WEBSCALE-RL: AUTOMATED DATA PIPELINE FOR SCALING RL DATA TO PRETRAINING LEVELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have achieved remarkable success through imitation learning on vast text corpora, but this paradigm creates a training-generation gap and limits robust reasoning. Reinforcement learning (RL) offers a more data-efficient solution capable of bridging this gap, yet its application has been constrained by a critical data bottleneck: existing RL datasets are orders of magnitude smaller and less diverse than web-scale pre-training corpora. To address this, we introduce the **`Webscale-RL` pipeline**, a scalable data engine that systematically converts large-scale pre-training documents into millions of diverse, verifiable question-answer pairs for RL. Using this pipeline, we construct the **`Webscale-RL` dataset**, containing 1.2 million examples across more than 9 domains. Our experiments show that the model trained on this dataset significantly outperforms continual pretraining and strong data refinement baselines across a suite of benchmarks. Notably, RL training with our dataset proves substantially more efficient, achieving the performance of continual pre-training with up to 100× fewer tokens. Our work presents a viable path toward scaling RL to pre-training levels, enabling more capable and efficient language models.

1 INTRODUCTION

Large Language Models (LLMs) have achieved remarkable success, primarily through learning on vast text corpora. However, this predominant paradigm, which includes pretraining with next-token prediction and supervised fine-tuning (SFT), is fundamentally in the form of imitation learning. By training models to mimic static offline datasets, imitation learning creates a "teacher-forcing" dependency that makes models vulnerable to distribution shifts (Ross et al., 2011; Levine et al., 2020; Foster et al., 2024) and leads to a significant gap between training and generation dynamics (Chen et al., 2024c; Bachmann & Nagarajan, 2024; Cen et al., 2024). Consequently, models trained in this way struggle with distribution shift and lack the robust reasoning abilities required for complex problem solving.

Reinforcement learning (RL) offers a powerful alternative to overcome these challenges (Shao et al., 2024; DeepSeek-AI et al., 2025). By learning from online reward feedback on its own generations, an RL-trained model can explore a wider solution space and is not confined to a static dataset, bridging the training-inference gap. This online learning process makes RL a significantly more data-efficient training paradigm. As our empirical results demonstrate, RL can achieve performance gains comparable to continual pretraining with up to two orders of magnitude fewer tokens, providing a compelling motivation for scaling RL to unlock new levels of model capability and efficiency.

Despite the clear advantages of RL, its adoption at scale is severely hampered by a critical data bottleneck and most existing practice in RL training is mainly limited to reasoning tasks such as math and coding in the post-training stage. Pretraining corpora are measured in trillions of tokens, whereas existing RL datasets are orders of magnitude smaller (e.g., $<$10B tokens for RL vs. $>$1T tokens for pretraining) and lack the diversity of web-scale data. This scarcity is driven by the high cost of generating high-quality, verifiable question-answering (QA) pairs, which are essential for effective RL-based reasoning tasks. This immense disparity in data scale and diversity prevents RL from realizing its full potential to enhance the general reasoning capabilities of LLMs.

To address these limitations, we introduce **`Webscale-RL`**, a scalable data pipeline that systematically converts large-scale pretraining corpora into massive, diverse, and verifiable RL-ready datasets.

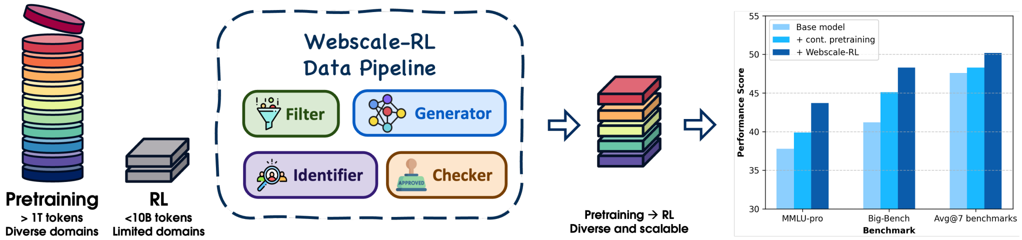

Figure 1: The scaling on LLM RL is fundamentally bottlenecked by the scarcity of high-quality RL data. While pretraining leverages >1T diverse web tokens, RL datasets remain limited to <10B tokens with limited diversity. We propose `Webscale-RL` data pipeline to fundamentally improve the scalability of RL data: we convert the pretraining corpora to verifiable query and ground-truth answer pairs, scaling RL data to pretraining levels while preserving the diversity. The experiments show that RL with `Webscale-RL` data is significantly more effective and efficient than continual pretraining and data refinement baselines.

Our pipeline is designed to bridge the data gap between pretraining and reinforcement learning, unlocking the potential to train LLMs with RL at a scale previously unattainable while preserving the vast diversity of the original pretraining data.

Our main contributions are threefold:

- We propose **`Webscale-RL` pipeline**, an automated and scalable data engine that converts web-scale pretraining documents into verifiable question-answer pairs for RL. The pipeline incorporates stages for data filtering, domain and persona-driven generation, and quality verification to ensure high-quality output.
- We construct **`Webscale-RL` dataset**, a large-scale and diverse RL dataset containing 1.2 million verifiable QA pairs spanning over nine domains. Our analysis shows it is significantly more diverse than existing large-scale RL datasets.
- We provide empirical evidence demonstrating the effectiveness of our approach. The model trained with RL on the `Webscale-RL` dataset significantly outperforms continual pretraining on the source data and strong data refinement baselines across a wide range of benchmarks. Furthermore, our results show that RL with our dataset is substantially more data-efficient, achieving comparable performance to continual pretraining with 100× fewer tokens.

Our work demonstrates that by converting massive pretraining corpora into a format suitable for RL, we can unlock significant performance and efficiency gains. This provides a path toward scaling reinforcement learning to match the scale of pretraining, leading to a new generation of more capable and robust language models.

