

Synthetic Data Generation & Multi-Step RL for Reasoning & Tool Use

Anna Goldie & Azalia Mirhoseini
Department of Computer Science
Stanford University
{agoldie, azalia}@cs.stanford.edu

Hao Zhou & Irene Cai
Google DeepMind
Mountain View, CA, USA
{haozhouml, irenecai}@google.com

Christopher D. Manning
Department of Computer Science
Stanford University
{manning}@cs.stanford.edu

Abstract

Reinforcement learning has been shown to improve the performance of large language models. However, traditional approaches like RLHF or RLAIIF treat the problem as single-step. As focus shifts toward more complex reasoning and agentic tasks, language models must take multiple steps of text generation, reasoning and environment interaction before generating a solution. We propose a synthetic data generation and RL methodology targeting multi-step optimization scenarios. This approach, called Step-Wise Reinforcement Learning (SWiRL), iteratively generates multi-step reasoning and tool use data, and then learns from that data. It employs a simple step-wise decomposition that breaks each multi-step trajectory into multiple sub-trajectories corresponding to each action by the original model. It then applies synthetic data filtering and RL optimization on these sub-trajectories. We evaluated SWiRL on a number of multi-step tool use, question answering, and mathematical reasoning tasks. Our experiments show that SWiRL outperforms baseline approaches by 21.5%, 12.3%, 14.8%, 11.1%, and 15.3% in relative accuracy on GSM8K, HotPotQA, CofCA, MuSiQue, and BeerQA, respectively. Excitingly, the approach exhibits generalization across tasks: for example, training only on HotPotQA (text question-answering) improves zero-shot performance on GSM8K (a math dataset) by a relative 16.9%.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in Natural Language Processing (Gemini Team et al., 2024; Anthropic, 2024; OpenAI et al., 2024). However, they often struggle to answer complex queries that require reasoning and tool use across multiple steps (Wu et al., 2024), such as multi-hop question-answering, mathematical problem-solving, coding, and other agentic tasks, (Yang et al., 2018; Trivedi et al., 2022; Wu et al., 2024; Cobbe et al., 2021; Jimenez et al., 2024; Ehrlich et al., 2025; Li et al., 2022).

Traditional reinforcement learning (RL) approaches, such as RL From Human Feedback (RLHF) (Christiano et al., 2023), RL from AI Feedback (RLAIF) (Bai et al., 2022), and RL from Execution Feedback (RLEF) (Gehring et al., 2025), have focused on single-step optimization, leaving the challenge of multi-step tasks largely unaddressed. Many real-world problems require a sequence of interrelated actions; for example, when answering a challenging question, a model must determine not just what information to seek, but when to stop searching and synthesize its findings. Multi-step reasoning creates a compounding challenge, as incorrect intermediate steps often lead to incorrect final results, making it critical to maintain accuracy across the entire chain of actions or learn to effectively recover from such errors.

To address this challenge, we present Step-Wise Reinforcement Learning (SWiRL), an offline multi-step optimization technique. We consider a setting where the model has access to a tool, such as a search engine or calculator, and can run a sequence of tool use calls as needed to answer the question. Our goal is to teach the model how to decompose complex problems into a sequence of more manageable subtasks, when to call the tool, how to formulate a call to the tool, when to use the results of these queries to answer the question, and how to effectively synthesize its findings. In particular, we propose a two stage approach, in which we first generate multi-step synthetic data and then learn from these data using a step-wise reinforcement learning method. This approach has the key practical advantage that we can quickly generate large volumes of multi-step training data via parallel calls to avoid throttling the training process with slow tool use execution. In addition, this offline process enables greater reproducibility due to having a fixed dataset.

To generate multi-step synthetic training data, we provide an open-source LLM (Gemma 2 (Gemma Team et al., 2024b)) with access to a relevant tool (e.g., a search engine or calculator). We iteratively prompt the model to generate multi-step trajectories; at each step, the model is free to generate a chain of thought, and may either call a tool or produce a final answer, which we refer to as the model’s action. If the model generates a tool use call, its query is automatically extracted from the overall response and executed in the environment, and the result is presented to the model in the next step. The trajectory ends when the model generates an answer to the original question, which it indicates using special markers. We convert each trajectory with k actions into k subtrajectories, containing the context from the beginning of the trajectory up to that action. We then use a step-wise reinforcement learning approach to optimize over this dataset, employing a generative reward model that evaluates each action in the context of its subtrajectory.

This granular approach enables us to apply direct feedback after each step of the trajectory, and to do so in a manner that is contextually aware. Unlike prior RL finetuning approaches used in frontier open-source models like DeepSeek-R1 (DeepSeek-AI and others, 2025) and Llama-3 (Grattafiori et al., 2024), we do not solely optimize for final performance, and use no golden labels; however, by optimizing for the reasonableness of each step given prior steps, SWiRL does in fact improve final performance.

