A question about STEM ? Just ask Qwen !

Mathis Krause | 328110 | mathis.krause@epfl.ch
Anoush Azar-Pey | 326887 | anoush.azar-pey@epfl.ch
Mathieu Sauser | 328777 | mathieu.sauser@epfl.ch
Emilien Silly | 341507 | emilien.silly@epfl.ch
MNLP Final Report; Team GOAT

Abstract

We present four models derived from the Qwen3-0.6B-Base language model, targeting two distinct tasks: STEM multiple-choice question answering (MCQA) and preference-based instruction tuning via Direct Preference Optimization (DPO). Despite constraints in computational resources and time, our methods yield a 5% absolute accuracy gain on MCQA and a 35% improvement when incorporating Retrieval-Augmented Generation (RAG). To overcome resource limitations, we employ several targeted strategies, including answer choice shuffling for data augmentation, staged supervised fine-tuning (SFT), and adaptive learning rate scheduling. For RAG, we construct a high-quality retrieval corpus by manually curating support documents tailored to challenging questions and by synthesizing factual knowledge for enhanced context retrieval. Our DPO model is explicitly aligned for STEM reasoning tasks, trained on over 300,000 high-quality preference pairs, with negative examples generated using Gemini-Flash to ensure plausibility. Results emphasize the importance of alignment data quality, task-specific fine-tuning, and targeted retrieval strategies in developing lightweight yet performant language models for STEM domains.

1 Introduction

Small language models (SLMs) offer advantages in cost and speed, making them well-suited for educational and domain specific tasks. However, ensuring their accuracy and consistency especially in complex domains like STEM remains difficult. Although pretrained on STEM data, most models lack explicit optimization for reasoning, factual correctness, or formats like multiple-choice QA.

This project aligns Qwen3-0.6B-Base (Yang et al., 2025) for STEM tasks through a multi-stage process. We begin with supervised fine-tuning (SFT) on a filtered STEM Wikipedia subset and the SCP-116k dataset. We then apply preference optimization—selecting Direct Preference Optimization (DPO) after empirical comparison with KTO and CPO.

In parallel, we improve multiple-choice QA using explanation-aware formatting and data augmentation, and enhance retrieval through fine-tuning of the document encoder for our RAG system. Overall, these strategies significantly improve performance on STEM benchmarks, highlighting the value of high-quality alignment, task-specific tuning, and retrieval-based enhancement.

2 Approach

Our methodology has two main stages. First, we extend pretraining with supervised fine-tuning (SFT) on curated high-quality STEM texts and reasoning datasets. Second, we perform task-specific fine-tuning on carefully prepared MCQA and Direct Preference Optimization (DPO) datasets. Figure 1 shows the pipeline, with intermediate models in blue and final submissions in green.

Although Qwen3-0.6B-Base was pretrained on 5 trillion tokens including STEM content (Yang et al., 2025), subsequent pretraining to extend the context window to 32k tokens may have reduced domain precision. We addressed this by reintroducing focused STEM data: a curated STEM Wikipedia subset for broad knowledge and SCP-116k for reasoning quality. This improved performance over the baseline.

In fine-tuning, we used high-quality MCQA and DPO data, combining external sources with our M1 preference pairs. We optimized hyperparameters and training objectives to maximize capability.

Finally, our Retrieval-Augmented Generation (RAG) system enhances the generator with self-created support documents, and we fine-tuned the embedding model to boost retrieval accuracy.