2 Related Works

Training Data Development and Synthesis. The development of LLMs hinges on the availability of vast, high-quality training datasets. However, curating such corpora, especially labeled and across diverse domains, is often prohibitively expensive and time-consuming. This challenge has spurred significant research into efficient synthetic data generation pipelines (Chen et al., 2024a; Ma et al., 2025; Fan et al., 2025). Current pre-training corpora are typically compiled from a variety of public sources, such as Wikipedia (Foundation), large-scale web crawls (Computer, 2023; Paster et al., 2023; Penedo et al., 2024a; Li et al., 2024; Raffel et al., 2019), and code repositories (Lozhkov et al., 2024). This is often supplemented with content from digitized books (Stroube, 2003) and data generated by other LLMs (Ben Allal et al., 2024; Huang et al., 2024). To endow LLMs with comprehensive knowledge and robust downstream capabilities, these corpora are constructed on a massive scale, often containing tens of trillions of tokens, as exemplified by the 30-trillion-token RedPajama dataset (Computer, 2023) and the 67.5-trillion-token Stack-v2 (Lozhkov et al., 2024). More recently, the success of models like DeepSeek-R1-Zero (DeepSeek-AI et al., 2025) and DeepSeek-R1-Zero (DeepSeek-AI et al., 2025) and Grok-4 (xAI, 2025), which integrate reinforcement learning

(RL) at the pre-training stage, is built on and is built on and has intensified the demand for similarly large and high-quality synthetic synthetic RL datasets. There have been multiple lines of work in large-scale data synthesis for LLM training: DeepScaler (Luo et al., 2025) curated a small (40K) RL dataset for mathematical reasoning; OpenR1-Math (Hugging Face, 2025) further scales up the mathematical RL dataset for both SFT and RL via distillation, resulting in a 220K dataset; WebInstruct (Yue et al., 2024), OpenThoughts (Guha et al., 2025) and NatureReasoning (Yuan et al., 2025) expand the distillation path to multiple domains and synthesize over 1M data using teacher models for SFT, respectively; Nemotron (Bercovich et al., 2025) extends the dataset size to 3.9M for both SFT and RL.

Reinforcement Learning in LLMs. Recent advancements in large-scale RL have significantly enhanced the capabilities of LLMs, as demonstrated by models such as OpenAI's o-series (OpenAI, 2024a; Jaech et al., 2024; OpenAI, 2024b) and DeepSeek-V3/R1 (DeepSeek-AI et al., 2024; 2025). Besides these models, many other works show that LLMs trained to reason with Chain-of-Thought (CoT) prompting have shown substantial performance gains in diverse areas, including mathematical and scientific reasoning (Xie et al., 2025; Cen et al., 2025; Shao et al., 2024; Luong et al., 2024; Chen et al., 2024b; Cui et al., 2025), code generation (Le et al., 2022; Wei et al., 2025), and tool use (Zhang et al., 2025a; Qian et al., 2025). The optimization of the RL objective in these models is primarily driven by foundational algorithms like Proximal Policy Optimization (PPO) (Schulman et al., 2017) and its variant, Group Relative Policy Optimization (GRPO) (Shao et al., 2024). Rooted from post-training RL, many works further extend RL to a significantly larger scale (Liu et al., 2025c;b; xAI, 2025) or an earlier stage like pre-training (Zelikman et al., 2024; Dong et al., 2025; Li et al., 2025), indicating the effectiveness of prolonged, large-scale RL training.

3 METHODOLOGY

In this section, we first provide a brief comparison of the pretraining and RL training paradigms and then present our **`Webscale-RL`** data pipeline that systematically converts large-scale pretraining data into RL data while preserving the scale and diversity of web data.

3.1 PRELIMINARIES

Pretraining. In the pretraining stage, a large-scale corpus $\mathcal{D}_{\text{pretraining}}$ (usually $>$1T tokens) is constructed by filtering and deduplicating publicly available web data sources (Penedo et al., 2024b; Weber et al., 2024; Li et al., 2024). Given this static dataset, the LLM is trained in a teacher-forcing manner to imitate the next-token distribution of the data by minimizing the negative log-likelihood:

$$\min_{\theta} -\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{pretraining}}}\left[\sum_{t=1}^{T}\log P_\theta(x_t \mid \mathbf{x}_{(<t)})\right], \tag{1}$$

where $\mathbf{x} = [x_1, \dots, x_T]$ is a token sequence sampled from the pretraining dataset $\mathcal{D}_{\text{pretraining}}$. This imitation-based objective enforces the model to learn the given pattern from the demonstration data but does not expose the model to the distribution induced by its own generations, suffering from the distribution shift issue (Bachmann & Nagarajan, 2024; Ross et al., 2011) and leading to a training-inference gap (Bengio et al., 2015; Levine et al., 2020).

Reinforcement Learning (RL). RL instead optimizes the model as a policy that *generates answers online* and maximizes expected reward on a query $\mathbf{q}$:

$$\max_{\theta} \mathbb{E}_{\mathbf{q}\sim Q,\mathbf{a}\sim P_\theta(\cdot|\mathbf{q})}\left[R(\mathbf{q},\mathbf{a})\right], \tag{2}$$

where Q is the query set and R is a task-specific reward function. The online generation and feedback loop enable the model to narrow the training-inference gap. In our setup, we adopt a binary reward that returns 1 only when the model's final answer matches the ground-truth answer and 0 otherwise. Consequently, each RL training instance is a verifiable question-answer pair.

3.2 WEBSCALE-RL DATA PIPELINE

While RL has shown promise in enhancing LLM capabilities (Jaech et al., 2024; DeepSeek-AI et al., 2025), its effectiveness is constrained by the limited scale and diversity of existing RL datasets.

Therefore, the RL training is typically conducted on a much smaller scale on limited domains in the post-training stage. This discrepancy arises from the high costs associated with human annotation and the challenges in generating verifiable QA pairs at large scale. Furthermore, most existing RL datasets focus on specific tasks or domains and thus lack the breadth of topics and styles found in web-scale corpora, limiting their generalization to diverse real-world scenarios. To address the limited volume and diversity of existing RL datasets, we propose a **`Webscale-RL`** data pipeline that converts pretraining documents into RL data at scale while preserving the diversity of web data.

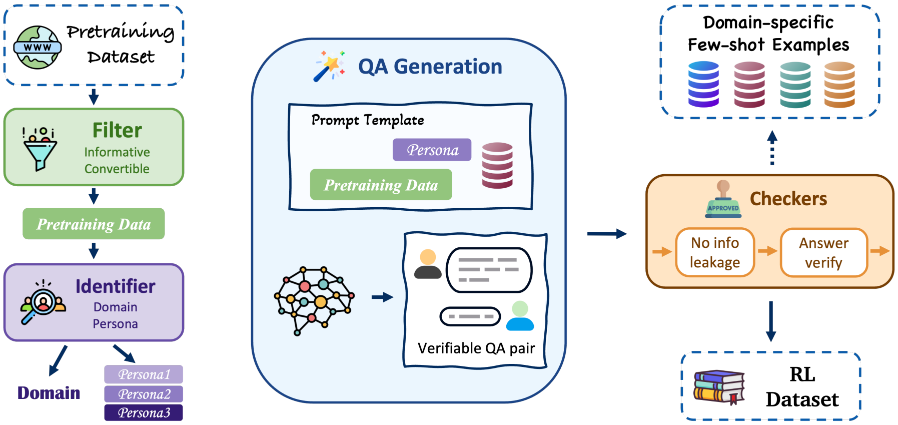

Figure 2: Overview of the **`Webscale-RL`** data pipeline that systematically converts large-scale pretraining data into RL data while preserving the scale and diversity of web data. The pipeline maintains a domain-specific demonstration library for few-shot examples for high quality generation and assigns multiple personas to each document to encourage reflecting different viewpoints. The generated QA pairs are verified for correctness and leakage prevention to ensure the reliability of the RL dataset. The prompt templates of four stage are listed in Appendix B.1.1.

