Training an Educational LLM Tutor for EPFL

Albert Fares | 341018 | albert.fares@epfl.ch
Martina Gatti | 341013 | martina.gatti@epfl.ch
Daniel Polka | 326800 | daniel.polka@epfl.ch
Ahmed Abdelmalek | 344471 | ahmed.abdelmalek@epfl.ch
404BotNotFound

Abstract

High-quality tutoring remains inaccessible to many, but large language models (LLMs) offer a scalable alternative. We fine-tune Qwen3-0.6B-Base into an educational assistant using Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO), Retrieval-Augmented Generation (RAG), and model quantization.

Our experiments show RAG boosts accuracy on knowledge-intensive tasks when relevant content is retrievable, DPO improves alignment with human preferences, and quantization enables efficient deployment with minimal performance loss. Combined, these techniques improve both performance and usability of LLM-based tutoring systems.

1 Introduction

Hugging Face's open-source models and datasets enables a practical, experimentation-driven approach to learning natural language processing (NLP). Today, with basic compute access, users can explore a rich ecosystem of pre-trained tools.

This project is a hands-on exploration into the construction of a modular, end-to-end natural language system aimed at solving STEM multiple-choice questions (MCQs). Using pre-trained components and custom fine-tuning, we assemble a pipeline combining supervised learning, preference alignment, retrieval augmentation, and model compression to create an efficient educational assistant.

Rather than focus narrowly on model accuracy alone, we study the roles of each component: does DPO improve rationales? How much does retrieval help? Can quantization retain quality?

Through this modular approach, we aim to enhance MCQ performance and extract broader insights into building educational tools with open-source LLMs.

2 Approach

We develop a modular assistant for solving STEM MCQs at the university level. The system combines supervised answer generation, preference alignment, retrieval augmentation, and quantization to produce accurate and explainable responses to real coursework questions.

Built on Qwen3-0.6B-Base, a 28-layer decoder-only transformer (36T tokens, 32k context), the core MCQA model is fine-tuned to generate answers and rationales. It is enhanced via DPO (reward modeling on pairwise preferences), RAG (retrieval with a dual encoder), and post-training quantization for deployment efficiency.

Modules are trained independently. We evaluate two final variants: a DPO-aligned model with retrieval (Fig. 2, Appendix A) and its quantized counterpart (Fig. 3, Appendix A). Ablations are described in Section 3.2.

MCQA Model. The model is trained via SFT to answer MCQs and produce rationales, which improve reasoning [1, 2]. Each instance consists of a question q and four options $\{o_j\}_{j=1}^{4}$, and the model outputs $y = (a^\star, r)$, where $a^\star$ is the correct answer and r a free-form explanation. Given input $x_i = (q_i, \{o_{i,j}\})$, the model produces $y_i = (a_i^\star, r_i)$, trained with:

$$\mathcal{L}_{\text{MCQA}}(\theta) = -\frac{1}{N}\sum_{i,t} \log p_\theta(y_{i,t} \mid y_{i,<t}, x_i)$$

where T_i is output length. Input tokens are excluded from the loss. Evaluation considers only $a^\star$. Prompt formatting is in Appendix B.

Reward Model. We apply DPO to align generations with human preferences. Each training pair includes a prompt p, preferred response r^+, and less preferred r^-. The reward model assigns scalar scores $f_\theta(p, r)$, and is trained via:

$$\mathcal{L}_{\text{DPO}} = -\frac{1}{N}\sum_{i} \log \sigma(\beta[f_\theta(p_i, r_i^+) - f_\theta(p_i, r_i^-)])$$

with σ the sigmoid and $\beta = 0.1$ controlling preference sharpness.

Quantized Model. We apply post-training quantization to reduce memory and improve deployability, using QLoRA [3] and Optimum-Quanto [4].

QLoRA performs 4-bit NF4 quantization with low-rank adapters. Given weights $\mathbf{W}$, the model is: $\mathbf{W}_{\text{QLoRA}} = Q_{\text{4-bit}}(\mathbf{W}) + \mathbf{AB}$, where $\mathbf{AB}$ is a trainable low-rank update.

Optimum-Quanto applies W4A8 quantization with 4-bit weights (qint4) and 8-bit activations (qint8). Both methods reduce VRAM with minimal performance degradation.

RAG Model. To improve factual accuracy, we incorporate RAG. A dual-encoder retriever selects a relevant document from a fixed corpus based on the prompt. The retrieved snippet is prepended to the input and passed to a decoder for generation. We use bge-base-en-v1.5 [5], fine-tuned using a multiple negatives ranking loss (MNRL) from `SentenceTransformer`:

$$\mathcal{L}_i = -\log\left(\frac{\exp(f(q_i^+, d_i^+))}{\sum_{j=1}^{N}\exp(f(q_i^+, d_j^+))}\right)$$

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i$$

$$f(x, y) = \text{similarity score between } x \text{ and } y$$

3 Experiments

3.1 Data

3.1.1 MCQA Dataset

We unified five training splits of public STEM MCQA datasets into a standardized corpus of questions with four options, one correct answer, and a rationale (human- or GPT-generated). The datasets are SciQ [6], AQUA-RAT [7], MedMCQA [8], ARC-Challenge [9], and OpenBookQA [10] (see Table 1).