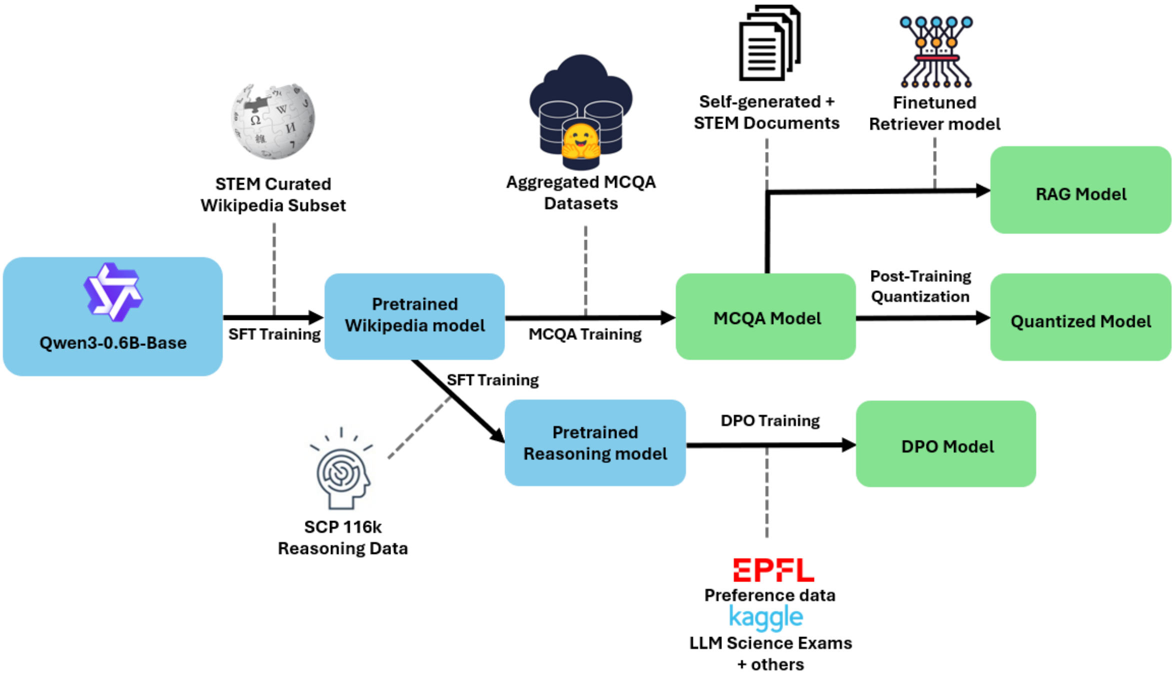

Figure 1: Model creation pipeline; Green boxes: Submitted models; Blue Boxes: intermediate models

3 Experiments

3.1 Datasets

This section outlines the datasets created for our model's alignment stages. Some format example can be found in the appendix.

3.1.1 SFT-Datasets

To complete the pretraining phase of our model, we employed two high-quality supervised fine-tuning (SFT) datasets.

First, we curated a STEM-specific subset of the Wikipedia dataset (Johnson et al., 2024). This was accomplished by applying keyword-based filtering: articles were retained if they contained at least one STEM-related term (e.g., algorithm, biological), while articles containing non-relevant terms (e.g., biography, TV show) were excluded. This filtering process reduced the dataset from approximately 6 million articles to a focused set of 20,000 high-quality STEM documents.

Second, we incorporated the SCP-116k dataset (Lu et al., 2025), a collection of complex STEM questions and detailed answers generated by the DeepSeek R1 reasoning model. While we did not utilize the accompanying reasoning traces, the long-form answers—driven by high-quality reasoning—serve as a valuable signal. These responses are particularly beneficial for training our Direct Preference Optimization (DPO) model, which must assess the quality of answers to challenging questions.

3.1.2 DPO Dataset

We use different datasets across our experiments. All datasets follow the standard format required by preference optimization: each sample includes a prompt, a preferred answer, and a rejected alternative.

Below is an overview of the datasets used in each experiment.

Comparison of DPO, KTO, and CPO:
Preference pairs generated for Milestone 1.

Reproduction of SmolLM2 alignment setup:
`metamathqa-50k` subset of the SmolTalk dataset (Allal et al., 2025), followed by training with code related UltraFeedback(Bartolome et al., 2023).

Large-scale DPO training:
We constructed a large-scale STEM-focused preference dataset. Where possible, we reused existing datasets with native pairwise preference structure. For datasets that only contained question–answer pairs, we converted them into preference-formatted data by keeping the original answer as the preferred response and generating a rejected alternative using `google/gemini-2.0-flash-lite-001`. Each prompt included the original question along with its correct answer, and the model was instructed to produce a semantically plausible but incorrect

response from the true answer.