At a high level, `Webscale-RL` leverages a generative model to convert narrative pretraining documents into *verifiable* QA pairs for RL training. To cover a wider range of topics and question styles, we first maintain a domain-specific demonstration library for few-shot examples to guide the generation process. We further assign multiple *personas* to each document to encourage reflecting different viewpoints. Figure 2 illustrates the pipeline, which consists of four main stages:

Data Filtering. Our pipeline takes pretraining corpora spanning multiple domains as input instead of focusing on data in limited domains Toshniwal et al. (2024); Liu et al. (2025a). This stage aims to remove inputs that are unlikely to yield verifiable high-quality questions. We first use heuristics to filter out obviously low-quality documents ($<$ 50 tokens) and then employ an LLM for further fine-grained filtering (Gunasekar et al., 2023; Wettig et al., 2024). Different from previous pipelines that strictly filter data from multiple dimensions (e.g., difficulty (Fan et al., 2025; Ma et al., 2025), format (Zhou et al., 2025), with sophisticated reasoning traces (Yuan et al., 2025), etc.), our filter aims to select data for the following stages while maximally preserving the diversity of the original materials. Therefore, the LLM-based filter only identifies and removes (i) non-informative pages where most contents are boilerplate (e.g. navigation, headers, or footers in website html), and (ii) non-self-contained fragments that lack sufficient context to verify answers. This two-stage filtering ensures that the retained documents are both *informative* and *convertible* into verifiable RL data.

Domain Classification and Persona Assignment. After filtering, we then classify each document into a specific domain (e.g., commerce, healthcare, social science, etc.) using a LLM-based classifier. Due to extreme high diversity of the pretraining data, our pipeline adopts different few-shot examples for each domain to ensure that the generated questions are contextually appropriate and verifiable, which is absent in existing pipelines (Fan et al., 2025; Yuan et al., 2025). The domain tags are then used to collect relevant few-shot exemplars in the subsequent QA generation step. Additionally, to further enhance the diversity of the generated QA pairs, we assign multiple *personas* who will be interested in the content to each document (Ge et al., 2024), which defines the style and perspective from which questions will be generated. For example, a document classified under the "healthcare" domain might be assigned personas such as "medical expert," "patient," or "health journalist." This persona assignment encourages reflecting different viewpoints and information needs in question generation given the same document, thereby capturing more information in the source data and enriching the RL dataset's diversity.

Verifiable QA Generation. Conditioned on the source document, domain tag, and chosen persona, the LLM-based QA generator produces verifiable question-answer pairs. Specifically, we first sample few-shot examples from the domain-specific demonstration library, a curated pool covering a range of question types and complexities within each domain to ensure that the generated questions are of high quality. We then incorporate all contexts with a prompt template (in Appendix B.1.1) to guide the LLM-based generator to extract diverse question-answer pairs from the perspective of the assigned persona. For question generation, beyond extracting the questions originally contained in the document (Yue et al., 2024), our generator can also raise new questions that are answerable according to the pretraining data. Since the trained model is not allowed to access the source document during RL, we further instruct the generator to provide necessary contexts to ensure that the question is self-contained. Meanwhile, we only require a relatively short and verifiable ground-truth answer (e.g., a number, a name, or a phrase) grounded by the pretraining materials instead of a long explanation or detailed reasoning steps composed by a strong LLM (Yue et al., 2024; Yuan et al., 2025), which significantly reduces the generation complexity and reliance on the backend LLMs. In other words, our generation is to extract the answer from the document instead of distilling from a powerful LLM. This design choice allows us to leverage more cost-effective LLMs for generation while still producing high-quality, verifiable QA pairs suitable for RL training. We provide a conversion example in Appendix B.2.1.

Quality Check and Leakage Control. While the most generated question-answer pairs are of high quality, some may still contain errors or hallucinations. To ensure the reliability of the RL dataset, we leverage an LLM-based verifier to implement a multi-stage checking process (Liu et al., 2024; Prabhakar et al., 2025): 1) *Correctness verification*. Unlike accuracy-based post-processing in previous works (Zhou et al., 2025; Ma et al., 2025), our verification assesses the correctness of the answers by checking if the extracted QA data are grounded by the source document, which is much less biased by the backend LLMs and effectively reduces the wrong reward signals during RL training; 2) *Leakage prevention* ensures that the questions do not reveal answers explicitly (e.g., the ground truth is not trivially embedded in the prompt). The verifier filters out any QA pairs that fail to meet these criteria, ensuring that the final dataset truly tests the model's knowledge or reasoning capabilities rather than its ability to summarize or retrieve information directly from the prompt.

The prompt templates of four stages are listed in Appendix B.1.1. The examples of pretraining to RL conversion are in B.2. We further apply data decontamination by lm-eval-harness (Gao et al., 2024) to remove overlaps with the evaluation. With this pipeline, we can systematically convert large-scale pretraining data into a massive, diverse, and verifiable RL-ready dataset that closely matches the scale and diversity of the original pretraining corpus. This approach effectively addresses the RL data scarcity issue and enables scaling up RL training of LLMs across a wide range of tasks and domains. More discussions are described in Appendix B.1.

4 WEBSCALE-RL DATASET

4.1 DATASET CONSTRUCTION

We construct **`Webscale-RL`** dataset by running the data pipeline over a subset ($\sim$1M data in total) of the mixture of pretraining corpora including DCLM (Li et al., 2024), Wikipedia (Foundation), MegaMath (Zhou et al., 2025), Stack-v2 (Lozhkov et al., 2024), etc. The choice of pretraining data here aims to cover diverse domains and mimics previous practice on pretraining (Bakouch et al., 2025). The data selection is flexible and can be adjusted based on the target model and application.

In RL data conversion, we use GPT-4.1-mini for domain classification and final quality check, and GPT-4.1 for data filtering and QA generation. As we mentioned in QA generation stage in Sec. 3.2, our pipeline aims to extract answer grounded by the pretraining document instead of distilling from a strong LLM, which reduces the bias and reliance on the backend LLMs. Therefore, the pipeline can be extended to other open-source LLMs such as GPT-OSS (Agarwal et al., 2025) and Deepseek series (DeepSeek-AI et al., 2025). For each qualified document, we assign up to 3 personas to generate diverse QA pairs. The final dataset contains $\sim$1.2M QA pairs covering 9+ domains. Note that the dataset can easily be further scaled up to the pretraining level with our **Webscale-RL** pipeline. More details of dataset construction are described in Appendix B.2.