Dataset	Raw Size	Used	Rationale Source
SciQ	13,679	10,000	Human-written (filtered)
AQUA-RAT	100,000	80,000	Human-written (filtered)
MedMCQA	272,206	50,000	Human-written (filtered)
ARC-Challenge	2,590	2,590	GPT-generated
OpenBookQA	5,957	5,000	GPT-generated

Table 1: MCQA training set composition after filtering and standardization.

All samples were filtered to ensure format consistency and quality: distractors were sampled when needed, items with external references and low-quality rationales were excluded. ARC-Challenge rationales were generated via few-shot GPT prompting (see Appendix C) and validated using a critique-and-verify pipeline.

We held out 10% from each dataset for validation. The final dataset includes ∼150k training and ∼14k validation examples, with uniform distribution across answer choices (A–D).

3.1.2 DPO Dataset

We trained our reward model on a combined preference dataset from MetaMath FewShot [11] and EvolCodeAlpaca v1 [12], totaling 233,857 samples (Table 2). Each sample includes a prompt with a preferred (`chosen`) and a disfavored (`rejected`) response.

Dataset	Samples	Domain
MetaMath FewShot	199,144	Math (reasoning, equations)
EvolCodeAlpaca v1	34,713	Programming (code generation)
Total	**233,857**	STEM (math + code)

Table 2: DPO preference dataset composition.

MetaMath focuses on math reasoning and EvolCodeAlpaca on code generation. We truncated MetaMath chains to the final question and removed system prompts from EvolCodeAlpaca. All samples were tokenized using the Qwen3 tokenizer, with prompts capped at 400 tokens and total tokens at 800.

3.1.3 Quantized Dataset

We employed two quantization pipelines:

- **Post-Training Quantization (Optimum-Quanto):** Direct quantization of the pre-trained MCQA model without additional training data.
- **QLoRA:** Fine-tuning with LoRA adapters using 15% of the original MCQA dataset to stabilize the model post-quantization.

3.1.4 RAG Dataset

The base embedder performed well, but we fine-tuned it to improve STEM-specific retrieval.

Training Dataset. We combined SciQ, AQUA-RAT, ARC (Easy + Challenge), and PubMedQA [13] (see Table 3).

Dataset	Raw Size	Used
SciQ	11,679	11,679
AQUA-RAT	100,000	20,000
ARC (Easy + challenge)	3,370	3,370
PubMedQA	211,000	20,000

Table 3: Embedder training set composition

Corpus. During inference, we used the training dataset as the corpus, formatting entries as: "*Example of related question Q and answer A: Q:* **(...)** *A:* **(...)**".

3.2 Evaluation Method

We evaluate performance using multiple-choice accuracy. Given a question q and four options $\{o_j\}_{j=1}^4$, the model selects the answer $a^\star \in \{A, B, C, D\}$ with the highest log-probability: $a^\star = \arg\max_{a_j} \log p_\theta(a_j \mid x)$, where $x = (q, \{o_j\})$. We use the `lighteval` framework, based on EleutherAI's LM Evaluation Harness, to compute log-likelihoods and ensure consistent prompt formatting. Evaluation is done under zero-shot (ZS) and few-shot (FS, 2 exemplars) settings (see Appendix D for example prompts), with accuracy measured solely on $a^\star$.

Benchmarks and Model Variants. We evaluate on **MMLU (STEM)** [14] (∼3.3k expert-level MCQs across 16 STEM subjects) and **M1 preference data** (∼645 MCQAs with four+ answers). We compare eight configurations to isolate enhancement impacts:

- **Base**: Qwen3-0.6B-Base (no fine-tuning)
- **MCQA**: Supervised fine-tuned model
- **MCQA + DPO**: MCQA aligned via DPO
- **Quantized MCQA**: Compressed MCQA
- **RAG + MCQA**: Retrieval-augmented MCQA
- **RAG + Aligned**: RAG + DPO-aligned model
- **RAG + Quantized MCQA**: RAG + quantized MCQA
- **RAG + Quantized + Aligned**: Full system combination

3.3 Baselines

As a baseline, we evaluate the pre-trained Qwen3-0.6B-Base without fine-tuning, under both ZS and FS settings using the setup in Section 3.2. In ZS, the model sees only the question and options; in FS, two example QA pairs are prepended (see Appendix D). In both cases, the answer $a^\star$ is selected via log-likelihood. This baseline helps us quantify the benefits of fine-tuning, DPO alignment and retrieval in our modular system.

3.4 Experimental details

3.4.1 MCQA Model Configuration and Training Protocol

We fine-tune Qwen3-0.6B-Base for STEM multiple-choice QA using a 4-stage progressive strategy to incrementally increase corpus diversity and domain coverage.

Training Configuration. The model generates both the correct answer and brief rationale following the format: `<Letter>. <Option> Explanation: <1-2 sentence rationale> <eos>`. Only answer and rationale tokens contribute to loss (prompt tokens masked with $\ell = -100$). We apply proportional interleaving with fixed mixing probabilities inversely proportional to dataset size to prevent larger corpora from dominating. Training uses a single NVIDIA A100-40GB GPU with bfloat16 precision, 8-bit AdamW (learning rate 4×10^{-6}, weight decay 0.10), linear warmup over 8% of steps, cosine decay, dropout 0.15, and 384-token context length. Gradient accumulation (8 steps) with batch size 8 totals 64 sequences per update.