The sources cover a broad range of STEM domains, including mathematics, computer science, chemistry, physics, medicine, and general science. A complete list of datasets is provided in Appendix A.3.

This aggregation yielded over 1.4 million entries. To prevent over-representation of any single field—especially chemistry—we excluded more than 600 k samples from `AI4Chem/ChemData700K` because this dataset contains over 700k entries and would otherwise make the overall dataset overly biased toward chemistry.. The resulting pool contained roughly 680k entries. As full training on this volume was computationally impractical, we sampled one entry out of every two, resulting in a final training set of 340k preference pairs.

3.1.3 MCQA Dataset

We construct a MCQA training dataset by reformatting and concatenating AQUA-RAT (Ling et al., 2017), AI2 ARC (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), SciQ (Johannes Welbl, 2017), the Kaggle LLM Science Exam (Will Lifferth and Howard, 2023) and PAL-2 (Pal et al., 2022), and a MCQA evaluation dataset from MMLU (Hendrycks et al., 2021b), ETHICS (Hendrycks et al., 2021a) and AI2 ARC (Clark et al., 2018).

We preprocess the merged 43k questions by downsampling AQUA-RAT to 20k examples; converting each into a (question, choices, correct-index) tuple (see Appendix A.2); tokenizing with qwen; truncating/padding to 512 tokens; encoding correct answers as integer labels.

Similarly, we construct a MCQA evaluation dataset to evaluate our models' generality.

3.1.4 RAG Dataset

Our RAG documents system relies on two main data sources: (1) high-quality existing corpora, including filtered Wikipedia chunks on STEM topics and MCQA datasets' "lecture" fields such as QASC (Khot et al., 2020) and ScienceQA (Lu et al., 2022); and (2) synthetic documents generated via the ‘gptWrapper‘ (from Milestone 1) for questions the model answered incorrectly. These documents were created by prompting the wrapper with the correct answer and asking it to generate contextual explanations of relevant keywords, aiming to enrich model understanding without leaking the answer. While this approach is inherently limited to the questions used for generation, we mitigated this by maximizing topic coverage across STEM domains.

3.2 Evaluation Method

We evaluate our MCQA model using the *lighteval* framework, which measures accuracy by computing the single-token log-likelihoods of the four answer choices. A prediction is correct if the model assigns the highest probability to the correct letter. Though this reduces performance to a scalar, it reliably tracks training progress and overall ability.

For DPO evaluation, we also use *lighteval*. The model is considered correct when the preferred answer has a higher log-probability than the rejected one, using the *zechen-nlp/MNLP-dpo-evals* dataset.

Throughout training, we monitored accuracy to track learning dynamics. For MCQA, we evaluated on *zechen-nlp/MNLP-STEM-mcqa-evals* and a custom benchmark combining datasets like *MMLU* and *AI2 ARC*. Categorizing questions by STEM subdomain helped us identify weak areas and adjust training data accordingly.

We also performed regular sanity-checks to ensure our model could still be used as an assistant and answered open questions correctly, which served us to detect catastrophic forgetting situations of some training parameters.

3.3 Baseline

To evaluate downstream performance, we compare our models against two baselines. The first is the original pretrained-only Qwen3-0.6B-Base model, which serves as our starting point. The second is the model obtained after our intermediate pretraining phase on the curated Wikipedia STEM subset and the SCP-116k dataset. This comparison allows us to quantify the impact of our additional pretraining and assess the effectiveness of the subsequent task-specific fine-tuning.

3.4 Experimental Details

3.4.1 DPO Experimental Details

The DPO training pipeline was structured in three stages to progressively select, validate, and scale the alignment strategy on STEM-focused preference data: All of the following stages were trained under identical hardware constraints Hyperparameters were chosen from this paper (Allal et al., 2025). Three hyperparameters were changed. Learning rate = 10^{-7}, warmup ratio = 0.1, max gradient norm = 1.0.