4.2 Dataset Analysis

We compare our Webscale-RL dataset with other widely used pretraining datasets (RedPajama-v2 (Weber et al., 2024), FineWeb-Edu (Penedo et al., 2024b), DCLM-baseline (Li et al., 2024)), SFT datasets (NaturalReasoning (Yuan et al., 2025), Nemotron (Bercovich et al., 2025)) which include reasoning CoT in the answers, and RL datasets (DeepScaler (Luo et al., 2025), OpenR1-Math (Hugging Face, 2025), OpenThoughts3 (Guha et al., 2025)) which include a ground-truth answer for each question. The detailed comparison is listed in Table 1.

Table 1: The comparison of various datasets with our Webscale-RL dataset. The number of data indicates the number of documents (for pretraining datasets) or the number of QA pairs (for SFT and RL datasets). The **scalability** indicates the potential of scaling up the dataset size: DeepScaler has low scalability because it is collected from competitions and relies on human annotation. Other post-training datasets generate answers by distillation but they collect queries from limited sources, which limits the further scaling. In contrast, both the questions and answers in the Webscale-RL dataset are converted from and grounded by the pretraining datasets, which can be easily scaled up to pretraining level.

Dataset	Type	# of data	Domain	Data Source	Scalability
RedPajama-v2	Pretrain	>100B	Multi-domain	Web crawling	/
FineWeb-Edu	Pretrain	>3B	Multi-domain	Web crawling	/
DCLM-baseline	Pretrain	>3B	Multi-domain	Web crawling	/
DeepScaler	RL	40K	Math	Competition and other math datasets	Low
OpenR1-Math	SFT/RL	220K	Math	Distilled from DeepSeek-R1	Medium
OpenThoughts3	SFT	1.2M	Math, Code, Science	Distilled from QwQ-32B	Medium
NaturalReasoning	SFT	1.1M	Multi-domain	Converted from pretrain + distillation	High
Nemotron	SFT/RL	3.9M	Math, Code, Science	Distilled from multiple models	Medium
Webscale-RL	RL	1.2M	Multi-domain	Converted from pretrain	High

The comparison shows that the pretraining corpora are orders of magnitude larger and span broad domains, whereas existing SFT/RL datasets are significantly smaller and often focus on a few areas (notably math and code), which limits coverage of general knowledge and open-ended reasoning found in web-scale text. The Nemotron dataset includes data in other domains such as general QA and safety, which however only constitutes a small portion of the dataset. It is also worth noting that while some datasets have a relatively large data volume (e.g., OpenThoughts3, Nemotron), they still encounter the challenge of further scaling due to their limited sources of queries. In contrast, our Webscale-RL dataset is constructed by converting from the pretraining documents, allowing for easy expansion to pretraining scale.

We also obverse that a large fraction of the SFT/RL data is distilled from other teacher models. This couples dataset quality and ceiling to teacher capability and availability. In contrast, Webscale-RL is *grounded in source documents*: the generator does not need to solve the problems during construction; instead, we extract verifiable QA pairs from existing texts, reducing the dependence on strong teachers. Furthermore, because both questions and answers are derived from pretraining documents and verified against the source, Webscale-RL can scale naturally with the size of available corpora (i.e., the pretraining scale) while maintaining diversity, unlike human-labeled or fully distilled datasets whose growth is bottlenecked by annotation or query generation.

We list the domain distribution of our dataset in Fig. 3 left. Webscale-RL spans 9+ domains inherited from pretraining sources, substantially more diverse than most public post-traininig datasets. While we observe that the STEM-related domains (Math, Science, Code) constitute a significant portion of the dataset, it is also worth noting that the underrepresented domains in existing RL datasets, such as Lifestyle ($> 8.6\%$), Commerce ($> 3.3\%$), etc., are well covered in Webscale-RL, which are essential for general-purpose assistants.

To further illustrate the diversity of Webscale-RL dataset, we compare it with Nemotron, a large-scale SFT/RL dataset mainly covering math, code, and science. Since we focus on question diversity, we first randomly sample 5K questions from each dataset and encode them using the Qwen3

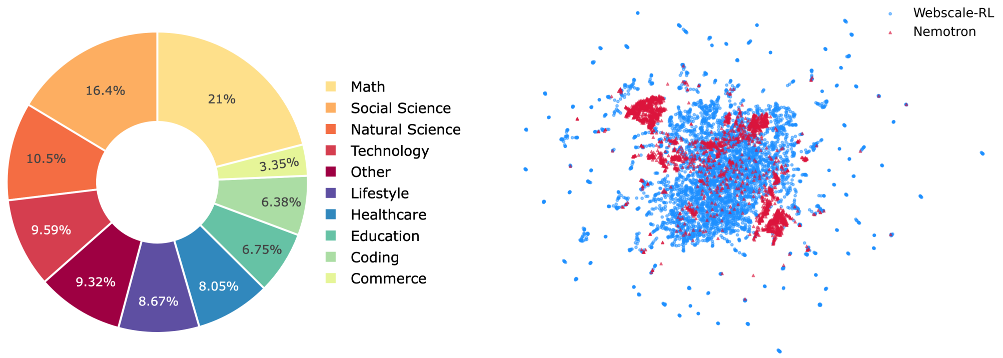

Figure 3: Left: The domain distribution of `Webscale-RL` dataset. Right: The comparison on question embedding of `Webscale-RL` and Nemotron data. We randomly sample 5K questions from each dataset and visualize the embedding (by Qwen3-Embedding) reduced to 2D using UMAP.

Embedding model (Zhang et al., 2025b). We then reduce the embedding dimension to 2 using UMAP (McInnes et al., 2018) for visualization. The results are shown in Fig. 3 right. Although both datasets cover multiple domains, Nemotron data points are mainly clustered in several regions, indicating a focus on specific topics. In contrast, the `Webscale-RL` data points are converted from a larger variety of documents and are generated from diverse perspectives by different personas, resulting in a distribution that is more uniform and more scattered, indicating a broader coverage of topics and knowledge areas. The diversity along with the large scale of Webscale-RL can help models learn a wide range of knowledge and reasoning skills, enhancing their versatility and performance across various tasks.

5 Experiments

In this section, we conduct experiments to evaluate the effectiveness of the `Webscale-RL` dataset generated by our proposed pipeline. Our experiments aim to address two main questions: (1) Can RL data generated by our pipeline enhance model performance across various benchmarks? (2) Does RL training scale more effectively and efficiently than standard teacher-forcing training?

5.1 Experiment Setup

Baselines. To answer these questions, we finetune a Qwen2.5-3B model (Yang et al., 2024a) using GRPO (Shao et al., 2024) on the `Webscale-RL` dataset and compare it with continual pretraining on the corresponding base dataset, i.e., the original pretraining data prior to RL conversion. We further compare our method with several advanced data refinement techniques: (1) QuRating(Wettig et al., 2024), which selects high-quality data via LLM ranking and filtering; (2) ProX(Zhou et al., 2024), which uses programmatic cleaning to enhance data quality; and (3) Generative Data Refinement (GDR) (Jiang et al., 2025), which originally uses LLM to improve the safety of the corpus (e.g., remove personally identifiable information, toxic content). In our experiment, we use it to improve the quality of the pretraining dataset. For these baselines, we refine the pretraining data using each method and perform continual pretraining on the resulting datasets.

