensure cultural appropriateness and accuracy.

6 Conclusion

This project demonstrates that compact language models can be effectively optimized for educational MCQA through targeted fine-tuning strategies. Our key finding reveals that non-reasoning models achieve slightly higher accuracy than reasoning-augmented variants, despite reduced interpretability - highlighting important trade-offs in educational AI design. Supervised fine-tuning proved most consistently beneficial across all conditions, while DPO and RAG showed context-dependent improvements, and quantization maintained competitive performance with significant compression.

References

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

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 preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems, 2021. *URL https://arxiv. org/abs/2110.14168*, 9.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

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

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. 2018. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for llm compression and acceleration.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

OpenAI. 2024. GPT-4o mini: advancing cost-efficient intelligence — openai.com. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. [Accessed 11-06-2025].

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.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Judy Hanwen Shen, Archit Sharma, and Jun Qin. 2024. Towards data-centric rlhf: Simple metrics for preference dataset comparison. In *NeurIPS*.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024a. Raft: Adapting language model to domain specific rag.

Ziyin Zhang, Lizhen Xu, Zhaokun Jiang, Hongkun Hao, and Rui Wang. 2024b. Multiple-choice questions are efficient and robust llm evaluators. 2024d. *URL https://doi. org/10.48550/arXiv*, 2405.

A AI Usage Appendix

For RAG, we used chatgpt-4o-mini to generate synthesized questions to train our generator and

encoder. In particular, among our set of documents, we chose in random a subset of 20000 documents and generated, for each of them, a question based on the document. We verified the correctness by randomly sampling rows of the generated data.

B Hyperparameter for DPO

Loss: Hinge Loss, beta: 0.0009, warmup_ratio: 0.15, scheduler: cosine_with_restarts, num_train_epochs: 4, Learning_rate: 1e-5, weight_decay: 0.1

C Training Data for MCQA

Table 5

Dataset	Count	Percentage (%)
MathQA (Amini et al., 2019)	29837	46.37
SciQ (Welbl et al., 2017)	11679	18.15
GSM8K (Cobbe et al., 2021; Zhang et al., 2024b)	7468	11.61
MATH (Zhang et al., 2024b; Hendrycks et al., 2021)	6540	10.16
OpenbookQA (Mihaylov et al., 2018)	4957	7.70
ARC Easy (Clark et al., 2018)	2241	3.48
ARC Challenge (Clark et al., 2018)	1117	1.74
EPFL Subset	506	0.79
Total	64345	100

Table 5: Dataset used for MCQA training.

D Training Prompt Templates

This section presents the prompt formatting templates used during fine-tuning for both model variants. The templates define the structure and format of training examples presented to the models.

D.1 Non-reasoning Model Template

Non-reasoning Training Template

```
Question: {question}

Choices:

{choices_formatted}

Answer: {answer_letter}. {choice_text}
```

D.2 Reasoning Model Template

Reasoning Training Template

```
Question: {question}

Choices:

{choices_formatted}

Let's think step-by-step: {explanation}

Answer: {answer_letter}. {choice_text}
```

E MCQA Qualitative Example

This appendix presents representative examples of model outputs from both the reasoning and non-reasoning variants when presented with identical prompts from the EPFL dataset subset.

E.1 Example: Chi-squared Statistics Interpretation

E.1.1 Non-reasoning Model Output

Input Prompt

If for the χ^2 statistics for a binary feature, we obtain $P(\chi^2 | DF = 1) < 0.05$, this means:
Options:

label=. That the class labels depends on the feature

lbbel=. That the class label is independent of the feature

lcbel=. That the class label correlates with the feature

ldbel=. No conclusion can be drawn

Model Output (Non-reasoning)

Answer: B

DPO Model Output Qwen Based

Answer: The statement " $P(\chi^2 \mid \text{DF} = 1) < 0.05$ " refers to the p-value obtained from the χ^2 (chi-squared) test for a binary feature (i.e., two categories). When this p-value is less than 0.05, it indicates that the observed data is significantly different from what would be expected under the null hypothesis, which is that the class labels are independent of the feature.
Therefore, the correct option is:
D. No conclusion can be drawn - In this case, we cannot draw a definitive conclusion about the relationship between the feature and the class labels, as the given p-value only tells us that the observed data significantly deviates from the null hypothesis, but it does not specify the exact nature of the relationship (e.g., whether the relationship is positive, negative, or has a curvilinear effect).

E.1.2 Reasoning Model Output

Input Prompt

If for the χ^2 statistics for a binary feature, we obtain P(χ^2 |DF = 1) < 0.05, this means:
Options:

label=. That the class labels depends on the feature

lbbel=. That the class label is independent of the feature

lcbel=. That the class label correlates with the feature

ldbel=. No conclusion can be drawn

Let's think step-by-step:

Model Output (Reasoning)

Answer: A. That the class labels depends on the feature.
Explanation: If for the χ^2 statistics for a binary feature, we obtain P(χ^2 |DF = 1) < 0.05, this means that the class labels depend on the feature.
Answer: A

Ground Truth: A