1. Method selection.

We empirically compared three recent preference optimization algorithms: Direct Preference Optimization (DPO) (Rafailov et al., 2024), Kahneman & Tversky Optimization (KTO) (Ethayarajh et al.), and Contrastive Preference Optimization (CPO) (Xu et al., 2024). Each method fine-tunes a policy model π_θ on a dataset $\mathcal{D} = \{(x_i, y_i^{\text{chosen}}, y_i^{\text{rejected}})\}_{i=1}^N$, where $y^{\text{chosen}} \succ y^{\text{rejected}}$. These methods are definied in the appendix A.1.

2. Reproduction and validation.

To validate the robustness of DPO on STEM data, we reproduced the alignment procedure described in (Allal et al., 2025) (Section 5.2 and 5.3), using the `metamathqa-50k` subset of the SmolTalk dataset and UltraFeedback (Cui et al., 2024) as the source of preference data. The base model used for this experiment was the pretrained reasoning model shown in Figure 1.

Two modifications were made to the original setup: (i) the max length was reduced to 1024 tokens, and (ii) training was restricted to the `metamathqa-50k` subset to focus on STEM-related tasks and to fit hardware and time constraints.

3. Large-scale DPO training.

Building on previous results, we scaled up DPO training using a curated dataset of over 300,000 STEM-focused preference pairs. The data was sourced from domain-specific QA datasets covering mathematics, programming, chemistry, medicine, physics, and general science. Each sample was converted into a (prompt, preferred, rejected) triplet format, as described in Section 3.1.2. Rejected responses were generated using `gemini-2.0-flash-lite-001`, selected for its efficiency and low cost.

3.4.2 MCQA Training

To align with our objective of answering factual STEM questions without requiring multi-step reasoning, we filtered out overly complex items—particularly those involving advanced math or physics from datasets like AQUA-RAT.

We explored various output formats. Predicting only the answer letter yielded limited improvement. Appending the full answer (e.g., "C. Photosynthesis") performed slightly better. The best results came from including explanations with the answer, likely due to richer supervision signals. However, placing the explanation before the answer reduced accuracy on our evaluation task, which expects single-token letter predictions. Detailed results are shown in Table 1.

Overfitting emerged after two epochs due to the modest dataset size. To address this, we applied data augmentation by rotating the correct answer among positions A–D and shuffling distractors. This effectively quadrupled the dataset and reduced overfitting by encouraging the model to learn the correct answer independently of fixed positional patterns.

Learning rate (LR) tuning was also critical. Values above 2×10^{-6} consistently caused performance degradation, likely due to catastrophic forgetting. In contrast, very low LRs (e.g., 1×10^{-7}) underperformed even with longer training. We observed a narrow optimal LR window necessary for stable and effective fine-tuning.

3.4.3 RAG Experiments

Since we self-generated documents related to specific questions (see Rag Data section), we are able to train the embedding model on our examples. To do this, we constructed triplets *(query, positive, negative)* where the query and positive are paired question and generated document, and the negative is a randomly sampled document from an unrelated STEM topic (e.g., medicine for a math query). We fine-tuned the `thenlper/gte-small` model (Li et al., 2023b) using triplet loss, and as can be seen in Figure 2, the encoder retrieval accuracy gains have a significant effect on the whole system's accuracy on MCQA questions.

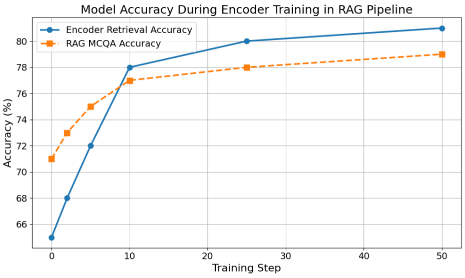

Figure 2: Results of RAG system and encoder accuracy as a function of encoder training epochs.

3.4.4 Quantization Experiments

We selected Post-Training Quantization (PTQ) over Pre-Training Quantization due to its implementation efficiency. Prior to quantization, we applied `SmoothQuant` (Xiao et al., 2024), a technique that migrates a proportion of activation outliers into weights via an offline mathematical transformation. The smoothed model was then quantized to 4-bit (`QLoRA` (Dettmers et al., 2023)) and 8-bit (`LLM.int8()` (Dettmers et al., 2022)) precision using the `BitsAndBytes` library (`transformers`).