Notably, we observe that RL training substantially improves the model's instruction-following abilities, while the continual pretrained models may fail to start answering in the evaluation, especially for questions with zero-shot examples, potentially introducing an evaluation bias. To mitigate this and enable a fair comparison, we construct an SFT dataset comprising 10K high-quality examples. Specifically, we first generate QA pairs via our `Webscale-RL` pipeline and then use GPT-4.1 to distill a relatively short reasoning CoT for each question given the ground-truth answer.

Training. For the continual pretraining and data refining baselines, we start from the base model and continue to pretrain on a 1M corpus, which represents a superset of the source data for the `Webscale-RL` dataset. We then follow with SFT training with a smaller learning rate using the 10K high-quality examples. For RL training, we first apply SFT with the same SFT dataset for warm-up. We then sample 150K data points from the `Webscale-RL` dataset and run standard GRPO training. More details of the SFT dataset and training are described in Appendix B.3.

Benchmarks. We evaluate the models on a diverse set of benchmarks to assess their general capabilities and domain-specific performance, including general tasks (MMLU-pro (Wang et al., 2024), Big-Bench (Srivastava et al., 2023)), math & STEM tasks (GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), GPQA-diamond (Rein et al., 2024)) and coding tasks (MBPP (Austin et al., 2021) and EvalPlus (Liu et al., 2023)). For EvalPlus, we report the average score of HumanEval (Chen et al., 2021), MBPP, HumanEval+ and MBPP+. In evaluation, we use the same pipeline and configurations for all models. Specifically, we use zero-shot evaluation for Big-Bench, GPQA-diamond and MATH500. We use 5-shot for MMLU-pro and 8-shot for GSM8K evaluation following the default setting in lm-eval-harness (Gao et al., 2024). More details in evaluation are described in Appendix B.3.

5.2 Main Results

Table 2 summarizes the comparisons of Webscale-RL with other baselines. Our method outperforms all baselines across most benchmarks, including continual pretraining and advanced data refinement pipelines. We observe an average improvement of 3.4 over the strongest baseline (GDR). Notably, Webscale-RL even narrows the performance gap to the much larger Qwen2.5-7B model from 10.6 pts to 6.1 pts on average. This indicates that converting web-scale corpora into verifiable QA and optimizing with RL yields stronger downstream gains than further imitation on even refined text.

Particularly, the improvements are most pronounced on general knowledge and reasoning tasks (MMLU-pro, Big-Bench, GPQA-diamond), which significantly benefit from the diversity and breadth of the `Webscale-RL` dataset inherited from pretraining sources. On math tasks, we observe a large jump on MATH500 from 47.6 to 58.0 after RL training with Webscale-RL, which is close to the 7B model. This aligns with prior findings that RL can better incentivize math reasoning (Shao et al., 2024; Yang et al., 2024b) compared to simply imitating refined documents or QA demonstrations. The gain on GSM8K is relatively smaller, likely due to the saturation effect as the base model already achieves strong performance. Meanwhile, the performance improvement on coding tasks is relatively smaller, likely reflecting the lower proportion of coding data in the pretraining corpus. Notably, the 3B model finetuned with Webscale-RL substantially narrows the performance gap to the 7B base model on the macro average, suggesting a practical path to stronger small models via RL scaling.

Table 2: Comparison results of our Webscale-RL with baselines on various benchmarks. To mitigate evaluation bias, continual pretraining and data refinement baselines are followed by SFT training to enhance instruction following. While all finetuning are based on the Qwen2.5-3B model, we also compare with the 7B base model. **Blue bold** indicates the best result among 3B baselines; **green bold** shows where we match or exceed the 7B model.

Method	**MMLU-pro**	**BigBench**	**GPQA-D**	**MATH500**	**GSM8K**	**MBPP**	**EvalPlus**	**Avg**
Qwen2.5-3B	37.8	41.2	20.8	47.6	74.2	54.6	57.3	47.6
Qwen2.5-7B	48.3	58.7	29.6	60.8	84.4	63.4	62.2	58.2
Cont. Pretrain	39.9	45.1	18.6	44.0	77.4	**55.2**	**57.8**	48.3
QuRating	39.7	44.9	19.4	44.6	76.8	54.8	57.6	48.3
ProX	40.0	46.0	19.5	44.4	77.3	54.2	57.5	48.4
GDR	39.9	46.0	20.8	44.4	77.4	55.0	57.6	48.7
Webscale-RL	**43.7**	**48.3**	**23.2**	**58.0**	**78.5**	55.0	**57.8**	**52.1**

Despite using a small SFT set to reduce evaluation bias toward instruction-following, RL still maintains clear advantages over SFT-augmented continual pretraining baselines. This suggests that the gains from Webscale-RL are not solely due to improved instruction adherence but stem from the reward-driven online learning signal. Overall, these results demonstrate that our `Webscale-RL` data pipeline effectively scales up RL data by converting from pretraining corpus and enables significant capability improvements across diverse tasks and domains.

5.3 Performance Comparison of Scaling Training

While RL shows remarkable advantages over teacher-forcing training in terms of final performance, we further investigate the scaling efficiency of RL training compared to standard pretraining with respect to the amount of training tokens. To this end, we compare the performance of RL training and continual pretraining at different training scales by varying the amount of data sampled from the `Webscale-RL` dataset and the original pretraining corpus, respectively. Notably, we observe that the length of QA pairs in the `Webscale-RL` dataset differs from the length of document in the original pretraining corpus while their source data are the same. Therefore, for a fair comparison on *token efficiency of the original data*, we compute the token number of RL training by the original pretraining corpus used to generate the `Webscale-RL` dataset instead of the `Webscale-RL` dataset itself. For example, if we generate two 300-token QA pairs from a 4000-token pretraining text, then we count the RL training token number as 4000 instead of 600 when training on these two QA pairs. Note that for continual pretraining with different training data volume, we also apply the same SFT training as a follow-up using the 10K high-quality examples to mitigate the evaluation bias.