Training Stages and Results. We adopt a 4-stage process with incremental corpus introduction (Table 4). Stage 1 (AQUA datasets only) achieved the highest ZS accuracy (46.10%), while Stage 3 (full interleaved mix of all five MCQA datasets) delivered a strongest FS performance (45.98%) and the broadest generalization. Stage 4 variants exploring MathQA and ARC-Easy additions showed modest degradation, likely due to domain shift and annotation inconsistencies. We select **Stage 3 (Full Mix)** as our final model for its robustness, strong FS accuracy, and broad domain coverage suitable for real-world deployment.

Stage	Datasets	Samples	ZS	FS	Notes
Base	-	-	45.46	49.28	Baseline
1	AQUA	80K	**46.10**	**46.00**	Initial reasoning
2	+SciQ	92K	45.20	**46.00**	Add science
3	+MMCQA[1], ARC, OBQA[2]	331K	44.47	45.98	Full interleaved
4a	+MathQA	427K	43.50	44.90	Symbolic math
4b	+ARC-Easy	334K	44.30	45.60	Factual QA

[1]MedMCQA; [2]OpenBookQA

Table 4: Training stages, dataset composition, and MMLU-STEM accuracy (%).

3.4.2 Reward Model

We conducted iterative experiments to optimize our DPO training approach using Qwen3-0.6B-Base as both policy and reference model.

Dataset Evolution. Our initial experiment (Model 1) used 10,000 MetaMath samples but showed no improvement. We then explored incorporating M1 preference data but found substantial label noise. Model 2 used our filtered 69,425-sample dataset combining MetaMath and EvolCodeAlpaca, showing improvements. Our final Model 3 utilized the complete 233,857-sample dataset for comprehensive training.

Results. Model 3 achieved the best performance with 59.87% average accuracy, representing a 0.54

percentage point improvement over the base model. Training configurations are in Table 5, with benchmark results in Table 6.

Table 5: Training Parameters Comparison

Parameter	Model 1 (10k)	Model 2 (69k)	Model 3 (187k)
Learning Rate	5e-06	1e-05	1e-05
Beta (DPO Temp.)	0.1	0.3	0.3
Batch Size	16	16	32
Grad. Accum. Steps	8	16	8
f-DPO	-	0.1	0.1
Epochs	3	1	1
Optimizer	ADAM	ADAM	ADAM
Weight Decay	0	0	0
Warmup Ratio	0.1	0.1	0.1

Table 6: Reward Model Performance Comparison (**best scores in bold**)

Benchmark (Samples)	**Base (Qwen3)**	**Model 1 (10k)**	**Model 2 (69k)**	**Model 3 (187k)**
HH-RLHF (1,000)	44.1%	**45.0%**	43.0%	42.7%
UltraFeedback (1,000)	54.6%	54.0%	**55.3%**	**55.3%**
SHP (1,000)	54.9%	54.6%	**55.8%**	55.7%
PKU Safe (500)	44.2%	**45.2%**	44.6%	45.4%
RewardBench (2,985)	62.4%	**63.4%**	63.2%	63.0%
MNLP M3[1] (23,385)	97.9%	97.9%	99.2%	**99.3%**
MNLP M1[2] (24,240)	57.2%	57.1%	57.5%	**57.7%**
Average	59.33%	59.60%	59.80%	**59.87%**

[1]Test split from Section 3.1.2; [2]Student-generated preference data from M1

3.4.3 Quantization

We applied two post-training quantization strategies to compress our models while preserving performance.

QLoRA (W4A16). Starting from the aligned model, we used QLoRA [3] with 4-bit `nf4` weights and bfloat16 activations with double quantization enabled for stability. LoRA adapters were trained on 15% of MNLP-M3 (1 epoch) with attention module targets. After merging adapters, we exported the model in W4A16 format for evaluation and deployment.

Optimum-Quanto (W4A8). For comparison, we quantized the MCQA model (without DPO) using Optimum-Quanto [4], producing a static W4A8 model via qint4/qint8 compression. This faster, smaller model skips adapter training but lacks preference alignment.

Evaluation setup. Both models were evaluated on the MCQA benchmarks with the same setup. We report top-1 accuracy, peak VRAM, and a quantization score to compare efficiency-performance tradeoffs.

3.4.4 RAG

As discussed in Section 2, we used MNRL to fine-tune the embedder, with special focus on corpus quality. Because each document is prepended verbatim to the prompt, clarity and content relevance were essential. We used cosine similarity for both training and inference due to its scale invariance, avoiding length-based retrieval bias.

Corpus and Dataset. The training data had three fields: `question`, `context` (answer or hint) and `source`. We curated a balanced corpus to broaden topic coverage:

- **AQUA-RAT** (Algebra, 20k): Subset of cleaned MCQA training data, reduced via KMeans to avoid over-representing math.
- **PubMedQA** (Biomedical, 20k): Reduced with KMeans from the `pqa_artificial` subset of 211k samples.
- **SciQ + ARC** (Natural sciences, 15k): Provided non-math, non-medical content (physics, chemistry, biology).