3.5 Results

3.5.1 MCQA Models

The results summarized in Table 1 show a clear improvement over the baseline. Our best fine-tuned model surpasses the original Qwen3-0.6B-Base by approximately 5 percentage points, a notable gain considering the limited training time and resources, and the extensive pretraining already applied by the Qwen team. The Wikipedia-pretrained model also outperforms the base model, validating the relevance of our domain-specific pretraining strategy.

We further observe a significant performance boost with the RAG model. However, this improvement must be interpreted cautiously: part of the evaluation set had support documents explicitly generated for it. As noted earlier, model performance may degrade substantially on questions without relevant retrieved documents.

While our 4-bit Quantized model achieves greater efficiency at the cost of performance, our 8-bit variant maintains performance close to the "Multi-token + Explanation training" model with significantly reduced computational demands.

Model	Lighteval Accuracy (%)
Qwen3-0.6B	31.30
Qwen3-0.6B-Base	42.47
Wikipedia-pretrained	43.64
SCP116k-pretrained	40.98
Single-token training	44.29
Multi-token training	46.23
Explanation + Multi-token training	43.78
Multi-token + Explanation training	**47.14**
Rag model	**78.57**
4-bit Quantized model	37.53
8-bit Quantized model	45.84

Table 1: MCQA model accuracy across training setups. Accuracy on single-token prediction, no chat-template, MNLP-STEM-mcqa-eval dataset

3.5.2 Reward Model

Comparison of optimization methods. We fine-tuned `Qwen/Qwen3-0.6B-Base` using three preference optimization methods—DPO, KTO, and CPO—to identify the most effective variant for our setting. Due to time constraints, we limited the comparison to these three approaches. Among them, DPO consistently outperformed the others in terms of reward accuracy. While we observed this performance gap empirically, we do not currently have a theoretical explanation for DPO's advantage in this context.

Table 2: Lighteval accuracy for each alignment method

Method	Lighteval Accuracy (%)
KTO	43.29
CPO	42.71
DPO	**46.40**

Reproduction of alignment setup. The model fine-tuned with DPO on the `metamathqa-50k` subset and aligned using `UltraFeedback` reached a slightly lower accuracy than the base pretrained reasoning model (54.16% vs. 55.11%, see Table 3). While this result did not meet our initial expectation, it revealed key limitations—restricted context (1024 tokens) and narrow alignment data—that likely constrained performance. This outcome motivated us to scale alignment to a broader, more diverse dataset, which led to significant improvements (Table 4).

Table 3: Lighteval accuracy after alignment on SmolTalk subset

Model	Lighteval Accuracy (%)
Base Model	**55.11**
SmolTalk + UltraFeedback	54.16

Large-scale STEM preference model. Scaling DPO training to a large, diverse set of over 300,000 STEM-related preference pairs led to a substantial improvement in alignment performance. As shown in Table 4, the reward accuracy increased from 55.11% (base model) to 60.42% after alignment. This result confirms the benefit of both data volume and domain diversity when optimizing language models for preference consistency in STEM tasks. The reward margin is in the appendix A.4.

Table 4: Lighteval accuracy after large-scale DPO alignment

Model	Lighteval Accuracy (%)
Base Model	55.11
Aligned Model	**60.42**

4 Analysis

Effect of data quality.
On the metamathqa-50k subset, DPO with UltraFeedback resulted in a slightly lower scalar accuracy than the base model (54.16% vs. 55.11%). This suggests that narrow alignment datasets may require broader context or domain coverage to be fully effective. The same holds for the MCQA model, which requires quality knowledge-based questions.

Effect of data scale.
Scaling to 300 k heterogeneous STEM pairs yields a jump to 60.42% (Table 4), a larger benefit than switching optimization methods. Data volume and domain diversity therefore dominate further hyperparameter tuning at this model size. Even though we used data augmentation in MCQA training we noticed diminishing returns due to the small training dataset.

4.1 Reward Model

Method comparison.
DPO delivers a clear margin of $+0.032$ over KTO and $+0.037$ over CPO (Table 2). Direct policy updates appear to convert pairwise preferences into reward improvements more efficiently than temperature or constraint schemes in this 0.6B setting.