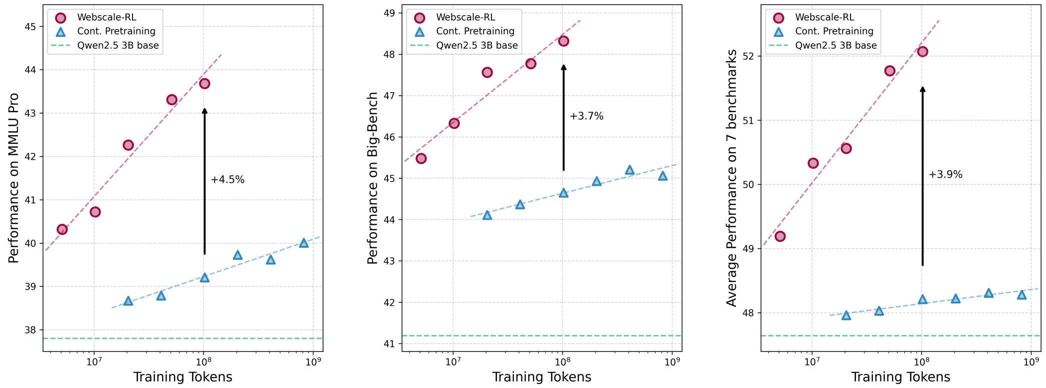

Figure 4: Scaling comparison between Webscale-RL training and continual pretraining with the original pretraining corpora. We report the performances on MMLU-pro (left), Big-Bench (middle) and average on all benchmarks (right). The token number for RL training is calculated based on the original pretraining corpus used to generate the `Webscale-RL` dataset. The each data point in continual pretraining baselines are followed by a SFT training using the same 10K high-quality examples. The RL training on `Webscale-RL` consistently outperforms continual pretraining at different training scales and exhibits better scaling efficiency.

Since the pretraining corpus mainly consists of general web text, we focus on evaluating the models on general tasks (MMLU-pro and Big-Bench) to better reflect the impact of training scale. We also report the average performance across all benchmarks to provide a holistic view.

Figure 4 illustrates the performance comparison between RL training with `Webscale-RL` dataset and continual pretraining with pretraining corpora at different training scales. We observe that RL training consistently outperforms continual pretraining across all three metrics (MMLU-pro, Big-Bench, and average performance) at various training scales. With the same amount of training tokens (100 millions), RL training achieves 4.4% improvement over Qwen2.5-3B base model in average while continual pretraining exhibits similar performance to the base model.

Notably, RL training achieves comparable or better performance with significantly fewer training tokens. For instance, on MMLU-pro, RL training with approximately 10M tokens attains similar performance to continual pretraining with 1B tokens, indicating over 100× improvement in data efficiency. Furthermore, RL training exhibits a steeper upward trend as the training scale increases, which is also true for other benchmarks, demonstrating that RL training not only leads to higher final performance, but scales more effectively and efficiently than standard teacher-forcing approaches.

6 CONCLUSION

In this paper, we introduced the Webscale-RL pipeline, an end-to-end data engine that converts web-scale pretraining corpora into verifiable, RL-ready data while preserving diversity. With this pipeline, we constructed the Webscale-RL dataset, which is orders of magnitude larger and more diverse than existing RL datasets. Empirically, training a LLM with RL on Webscale-RL improves performance across a diverse suite of benchmarks and delivers better data efficiency than continual pretraining at comparable token budgets, especially on general knowledge and open-ended reasoning (MMLU-pro, Big-Bench), with consistent improvements in math and STEM areas.

While our results demonstrate the promise of scaling RL data to pretraining levels, several limitations and future directions remain. The current Webscale-RL dataset lacks coverage of high-quality data in certain domains such as coding, which leads to smaller gains on coding benchmarks. Therefore, one future direction is to rebalance the domain distribution of the pretraining sources according to the target applications (e.g., to integrate repository-scale code data to enhance the coding capability). Meanwhile, the current RL training employs a generative reward model that provides binary feedback based on match with the ground truth. While this reward exhibits high performance and stability for RL training, it introduces a substantial extra inference cost, becoming one bottleneck for scaling up. Future work can explore more efficient reward models to further scale up RL training to larger models and datasets.

7 REPRODUCIBILITY STATEMENT

Our Webscale-RL data pipeline is built upon publicly available datasets and publicly available LLMs for generation and verification. The detailed data sources, prompts, and implementation details are described in Appendix B.1 and Appendix B.2. For the continual pretraining and RL finetuning, we use the standard pretraining and RL algorithms (GRPO), and the hyperparameters are detailed in Appendix B.3.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Wasi Uddin Ahmad, Sean Narenthiran, Somshubra Majumdar, Aleksander Ficek, Siddhartha Jain, Jocelyn Huang, Vahid Noroozi, and Boris Ginsburg. Opencodereasoning: Advancing data distillation for competitive coding. *arXiv preprint arXiv:2504.01943*, 2025.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Gregor Bachmann and Vaishnavh Nagarajan. The pitfalls of next-token prediction. *arXiv preprint arXiv:2403.06963*, 2024.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3, 2025.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Cosmopedia, 2024. URL https://huggingface.co/datasets/HuggingFaceTB/cosmopedia.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, et al. Llama-nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949*, 2025.

Zhepeng Cen, Yao Liu, Siliang Zeng, Pratik Chaudhari, Huzefa Rangwala, George Karypis, and Rasool Fakoor. Bridging the training-inference gap in llms by leveraging self-generated tokens. *arXiv preprint arXiv:2410.14655*, 2024.

Zhepeng Cen, Yihang Yao, William Han, Zuxin Liu, and Ding Zhao. Behavior injection: Preparing language models for reinforcement learning. *arXiv preprint arXiv:2505.18917*, 2025.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I Abdin. On the diversity of synthetic data and its impact on training large language models. *arXiv preprint arXiv:2410.15226*, 2024a.

Haolin Chen, Yihao Feng, Zuxin Liu, Weiran Yao, Akshara Prabhakar, Shelby Heinecke, Ricky Ho, Phil Mui, Silvio Savarese, Caiming Xiong, et al. Language models are hidden reasoners: Unlocking latent reasoning capabilities via self-rewarding. *arXiv preprint arXiv:2411.04282*, 2024b.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024c.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Together Computer. Redpajama: an open dataset for training large language models, 2023. URL `https://github.com/togethercomputer/RedPajama-Data`.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. Deepseek-v3 technical report. *CoRR*, abs/2412.19437, 2024. doi: 10.48550/ARXIV.2412.19437. URL `https://doi.org/10.48550/arXiv.2412.19437`.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL `https://doi.org/10.48550/arXiv.2501.12948`.

Qingxiu Dong, Li Dong, Yao Tang, Tianzhu Ye, Yutao Sun, Zhifang Sui, and Furu Wei. Reinforcement pre-training. *CoRR*, abs/2506.08007, 2025. doi: 10.48550/ARXIV.2506.08007. URL `https://doi.org/10.48550/arXiv.2506.08007`.

Run-Ze Fan, Zengzhi Wang, and Pengfei Liu. Megascience: Pushing the frontiers of post-training datasets for science reasoning. *arXiv preprint arXiv:2507.16812*, 2025.

Dylan J Foster, Adam Block, and Dipendra Misra. Is behavior cloning all you need? understanding horizon in imitation learning. *Advances in Neural Information Processing Systems*, 37:120602–120666, 2024.

Wikimedia Foundation. Wikimedia downloads. URL `https://dumps.wikimedia.org`.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL `https://zenodo.org/records/12608602`.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*, 2024.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. Lighteval: A lightweight framework for llm evaluation, 2023. URL `https://github.com/huggingface/lighteval`.