KMeans Filtering. For AQUA-RAT and PubMedQA, we filtered out samples over 512 tokens, embedded the rest with bge-base-en-v1.5, clustered them using FAISS KMeans and then retained only the nearest sample from each cluster (20k total).

Corpus Size. Our final corpus had 55k documents, which was under the 100k limit but emphasized topic balance over size. While we did not verify this hypothesis, we believe this trade-off improved retrieval robustness.

3.4.5 Merging the MCQA and Reward models

To leverage the complementary strengths of both the SFT MCQA model and the DPO-aligned preference model, we employed the Task-specific Incremental Ensemble Selection (TIES) merging algorithm [15]. TIES enables the combination of multiple fine-tuned models by identifying and resolving conflicts between model parameters while preserving task-specific capabilities. We conducted systematic experiments with different weight combinations using a TIES merging coefficient of 0.8 (Table 7). We also tried retraining DPO using the SFT model as base. Model 1, with weights of 0.3 SFT and 0.7 DPO, achieved the highest ZS MMLU accuracy.

4 Results

We report accuracy results for all model variants under both ZS and FS prompting on MMLU-STEM and M1 Data benchmarks in Table 8 . While our deployed systems use **RAG + Aligned** and **RAG +**

Table 7: Model Merging Experiments using TIES Algorithm (**best scores in bold**)

Model	SFT Weight	DPO Weight	TIES Coeff.	0-shot MMLU	FS MMLU
Model 1	0.3	0.7	0.8	**48.12%**	48.71%
Model 2	0.5	0.5	0.8	47.69%	**48.78%**
Model 3	0.6	0.4	0.8	47.35%	48.47%
DPO Retrain	–	–	–	46.36%	47.33%

Quantized + Aligned configurations, we include intermediate models to isolate each components' individual contributions. Per-subject MMLU-STEM breakdowns are provided in Appendix F (Tables 9 and 10).

Model Variant	MMLU ZS (%)	MMLU FS (%)	M1 Data ZS (%)	M1 Data FS (%)
Base	45.40	**49.05**	39.38	**45.74**
MCQA	44.47	45.98	41.4	42.96
MCQA + DPO	**48.12**	48.71	43.57	45.12
Quantized (MCQA)	40.07	41.84	37.83	40.78
Quantized (MCQA + DPO)	39.94	44.63	39.07	38.22
RAG + MCQA	44.32	45.37	41.71	43.41
RAG + (MCQA + DPO)	47.06	48.67	**43.72**	44.96
RAG + Quantized (MCQA)	42.09	43.10	36.74	40.93
RAG + Quantized (MCQA + DPO)	41.69	44.17	36.74	37.98

Table 8: Accuracy under ZS and FS prompting on MMLU-STEM and M1 Data. Best results in each column are bolded.

Per-Category Performance. To analyze model behavior across different STEM domains, Figure 1 shows detailed ZS and FS accuracy heatmaps for all model configurations, broken down by the general subject category of the questions in the MMLU STEM subset. For a detailed view of accuracy variations relative to base models, see figures 4a and 4b in Appendix E.

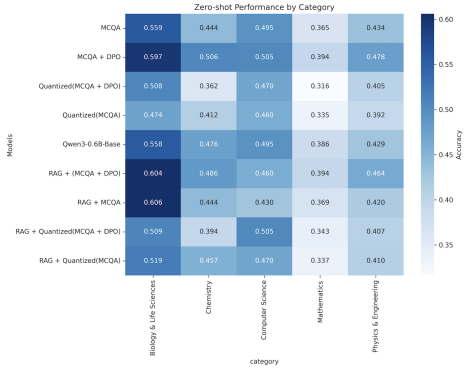

Figure 1: ZS accuracy of all model configurations across STEM subject categories (MMLU-STEM subset).

5 Analysis

We analyze how fine-tuning, alignment, retrieval and quantization affect model performance across prompting setups and subject categories.

Effect of Fine-tuning. MCQA fine-tuning slightly reduces MMLU-STEM accuracy (45.40%→44.47%) but improves M1 Data (39.38%→41.40%) (Table 8), likely due to domain mismatch. The dataset (Table 1) covers core STEM areas but lacks symbolic math and multi-step reasoning. As shown in Figure 1, gains appear in `Biology & Life Sciences` (55.8%→55.9%), while `Mathematics` drops (38.6%→36.5%) due to weak coverage and noisy labels. Overall, fine-tuning helps in-domain tasks but may hurt robustness in underrepresented areas.

Effect of Alignment via DPO. DPO improves ZS accuracy over MCQA-only on MMLU-STEM (44.47%→48.12%) and M1 Data (41.40%→43.57%) (Table 8), driven by structured reasoning in the DPO corpus. While not MCQ-specific, DPO enhances rationale coherence, complementing MCQA's format and improving factual accuracy. Largest gains occur in `Physics & Engineering` (43.4%→47.8%) and `Computer Science` (49.5%→50.5%), reflecting DPO's focus on logical and code-oriented reasoning.

Effect of RAG. RAG gives mixed results when added to aligned models. It boosts `Biology & Life Sciences` but hurts `Computer Science` (50.5%→46.0%) (Fig. 1), likely due to retriever bias toward factual QA (**SciQ**, **PubMedQA**, **AQUA-RAT**) and weak symbolic/code coverage. On MMLU-STEM, RAG underperforms MCQA + DPO but matches it on M1 (Table 8), showing domain sensitivity. It also recovers 1.75 of the 8.18-point quantization loss (Appendix E, Fig. 4), suggesting value in offsetting low-bit degradation when well-aligned.