Failure modes.
A gap of roughly 0.40 reward accuracy remains. Likely causes include: (i) the limited capacity of the 0.6 B-parameter backbone; (ii) memory constraints that forced training with `max_length` ≤ 1024, so full reasoning chains were never presented; and (iii) domain gaps in the preference corpus (e.g., electrical engineering) that leave some STEM niches under-represented. Enlarging the model, extending the context window, and broadening corpus coverage should shrink this residue.

4.2 MCQA Models

Our MCQA models are inherently limited due to the impossibility to use chain-of-thought during evaluation, however, integrating a well-designed RAG system—with a trained document encoder and high-quality retrieval corpus—significantly boosted performance by offloading factual knowledge from the model to external sources. Additionally, quantization had minimal impact on accuracy while halving the model size, highlighting its potential for deploying LLMs on resource-constrained devices.

5 Ethical considerations

We fine-tuned Qwen3-0.6B-Base on English STEM MCQs with a design that could be easily extends to high-resource languages (French, German) using native tokenizers and corpora, and to low-resource languages (Urdu, Swahili) via cross-lingual transfer, synthetic data, and local partnerships. When deployed responsibly, the model delivers rapid, personalized feedback to learners. however, it also carries risks such as academic dishonesty, amplification of cultural or demographic biases inherited from Aquarat, SciQ, and similar sources, and potential exposure of copyrighted or sensitive material. To address these concerns, we could enforce strict access policies, conduct regular bias and collaborate with regional experts to curate inclusive question sets. Limited expertise in training led us to run numerous experiments—many of which were discarded, and we acknowledge that this approach demanded substantial GPU time and computational resources, resulting in a non-trivial environmental footprint. Future work will prioritize more efficient training strategies to minimize both resource consumption and ecological impact.

6 Conclusion

We showed the training process for different fine-tuned models. Our experiments demonstrate that targeted pretraining and high-quality preference data yield substantial gains in both answer accuracy and reward consistency, while RAG and data augmentation further enhance performance. We also show that careful quantization can retain most of these benefits with minimal resource overhead. Despite these advances, the model's capacity and fixed context window remain limiting factors, and environmental costs of large-scale training warrant more efficient methods. Future work will explore chain-of-thought supervision, broader domain coverage, and low-cost alignment strategies to further improve reasoning in small language models.

7 AI disclosure

During this project and in preparing this report, we used large language models, specifically, GPT-4o to assist with LaTeX formatting and to ensure the report's English was grammatically correct by reformulating certain paragraph. For the coding portion, GPT-4o-mini-high was used mainly to generate code comments and support debugging.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. 2025. Smollm2: When smol goes big – data-centric training of a small language model.

Alvaro Bartolome, Gabriel Martin, and Daniel Vila. 2023. Notus. https://github.com/argilla-io/notus.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. 2024. Huatuogpt-o1, towards medical complex reasoning with llms.

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:1803.05457v1*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Ultrafeedback: Boosting language models with scaled ai feedback.

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

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

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Matt Gardner Johannes Welbl, Nelson F. Liu. 2017. Crowdsourcing multiple choice science questions.

Isaac Johnson, Lucie-Aimée Kaffee, and Miriam Redi. 2024. Wikimedia data for ai: a review of wikimedia datasets for nlp tasks and ai-assisted editing.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. Camel: Communicative agents for "mind" exploration of large scale language model society.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning.

Will Lifferth, Walter Reade, and Addison Howard. 2023. Kaggle - llm science exam. https://kaggle.com/competitions/kaggle-llm-science-exam. Kaggle.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *ACL*.

Dakuan Lu, Xiaoyu Tan, Rui Xu, Tianchu Yao, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. 2025. Scp-116k: A high-quality problem-solution dataset and a generalized pipeline for automated extraction in the higher education science domain.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering.

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. In *EMNLP*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model.

Walter Reade Will Lifferth and Addison Howard. 2023. Kaggle - llm science exam.