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

Siming Huang, Tianhao Cheng, Jason Klein Liu, Jiaran Hao, Liuyihan Song, Yang Xu, J. Yang, J. H. Liu, Chenchen Zhang, Linzheng Chai, Ruifeng Yuan, Zhaoxiang Zhang, Jie Fu, Qian Liu, Ge Zhang, Zili Wang, Yuan Qi, Yinghui Xu, and Wei Chu. Opencoder: The open cookbook for top-tier code large language models. 2024. URL `https://arxiv.org/pdf/2411.04905`.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL `https://github.com/huggingface/open-r1`.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao

Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, and Ilge Akkaya. Openai o1 system card. *CoRR*, abs/2412.16720, 2024. doi: 10.48550/ARXIV.2412.16720. URL `https://doi.org/10.48550/arXiv.2412.16720`.

Minqi Jiang, JoÃĢo GM AraÃšjo, Will Ellsworth, Sian Gooding, and Edward Grefenstette. Generative data refinement: Just ask for better data. *arXiv preprint arXiv:2509.08653*, 2025.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21314–21328, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

Siheng Li, Kejiao Li, Zenan Xu, Guanhua Huang, Evander Yang, Kun Li, Haoyuan Wu, Jiajia Wu, Zihao Zheng, Chenchen Zhang, Kun Shi, Kyrierl Deng, Qi Yi, Ruibin Xiong, Tingqiang Xu, Yuhao Jiang, Jianfeng Yan, Yuyuan Zeng, Guanghui Xu, Jinbao Xue, Zhijiang Xu, Zheng Fang, Shuai Li, Qibin Liu, Xiaoxue Li, Zhuoyu Li, Yangyu Tao, Fei Gao, Cheng Jiang, Bo Chao Wang, Kai Liu, Jianchen Zhu, Wai Lam, Wayyt Wang, Bo Zhou, and Di Wang. Reinforcement learning on pre-training data, 2025. URL `https://arxiv.org/abs/2509.19249`.

Haoxiong Liu, Yifan Zhang, Yifan Luo, and Andrew C Yao. Augmenting math word problems via iterative question composing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24605–24613, 2025a.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=1qvx610Cu7`.

Mingjie Liu, Shizhe Diao, Jian Hu, Ximing Lu, Xin Dong, Hao Zhang, Alexander Bukharin, Shaokun Zhang, Jiaqi Zeng, Makesh Narsimhan Sreedhar, et al. Scaling up rl: Unlocking diverse reasoning in llms via prolonged training. *arXiv preprint arXiv:2507.12507*, 2025b.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025c.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh RN, et al. Apigen: Automated pipeline for generating verifiable and diverse function-calling datasets. *Advances in Neural Information Processing Systems*, 37:54463–54482, 2024.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo, Evgenii

Zheltonozhskii, Nii Osae Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian McAuley, Han Hu, Torsten Scholak, Sebastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, Mostofa Patwary, Nima Tajbakhsh, Yacine Jernite, Carlos Muñoz Ferrandis, Lingming Zhang, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. Starcoder 2 and the stack v2: The next generation, 2024.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025. Notion Blog.

Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*, 3, 2024.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*, 2025.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*, 2025.

OpenAI. Learning to reason with LLMs. `https://openai.com/index/learning-to-reason-with-llms/`, 2024a. [Online].

OpenAI. O3 and o4 mini system card. `https://openai.com/index/o3-o4-mini-system-card/`, 2024b. URL `https://openai.com/index/o3-o4-mini-system-card/`. [Online].

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. Openwebmath: An open dataset of high-quality mathematical web text, 2023.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a. URL `https://openreview.net/forum?id=n6SCkn2QaG`.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024b.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, et al. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. *arXiv preprint arXiv:2504.03601*, 2025.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adri Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*, 2023.

Bryan Stroube. Literary freedom: Project gutenberg. *XRDS: Crossroads, The ACM Magazine for Students*, 10(1):3–3, 2003.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*, 2024.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Maurice Weber, Dan Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. Redpajama: an open dataset for training large language models. *Advances in neural information processing systems*, 37:116462–116492, 2024.

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I Wang. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. *arXiv preprint arXiv:2402.09739*, 2024.

xAI. Grok 4. https://x.ai/news/grok-4, 2025. [Online].

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024b.

Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Ilia Kulikov, Kyunghyun Cho, Dong Wang, Yuandong Tian, Jason E Weston, et al. Naturalreasoning: Reasoning in the wild with 2.8 m challenging questions. *arXiv preprint arXiv:2502.13124*, 2025.

Xiang Yue, Tianyu Zheng, Ge Zhang, and Wenhu Chen. Mammoth2: Scaling instructions from the web. *Advances in Neural Information Processing Systems*, 37:90629–90660, 2024.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*, 2024.

Shaokun Zhang, Yi Dong, Jieyu Zhang, Jan Kautz, Bryan Catanzaro, Andrew Tao, Qingyun Wu, Zhiding Yu, and Guilin Liu. Nemotron-research-tool-n1: Tool-using language models with reinforced reasoning. *arXiv preprint arXiv:2505.00024*, 2025a.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Fan Zhou, Zengzhi Wang, Qian Liu, Junlong Li, and Pengfei Liu. Programming every example: Lifting pre-training data quality like experts at scale. *arXiv preprint arXiv:2409.17115*, 2024.

Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P Xing. Megamath: Pushing the limits of open math corpora. *arXiv preprint arXiv:2504.02807*, 2025.

A Usage of LLMs

In paper writing, the LLMs are mainly used for proofreading and polishing the language, including grammar, spelling, and clarity. The main content, ideas, experiments and following presentations (e.g., results and visualizations) are done by the authors. The LLMs assist to draft the results analysis and conclusion sections based on the experimental results provided by the authors. The authors carefully checked the content and made necessary modifications to ensure the accuracy and correctness of the statements.

B Details of Dataset Construction and Training

B.1 Webscale-RL Data Pipeline Details

Our data pipeline employs GPT-4.1-mini for domain classification and quality checking, while utilizing GPT-4.1 for QA generation to ensure higher quality outputs. In the second stage, we assign up to 3 different personas to each document and generate tailored QA pairs for each persona respectively.