Quantization Trade-offs. We evaluate two post-training setups for the aligned MCQA model: W4A16 (QLoRA) and W4A8 (Optimum). Both reduce memory but cause accuracy drops, e.g. Quantized MCQA falls from 45.98% to 41.84% on MMLU-STEM; RAG + Quantized MCQA drops to 43.10% (vs. 45.37%). Losses are larger when combined with alignment and retrieval: RAG + Quantized (MCQA + DPO) scores 44.17%, down 4.5 points. While quantization enables efficient inference, it can hurt performance on reasoning-heavy tasks.

FS prompting improves accuracy, with gains varying by model. The Base model benefits most (+3.65% MMLU-STEM, +6.36% M1), relying on

examples for reasoning. MCQA and MCQA + DPO show smaller gains (+1.51%, +0.59%), indicating reduced FS dependence due to fine-tuning and alignment. On M1, MCQA + DPO gains +1.55%, while some quantized models worsen. Overall, FS helps less-specialized models, while fine-tuned ones perform well in ZS settings.

Qualitative Rationale Comparison. We compare model rationales on an **NLP4Education** [16] question where the correct answer is `C` (8 MB); full outputs are in Appendix H. The Base model selects `A` with flawed logic, miscomputing 2^{31} entries by confusing entry size with address space. MCQA picks `B`, using nonsensical formulas like "Page Size = 2^{32} Bytes" and somehow concluding 4 MB from 2^{48}B, showing terse but incoherent math. Base is verbose yet confused; MCQA is brief but illogical.

MCQA + DPO selects `C` with valid reasoning: it identifies 64-bit entries (20+12+32), computes 2^{32} entries, and correctly calculates 8 MB. This reflects DPO's impact on structured reasoning from sources like **MetaMath**, improving problem-solving even when only final answers are scored.

Summary and Deployment Rationale. While MCQA + DPO has the highest ZS accuracy (48.12%) and the Base model leads in the FS setting (49.05%), we prioritize models balancing performance and deployability. **Aligned + Retrieved** (RAG + MCQA + DPO) offers near-optimal accuracy (47.06% ZS, 48.67% FS) and strong factual robustness, making it ideal for full-precision use. **Aligned + Quantized + Retrieved** trades some accuracy for efficiency: accuracy in a ZS setting drops by 8.18 points, but using RAG recovers 1.75 (Fig. 4). These configurations support practical deployment, emphasizing generalizability, interpretability and efficiency over peak scores.

6 Ethical considerations

We briefly highlight ethical considerations related to model training, datasets, intended use and language adaptation.

Ecological Impact. The impact of prompting and training is non-trivial: models on the scale of GPT-4o are estimated to consume between 20–25 MWh of electricity during training [17]. While our models are smaller, we did not keep an eye on our eletricity usage. Future work in academic LLM use could consider ways to limit redundant computation and promote energy consumption monitoring.

Data Anonymity. All datasets used are public and research-oriented. To our knowledge, they contain no sensitive personal data. Any names appear in public or citation contexts, making privacy concerns minimal.

Application Ethics. Our system supports learning in STEM, but should not be treated as an authoritative source, especially in high-stakes fields like medicine or engineering. Users must critically evaluate its outputs. Furtheremore, the model should not be used for cheating during exams.

Language Adaptation. While our system was developed for English, it could be adapted to high-resource languages by fine-tuning on high-resource language MCQA datasets. Adaptation for low-resource languages could be achieved with multilingual pretraining after data augmentation, though reinforcing existing data and performance disparities would be a risk.

7 Conclusion

In this project, we developed a modular and scalable AI tutor system based on the Qwen3-0.6B-Base model. By combining SFT (MCQA), preference alignment (DPO), low-bit quantization and RAG, we evaluated each component's impact on multiple-choice STEM benchmarks.

Our results show that DPO alignment delivers the largest gains on reasoning tasks, while retrieval with a well-curated corpus improves factual recall in knowledge-heavy domains. Quantization introduces some accuracy loss but enables lightweight deployment without changing the model architecture, which is a valuable trade-off in resource-limited settings. The combined system shows alignment and retrieval to be complementary, whereas compression must be applied carefully to preserve good reasoning and retrieval behavior.

Our findings also emphasize the importance of data diversity and quality, both for supervised training and retrieval corpora. Performance varied with the structure and distribution of training data, highlighting the need for more targeted dataset design and domain balance. Future work could further improve retrieval by enriching the corpus with curriculum-aligned content and optimizing for long-context inputs.

References

[1] Pan Lu, Maximilian Bartolo, Sebastian Riedel, et al. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[2] Eric Zelikman, Yuhuai Wu, Noah Goodman, and Christopher D Manning. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, 2022.

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

[4] Hugging Face. Optimum-quanto: Efficient model quantization for transformers. https://github.com/huggingface/optimum-quanto, 2024. Accessed: 2025-06-07.

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

[6] Johannes Welbl, Nelson F Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 2017.

[7] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of ACL*, page 158–167, 2017.