In addition to evaluating SWiRL on challenging multi-hop question-answering and mathematical problem-solving tasks, we also study the generalization properties of this methodology. This is of key interest because there is an explosion of agentic applications for language models, and methods that generalize across datasets and tasks will be easier, cheaper and faster to adapt to new environments. We also measure the effectiveness of different synthetic data filtering strategies, study SWiRL’s ability to generalize across datasets and tasks, measure the impact of model size and dataset size, and explore the mechanism driving these performance improvements.

Our contributions are as follows:

- We propose Step-Wise Reinforcement Learning (SWiRL), an approach to synthetic data generation and offline RL that advances multi-step reasoning and tool use.
- We demonstrate generalization across datasets. For example, training SWiRL on HotPotQA not only improves performance on the dataset itself, but also yields superior performance on other multi-hop question-answering datasets, e.g., 21.5% on GSM8K (Cobbe et al., 2021), 15.3% on BeerQA (Qi et al., 2021b), 11.1% on MuSiQue (Trivedi et al., 2022) and 14.8% on CofCA (Wu et al., 2024).
- We also show transfer across disparate tasks, namely mathematical reasoning to question-answering and vice versa. Training only on multi-hop HotPotQA question-answering improves performance on GSM8K (Cobbe et al., 2021) (a math dataset) by 16.9%, and training on GSM8K improves performance on HotPotQA (multi-hop question-answering) by 9.2%.
- We analyze the impact of synthetic data filtering strategies in a multi-step reasoning and tool use setting, and demonstrate that models learn best from datasets which have been filtered step-wise to ensure high-quality reasoning traces, but which are not filtered by outcome (correct final answer).

- We explore the impact of training dataset size and model size on SWiRL, observing that significant gains can be achieved even with just 1000 trajectories and that smaller models (Gemma-2-2b and 9b) can benefit from in-domain SWiRL, but do not display the same generalization as their larger counterpart, Gemma-2-27b.
- We demonstrate that SWiRL effectively improves the average process reward, even when evaluated on out-of-distribution tasks, suggesting that the downstream performance gains are driven by improved multi-step reasoning.

2 Methodology

Our methodology, Step-Wise Reinforcement Learning (SWiRL), consists of two stages. In the first stage, we generate and filter synthetic data. In the second stage, we use a step-wise reinforcement learning approach to optimize a generative base model on the synthetic trajectories. SWiRL does not require golden labels or human annotations, and instead relies entirely on model-based judgments for data generation, filtering, and RL optimization. The overall flow of our methodology is depicted in Figure 1 (Stage 1) and Figure 2 (Stage 2).

2.1 Multi-Step Data Collection

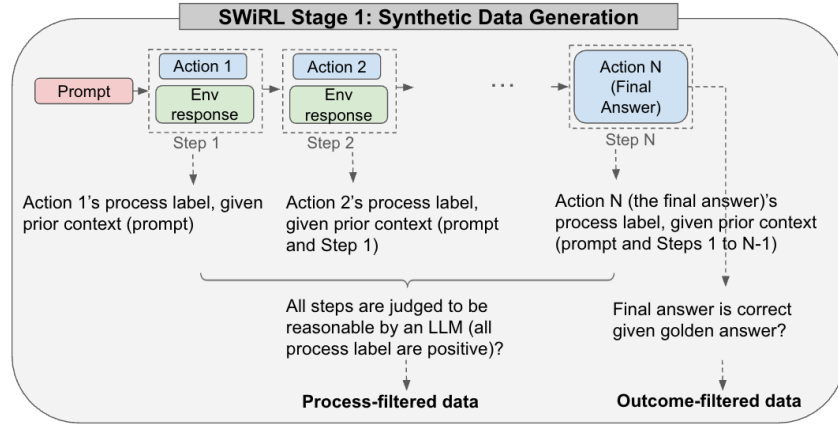


Figure 1: In SWiRL Stage 1, we generate and filter multi-step synthetic trajectories. At each step, the model is free to generate a chain of thought, call a tool such as a search engine or calculator, and/or produce an answer to original question. Process-filtered data corresponds to trajectories in which every step is judged to be reasonable by a model judge (Gemini 1.5 Pro Thinking). Outcome-filtered data corresponds to trajectories with a final answer that matches the golden label.

In Stage 1 (see Figure 1), we generate synthetic trajectories consisting of multiple steps of reasoning and tool use, which we use as training data for the step-wise RL methodology described in the next section. To compile a large-scale collection of synthetic trajectories, we augment a language model with a tool (e.g., a search engine or calculator), and iteratively prompt the model to generate multi-step trajectories. At each step, the model is asked to choose whether to call a tool or produce a final answer, and is always free to generate chains of thought (which it typically does). If the model generates a tool use call, it is parsed from the overall response, executed in the environment, and the result is presented to the model in the next step. See Appendix A for the prompt, which contains a question, explicit instructions regarding multi-step tool utilization, and the results of prior tool use calls.