B.1.1 Prompt Templates

Our pipeline consists of four main stages, each with carefully designed prompts to ensure high-quality data generation:

Stage 1: Data Filtering

Role: Data Analyst
Objective: Identify whether the material meets quality criteria for QA generation

Prompt:

> *You are a helpful data analyst. You will be given a material which can come from very diverse sources and may not be well-structured. In this stage, your task is to identify whether the material is qualified for the following criteria:*
>
> - *The material is informative and self-contained for the user*
> - *It's possible to extract question and corresponding answer from the material*
> - *The content has sufficient depth and clarity*
>
> *Based on the above instructions, identify whether the material is qualified or not.*
> {Raw Document}

Stage 2: Domain Classification & Persona Assignment

Role: Data Analyst
Objective: Classify domain and identify target personas

Prompt:

> *You are a helpful data analyst. You will be given a material which can come from very diverse sources and may not be well-structured. In this stage, your task is to identify the domain and persona of the material.*
>
> *Here are the instructions for the domain and persona:*
>
> - *The domain is the main topic of the material. You should choose from the following domains: {All Domains}*
> - *The persona is the intended audience of the material. If the material is intended for multiple personas, you should list several personas that will be interested in the material*
>
> *Based on the above instructions, identify the domain and persona of the material.*
> {Raw Document}

Stage 3: QA Generation

Role: Domain Expert (Persona-specific)
Objective: Generate high-quality question-answer pairs from source material
Prompt:

You will be given a material which can come from very diverse sources and may not be well-structured. In this stage, your task is to generate a question and answer pair from the material.

Here are the instructions for the question and answer generation:

- *You will act as a given persona. You should generate a question and answer pair from your perspective*
- *Both the question and answer should be totally from the material. Do not generate any information that is not in the material*
- *You should generate such a question that its corresponding answer is relatively short and can be easily and clearly verified*
- *Ensure the question is natural and reflects genuine curiosity from the target persona*

{Few-shot Examples}

Based on the above instructions and examples, generate a question and answer pair from the material.

{Raw Document}
{Persona}

Stage 4: Quality Check

Role: Data Labeler
Objective: Verify QA pair correctness and detect information leakage
Prompt:

You are a data labeler. You will be given a material and a question and answer pair generated from the material. Your task is to check whether the question and answer pair is correct according to the material and whether there is info leakage from question to answer.

Here are the instructions for checking:

- *For the answer correctness, you should check whether the answer is correct according to the original material*
- *The information leakage indicates that the question explicitly provides information about the answer and then the answer can be directly obtained from the question*
- *Ensure the question requires genuine understanding of the source material*

{Few-shot Examples}

Based on the above instructions, check the QA pair extracted from the original material in terms of the answer correctness and info leakage.

{Raw Document}
{QA Pair}

B.2 WEBSCALE-RL DATASET COMPOSITION

We curate our dataset from diverse pretraining corpora to ensure comprehensive domain coverage while emphasizing reasoning capabilities. The selected sources include DCLM (Li et al., 2024), Wikipedia (Foundation), MegaMath (Zhou et al., 2025), Stack-v2 (Lozhkov et al., 2024), with additional data from OpenMathReasoning (Moshkov et al., 2025) and OpenCodeReasoning (Ahmad et al., 2025) following SmolLM3 (Bakouch et al., 2025) protocols.

Table 3: Source distribution of the Webscale-RL dataset (∼1.2M total QA pairs)

Source Dataset	# of Converted QA Pairs
DCLM	∼550K
Wikipedia	∼350K
MegaMath	∼100K
OpenMathReasoning	∼100K
Stack-v2	∼50K
OpenCodeReasoning	∼50K

B.2.1 Data Conversion Example

The following example demonstrates our persona-driven conversion process:

Original Wikipedia Document: Alterna Bank

CS Alterna Bank (), operating as Alterna Bank (), is a Canadian direct bank and a wholly owned subsidiary of the Ontario-based credit union Alterna Savings. The bank offers chequing and high-interest savings accounts and mortgages.
Operating primarily as a direct bank since 2017, most customers access accounts using the bank's website, telephone service, and mobile apps. Unlike most other direct banks, some accounts can also be accessed through branches. There are two Alterna Bank locations in Gatineau, QC, and Alterna Savings branches also administer deposits and loans on its behalf...
The bank originated as the Civil Service Loan Corporation, founded 29 October 1992 and operating as CS Loan Corporation. It became CS Alterna Bank after receiving letters patent of continuation on 2 October 2000 as a federally regulated institution under the Bank Act...
Alterna Bank is a member of Canada Deposit Insurance Corporation (CDIC)...

Converted QA Pair: Financial Analyst Persona

Question: In examining the regulatory protection for depositors, is Alterna Bank a member of the Canada Deposit Insurance Corporation (CDIC)?
Answer: Yes, Alterna Bank is a member of Canada Deposit Insurance Corporation (CDIC).

Converted QA Pair: Commerce Student Persona

Question: In Canadian direct banking, what is notable about the way Alterna Bank allows its customers to access their accounts compared to most other direct banks?
Answer: Some Alterna Bank accounts can be accessed through branches, unlike most other direct banks.

B.3 Training Implementation Details

B.3.1 Baseline Implementation

Generative Refinement Baseline: Following (Jiang et al., 2025), we adapt their safety-focused approach to quality improvement. GPT-4.1 processes each document by: (1) assessing content quality similar to our filtering stage, returning original text if adequate; (2) refining documents by removing non-informative sections or discarding low-quality content entirely.

SFT Dataset Construction: Our 10K SFT dataset enhances instruction-following capabilities post-continual pretraining and provides RL training warmup. We sample 10K queries from a held-out Webscale-RL subset with no training overlap. Since original answers are concise, GPT-4.1 generates detailed Chain-of-Thought explanations based on ground truth, reducing hallucination compared to full model distillation.

B.3.2 Training Hyperparameters

Table 4: Training configuration and hyperparameters

Training Stage	**Hyperparameter**	**Value**
Continual Pretraining	Batch Size	256
	Learning Rate	1×10^{-5}
	Max Input Length	4096
Supervised Fine-tuning	Batch Size	128
	Learning Rate	5×10^{-6}
	Max Input Length	4096
Reinforcement Learning	Batch Size	256
	Learning Rate	5×10^{-6}
	Samples per Query	16
	Max Rollout Length	2560
	Algorithm	GRPO (Shao et al., 2024)

All experiments use AdamW optimizer with VeRL (Sheng et al., 2025) as the training backend. RL training employs binary rewards, where an LLM judges whether generated answers match ground truth responses.

B.3.3 Evaluation Framework

Table 5: Evaluation benchmarks and configurations

Benchmark	**Framework**	**Shots**	**Domain Focus**
MMLU-Pro	LM-Eval	5	Multi-domain Knowledge
BigBench	LM-Eval	0	Reasoning & Language
GPQA-D	LightEval	0	Scientific Reasoning
MATH500	LightEval	0	Mathematical Problem Solving
GSM8K	LM-Eval	8	Grade School Math
MBPP	EvalPlus	0	Python Programming
EvalPlus	EvalPlus	0	Code Generation & Testing

We employ LM-eval-harness (Gao et al., 2024), LightEval (Habib et al., 2023), and EvalPlus (Liu et al., 2023) with default settings for prompt templates, metrics, and decoding parameters. MMLU-Pro and GSM8K use few-shot evaluation (5 and 8 shots respectively) following standard protocols, while other benchmarks use zero-shot evaluation.