[8] Ankit Pal, Mrittika Bhatia, Koustuv Dey, et al. Medmcqa: A large-scale medical multiple-choice question answering dataset. *Nature Scientific Data*, 2023.

[9] Peter Clark, Isaac Cowhey, Oren Etzioni, et al. Think you have solved question answering? try arc, the ai2 reasoning challenge. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[10] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

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

[12] Aleksey Korshuk. Evolcodealpaca: A self-evolving code generation dataset, 2023.

[13] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics.

[14] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[15] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models, 2023.

[16] Beatriz Borges, Negar Foroutan, Deniz Bayazit, Anna Sotnikova, Syrielle Montariol, Tanya Nazaretzky, Mohammadreza Banaei, Alireza Sakhaeirad, Philippe Servant, Seyed Parsa Neshaei, Jibril Frej, Angelika Romanou, Gail Weiss, Sepideh Mamooler, Zeming Chen, Simin Fan, Silin Gao, Mete Ismayilzada, Debjit Paul, Philippe Schwaller, Sacha Friedli, Patrick Jermann, Tanja Käser, Antoine Bosselut, EPFL Grader Consortium, and EPFL Data Consortium. Could chatgpt get an engineering degree? evaluating higher education vulnerability to ai assistants. *Proceedings of the National Academy of Sciences*, 121(49):e2414955121, 2024.

[17] Josh You. How much energy does chatgpt use? https://epoch.ai/gradient-updates/how-much-energy-does-chatgpt-use, February 2025. Accessed: 2025-06-06.

Appendix

A Model pipeline diagrams

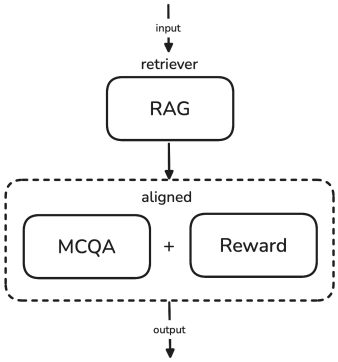

Figure 2: Retrieval-Augmented Aligned (AR) model: DPO-aligned MCQA receives the question and retrieved context.

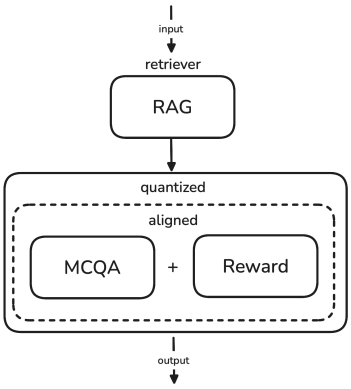

Figure 3: Aligned Quantized + Retrieved (AQR) model: DPO-aligned MCQA is quantized post-training and used with retrieval.

B MCQA Training Examples

Example 1

Prompt: The following are multiple-choice questions (with answers) about advanced STEM knowledge.
Question: What is the derivative of $\sin(x)$?
Options:

A. $-\cos(x)$

B. $\cos(x)$

C. $\sin(x)$

D. $-\sin(x)$

Answer: B. $\cos(x)$
Explanation: The derivative of $\sin(x)$ is $\cos(x)$ by standard differentiation rules.

C Rationale Generation Prompts

ARC-Challenge Prompt

The following prompt was used to generate concise, single-sentence rationales for ARC-Challenge questions, emphasizing factual correctness and minimal verbosity.

ARC-Challenge Prompt

You are a science tutor. For each multiple-choice question you will give ONE short sentence that **explains why the correct option is correct**.

- Do **not** reveal the correct letter.
- Do **not** mention the other options.

Keep it crisp, factual and no more than 25 words.
Example:
Q: Why does metal feel colder than wood at the same temperature?
A) Metal is colder. B) Wood stores more heat. C) Metal conducts heat faster. D) Wood conducts heat faster.
Answer: C
Explanation: Metal conducts heat away from your skin efficiently, so it removes body heat faster and feels colder.
Instruction: Now I'll give you a new question, choices and the correct letter. Return only the explanation sentence.

D Prompt format for Evaluation

We show examples of the exact prompts used during evaluation. Each prompt ends with `Answer:` and the model is expected to output an answer in the form `<Letter>. <Option>` optionally followed by an explanation.

ZS prompt (no examples):

ZS Prompt

Question: Which part of the cell is responsible for producing energy?
A. Nucleus B. Mitochondrion C. Ribosome D. Golgi apparatus
Answer:

FS prompt (2 examples):

FS Prompt (2 Examples)

The following are multiple choice questions (with answers) about STEM topics.
Question: What is the boiling point of water at sea level?
A. 90°C B. 100°C C. 110°C D. 120°C
Answer: B. 100°C
Explanation: Water boils at 100°C at standard atmospheric pressure.
Question: What gas do plants absorb during photosynthesis?
A. Oxygen B. Nitrogen C. Carbon dioxide D. Hydrogen
Answer: C. Carbon dioxide
Explanation: Plants absorb carbon dioxide from the air to make food.
Question: Which part of the cell is responsible for producing energy?
A. Nucleus B. Mitochondrion C. Ribosome D. Golgi apparatus
Answer:

The FS prompt includes two full QA+explanation examples formatted as during training. During inference, models may output just the answer or include a short rationale, but only the selected answer token is used to compute accuracy.