For each multi-step synthetic trajectory, we define the following annotations. The trajectory itself is denoted by $\tau = (s_1, a_1, \dots, s_K, a_K)$. The first state s_1 is the original prompt. Each following state s_i contains the entire context so far, containing state s_{i-1} , action a_{i-1} , and the environment (tool call) response to a_{i-1} . Each action a_i is the model response, given state s_i . The last action, a_K , is the model's answer to the original prompt.

In this work, we compiled a dataset of 50,000 synthetic trajectories seeded by 10,000 multi-step questions from the HotPotQA training set (Yang et al., 2018) (i.e., 5 trajectories per question), and a mathematical reasoning dataset of 37,500 synthetic trajectories seeded by the 7,500 questions in the GSM8K training set (Cobbe et al., 2021). Note that, for HotPotQA, we filtered out “Easy” questions, which can typically be answered with a single search query. To prevent synthetic trajectories from being excessively long, we set a maximum step count of 5 for HotPotQA questions, and 10 for GSM8K questions (which typically require 2-8 steps to solve).

Having compiled these datasets, we consider four different filtering strategies and measure their impact on performance (Figure 1): (1) No filtering; (2) Process filtering, where we retain trajectories in which each step was deemed reasonable given all previous steps. Concretely, a model (Gemini 1.5 Pro Thinking, in our case) is prompted to render a binary judgment as to whether action a_i is reasonable given the context s_i . See Appendix A for our prompt. No golden labels are used; (3) Outcome filtering, where we select trajectories based solely on whether the final response, a_K , matches the golden answer; and (4) Process and outcome filtering, in which we take the intersection of both filtering approaches and retain only trajectories that exhibit both step-wise soundness and correct final outcomes.

Recent approaches to synthetic data distillation, such as Deep-Seek R1 (DeepSeek-AI and others, 2025), have demonstrated that synthetic data filtered for correct outcomes can lead to good performance with single-step RL and supervised finetuning (SFT). In this work, we sought to explore whether this pattern would hold in a multi-step, tool use setting, and to explore the impact of both outcome and process filters. Like these prior work, we observed that filtering multi-step trajectories for correctness was effective for SFT, and in fact critical for good performance. However, we found that SWiRL, unlike SFT, can learn even from trajectories that end in incorrect final answers. In fact, we achieve our best results by including process-filtered data, regardless of the correctness of the outcome.

2.2 Step-Wise Reinforcement Learning Methodology

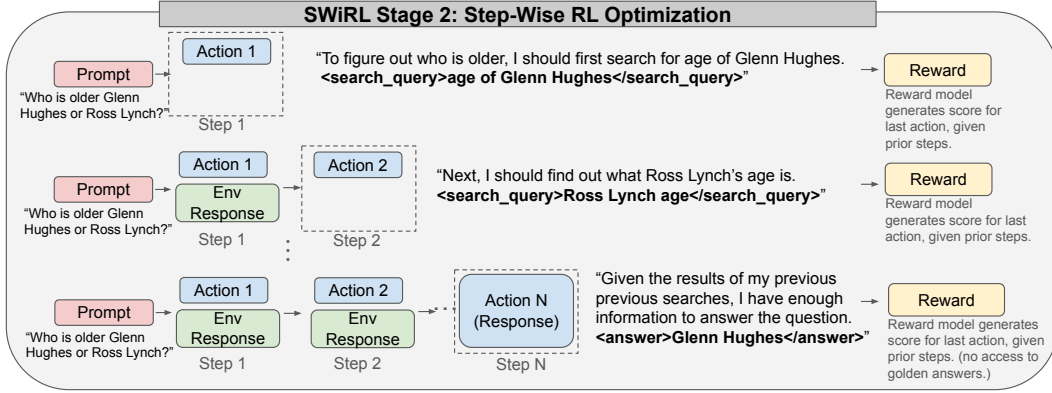


Figure 2: In SWiRL Stage 2, we perform step-wise RL to train on the synthetic multi-step trajectories from Stage 1. Each step contains an action, which corresponds to a tool call or the final response. The model is free to generate chains of thought during each step. The environment responses are captured in the prior steps of the synthetic trajectories, which were generated offline. Granular feedback is provided by a generative reward model, which is used to perform RL optimization directly on each action, given the prior context.

As shown in Figure 2, we propose a RL approach capable of learning effectively from the synthetic multi-step trajectories generated in Stage 1. At each step, a base model is optimized to predict either the next intermediate step or the final response based on preceding context. At each step i , the model has access to the full contextual history, including the original prompt, all previous model-generated steps and any applicable environment response corresponding to those steps.