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.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

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

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyuethayarajh2024ktomodelalignmentprospect Yue, Wanli Ouyang, Dongzhan Zhou, Shufei Zhang, Mao Su, Han-Sen Zhong, and Yuqiang Li. 2024. Chemllm: A chemical large language model.

A Appendix

A.1 Preference optimization agorithms

- **DPO** optimizes a KL-regularized objective without requiring an explicit reward model:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta;\pi_{\text{ref}}) = -E_{(x,y^+,y^-)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y^+|x)/\pi_{\text{ref}}(y^+|x)}{\pi_\theta(y^-|x)/\pi_{\text{ref}}(y^-|x)}\right)\right]$$

  where π_{ref} is a fixed reference model and $\beta > 0$ controls the strength of alignment.

- **KTO** generalizes DPO by learning an instance-specific temperature $\tau(x)$ instead of using a global β:

$$\mathcal{L}_{\text{KTO}} = -E_{(x,y^+,y^-)}\left[\log\sigma\left(\frac{1}{\tau(x)}\left(\log\pi_\theta(y^+|x) - \log\pi_\theta(y^-|x)\right)\right)\right]$$

  where $\tau(x)$ is predicted by an auxiliary neural network during training.

- **CPO** frames the objective as a contrastive learning problem, where the model learns to prefer correct completions over negatives drawn from a rejection pool:

$$\mathcal{L}_{\text{CPO}} = -\log\frac{\exp(\pi_\theta(y^+|x)/\tau)}{\exp(\pi_\theta(y^+|x)/\tau) + \sum_{y^-\in\mathcal{N}(x)}\exp(\pi_\theta(y^-|x)/\tau)}$$

  where $\mathcal{N}(x)$ is a set of contrastive negatives and τ is a temperature hyperparameter.

A.2 Prompt format for MCQA

Each multiple-choice example is formatted exactly as follows to induce reasoning through the rationale

```
Question: <question_text>
Options:
A. <option_1>
B. <option_2>
C. <option_3>
D. <option_4>
Answer and Rationale:
```

A.3 Large-scale DPO training dataset

We summarize below all datasets used throughout our experiments, grouped by experiment type and including citations where applicable.

- `argilla/ultrafeedback-binarized-preferences-cleaned` (Bartolome et al., 2023)
- `AI4Chem/ChemData700K` (Zhang et al., 2024)
- `meta-math/MetaMathQA` (Yu et al., 2023)
- `FreedomIntelligence/medical-o1-reasoning-SFT` (Chen et al., 2024)
- `ayoubkirouane/arxiv-physics`
- `camel-ai/physics` (Li et al., 2023a)
- `Sangeetha/Kaggle-LLM-Science-Exam` (Lifferth et al., 2023)
- `mlabonne/orpo-dpo-mix-40k`
- Preference pair generated during Milestone 1

A.4 Large-scale STEM preference model reward margin

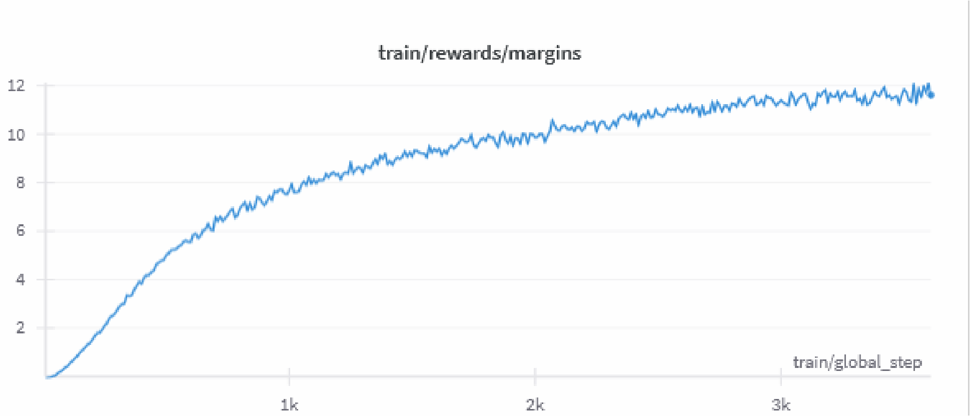

Figure 3: Reward margin