E Detailed Accuracy differences by model

This appendix provides a detailed view of accuracy variations relative to base models using heatmaps. The figures below show the relative gains or losses induced by each technique combination under ZS and FS prompting, respectively.

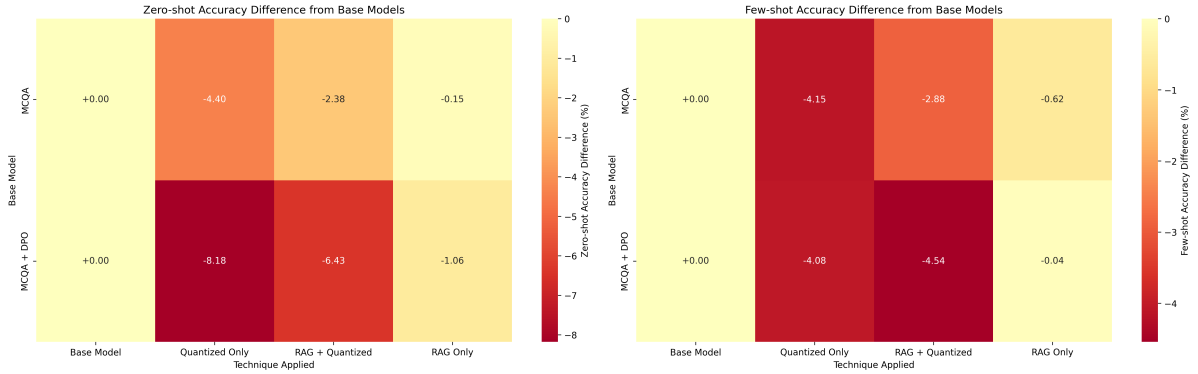

Figure 4: Relative accuracy differences (%) from base models under ZS (left) and FS (right) settings.

F Per-Subject ZS Accuracy on MMLU-STEM (Extended)

	Standard Models			Quantized		RAG Models			
Subject	**Base**	**MCQA**	**MCQA+DPO**	**MCQA**	**MCQA+DPO**	**MCQA**	**Q+MCQA**	**MCQA+DPO**	**Q+MCQA+DPO**
Abstract Algebra	30.00	25.00	29.00	24.00	27.00	26.00	20.00	**38.00**	36.00
Anatomy	43.70	47.41	51.85	43.70	49.63	**56.30**	47.41	51.85	45.19
Astronomy	53.29	52.63	**57.24**	50.00	50.00	55.26	53.29	**57.24**	48.36
College Biology	57.63	56.94	58.33	47.22	47.92	59.72	53.47	**60.42**	52.78
College Chemistry	43.00	41.00	**52.00**	39.00	36.00	42.00	45.00	44.00	36.00
College Computer Science	43.00	42.00	43.00	40.00	**46.00**	36.00	42.00	40.00	**46.00**
College Mathematics	39.00	30.00	36.00	31.00	**41.00**	31.00	29.00	32.00	31.00
College Physics	27.45	35.29	**38.24**	27.45	24.51	29.41	36.27	37.25	28.43
Electrical Engineering	**57.93**	52.41	57.24	48.97	53.79	45.52	42.76	57.24	52.41
Elementary Mathematics	44.71	44.18	**45.77**	41.53	39.68	44.71	41.80	43.92	40.48
High School Biology	66.13	63.23	**69.03**	51.29	54.84	65.81	54.84	**69.03**	54.84
High School Chemistry	52.22	47.78	49.26	43.35	46.80	36.45	46.31	**53.20**	42.86
High School Computer Science	56.00	57.00	**58.00**	52.00	48.00	50.00	52.00	52.00	55.00
High School Mathematics	35.19	36.30	**37.41**	35.93	31.85	35.56	35.37	36.67	31.48
High School Physics	33.11	33.11	**38.41**	30.46	33.77	37.75	31.79	33.77	33.77
High School Statistics	43.98	47.22	**49.07**	35.19	18.52	47.22	42.13	46.30	32.41
Average	45.40	44.47	**48.12**	40.07	39.94	44.32	42.09	47.06	41.69

Table 9: Per-subject ZS accuracy on MMLU-STEM. Bold values indicate best performance per subject.

G Per-Subject FS Accuracy on MMLU-STEM (Extended)

	Standard Models			Quantized		RAG Models			
Subject	**Base**	**MCQA**	**MCQA+DPO**	**MCQA**	**MCQA+DPO**	**MCQA**	**Q+MCQA**	**MCQA+DPO**	**Q+MCQA+DPO**
Abstract Algebra	36.00	34.67	33.50	30.50	35.00	34.33	**36.50**	36.33	33.50
Anatomy	51.11	50.62	49.63	**51.85**	47.41	50.12	**54.81**	49.14	50.00
Astronomy	54.61	51.54	54.93	47.04	51.32	53.51	49.34	**58.33**	52.30
College Biology	63.19	62.27	**65.28**	52.78	55.21	62.04	54.51	63.43	55.56
College Chemistry	48.00	42.00	45.50	37.00	39.00	**46.33**	38.50	45.33	35.00
College Computer Science	43.00	42.00	41.50	41.00	**44.50**	40.67	40.00	45.33	42.50
College Mathematics	37.00	34.33	**41.00**	31.00	34.50	28.33	30.00	40.00	33.50
College Physics	32.84	32.03	**32.84**	**32.84**	31.37	29.08	**34.80**	34.31	31.86
Electrical Engineering	**63.10**	54.71	62.76	48.62	59.31	52.18	48.97	54.71	57.93
Elementary Mathematics	45.50	46.21	47.35	44.31	42.72	45.59	42.99	**47.53**	41.00
High School Biology	**70.81**	64.62	70.48	55.81	66.45	63.33	55.65	69.14	65.48
High School Chemistry	**54.19**	48.77	51.23	43.60	46.80	46.80	45.81	52.71	44.09
High School Computer Science	**62.00**	57.00	60.00	52.00	58.00	59.33	54.50	60.00	56.00
High School Mathematics	**36.85**	34.57	40.37	31.67	31.11	34.81	32.41	33.33	32.41
High School Physics	35.43	32.89	33.44	30.46	31.13	33.11	30.79	**36.20**	31.79
High School Statistics	51.16	47.53	49.54	38.89	40.28	46.30	40.05	**52.93**	43.75
Average	**49.05**	45.98	48.71	41.84	44.63	45.37	43.10	48.67	44.17

Table 10: Per-subject FS accuracy on MMLU-STEM. Bold values indicate best performance per subject.

H Rationale Analysis

This appendix provides full model outputs for the qualitative rationale comparison presented in Section 5. We show the rationales generated by three system variants (Base model, MCQA, MCQA + DPO) on an **NLP4Education** [16] multiple-choice question from the domain of Computer Architecture, where the correct answer is `C` (8 MB). The prompt is reproduced below for reference.

Prompt

The following are multiple choice questions (with answers) about knowledge and skills in advanced master-level STEM courses.
Question: Assume a 32-bit architecture with a single-level page table. Each entry has 20 bits for PPN, 12 bits for permissions, and extra 32 bits used by the OS to store the access time of the page table entry. What is the size of page table required if a process uses all of its memory? (2 pts).
A. 2 MB
B. 4 MB
C. 8 MB
D. 16 MB
Answer:

Base Model Output

The size of the page table can be calculated as follows:
- The number of entries in each page table entry is given by Number of entries $= 2^{(20+12)} = 2^{32}$.
- Since there are 2 pages per entry (one for physical page number and one for permission), the total number of entries needed is $\frac{2}{2} \times 2^{32} = 2^{31}$.
- Therefore, the total size of the page table is 2^{31} bytes or 2 MB.
So, the correct answer is: **A. 2 MB**

MCQA (SFT) Output

B. 4 MB
Explanation: The page table contains information that allows the processor to navigate between different addresses in main memory and disk storage. In this case, the size of the page table can be calculated as follows: Page Size = 2^{32} Bytes Page Table Entry = 2^{32} + 2^{12} + 2^{32} = 2^{44} Bytes Total Page Table Size = Number of Pages * Page Size = 2^{44} * 2^{12} / 2^{32} = 2^{48} Byte = 4MB

MCQA + DPO Output

C. 8 MB
Explanation: The total number of bits available for each page table entry is 20 (PPN) + 12 (permissions) + 32 (access time) = 64 bits. With a 32-bit address space, we can create 2^{32} page tables. Therefore, the size of the page table required is $2^{32} \times 64$ bytes, which equals 8 megabytes.

I AI Usage Appendix

Per the course AI policy, we disclose our use of AI-based tools throughout this project.

I.1 AI Tools Used

We utilized three large language models for code development and testing:

- **GPT-4o** (OpenAI): Primary assistant for training pipeline development
- **GPT-o3** (OpenAI): Advanced reasoning for complex algorithmic implementations
- **Claude Sonnet 4** (Anthropic): Code review and debugging assistance

I.2 Usage Scope

AI tools were employed for the following components:

Training Code Development. We used AI assistance to implement training loops for supervised fine-tuning (MCQA), DPO alignment, quantization and RAG workflows. This included data preprocessing pipelines, loss function implementations, and hyperparameter configuration scripts.

Evaluation and Testing Infrastructure. AI tools helped develop evaluation scripts for computing multiple-choice accuracy, implementing the `lighteval` framework integration, and creating benchmark comparison utilities.

RAG Implementation. AI assisted in implementing the retrieval-augmented generation pipeline, including embedder fine-tuning code, corpus preprocessing scripts, and similarity-based retrieval mechanisms.

Model Merging and Quantization. AI tools also helped implement the TIES merging algorithm and quantization workflows (QLoRA and Optimum-Quanto), including parameter conflict resolution and adapter training scripts.

Report. We also used AI to generate the logo used at the top of the report and to help distill the paragraphs into more concise formulations.

I.3 Verification Methods

We verified AI-generated code correctness through multiple approaches:

- **Manual Code Review**: All AI-generated code was manually reviewed and understood before integration
- **Result Comparisons**: Comparing results across different model configurations and datasets
- **Literature Verification**: Ensuring implementations matched published methodologies and expected behaviors

I.4 Limitations and Human Oversight

While AI tools significantly accelerated development, the final code decisions, experimental design and result interpretations were performed by us. AI suggestions were treated as starting points that required validation and debugging to meet project requirements. The main research contributions, results analysis and conclusions represent original human work.