Thus, our objective function is the expected sum of stepwise rewards:

$$J(\theta) = E_{s \sim T, a \sim \pi_\theta(s)} [R(a|s)]$$

Here, π_θ is the base model parametrized by θ , which is finetuned via SWiRL (Note that we also use π_θ to generate synthetic data.) T denotes the set of all states in the synthetic multi-step trajectories, i.e. each incremental state s within each trajectory τ . The reward signal $R(a|s)$ is derived from a generative reward model, specifically Gemini 1.5 Pro in our experiments, which assesses the quality of the generated response a given the context s . No golden labels are used.

We optimize the above expected reward using the same policy gradient algorithm as used in Gemma 2 for optimizing the human feedback reward (Gemma Team et al., 2024a;b). Our granular, step-by-step finetuning paradigm enables the model to learn both local decision-making (next-step prediction) and global trajectory optimization (final response generation) while being guided by immediate feedback on the soundness of each prediction.

2.3 Step-Wise Inference-time Evaluation

As shown in Figure 3, at inference time, we iteratively prompt the model to either call a tool or produce a final answer. If the model generates a search query (indicated by `<search_query>` `</search_query>` tags), we parse out that query, embed it with a Gecko model, perform a nearest neighbor lookup in the corresponding vector database, and inject the retrieved article into the model’s context window. If the model generates a calculator tool call (indicated by `<math_exp>` `</math_exp>` tags), we parse out the mathematical expression, execute it with a SymPy interpreter, and inject the calculated results into the context window. This process terminates when the model either produces an answer (signaled by producing `<answer>` `</answer>` tags) or reaches the maximum number of queries (5 for question-answering datasets, and 10 for mathematical reasoning datasets). See Appendix E for example trajectories.

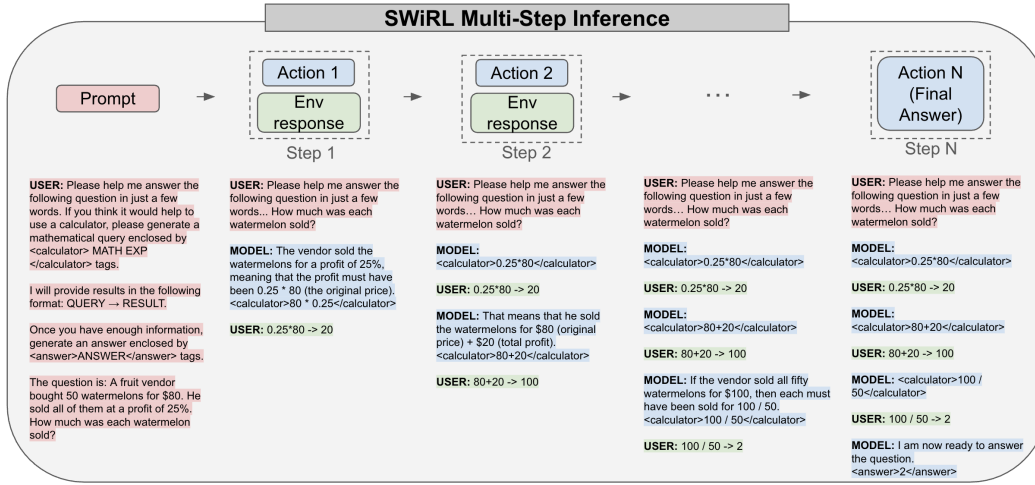


Figure 3: SWiRL Multi-Step Inference. At inference time, we iteratively prompt the model to call available tools as many times as necessary (up to a limit) before answering the original question. Here, prompts are truncated for clarity and to adhere to space limitations, but full trajectories are available in Appendix E.

3 Related Work

Reinforcement Learning for LLM Finetuning. One prominent approach, Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022; Christiano et al., 2023), con-

sists of training a reward model on human preference labels at the response level, followed by RL optimization using Proximal Policy Optimization (PPO) (Schulman et al., 2017). Building upon this framework, Reinforcement Learning with AI Feedback (RLAIF) (Bai et al., 2022) has emerged as a scalable alternative that leverages AI models to generate feedback based on predefined principles or constitutions, reducing the need for costly human annotations. RL from Execution Feedback (RLEF) (Gehring et al., 2025) uses environment feedback, such as pass rate on coding test cases, to calculate the reward, which it then optimizes via PPO. Besides PPO, other RL optimizations, such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) and its successors (e.g., Azar et al. (2023); Ethayarajh et al. (2024); Meng et al. (2024); Lanchantin et al. (2025)) as well as GRPO (Shao et al., 2024) have also proven to be effective for finetuning LLMs to maximize a target reward. A limitation of the above approaches is that they focus on single-step optimization with the reward being calculated only at the end of the episode, leading to suboptimal performance for multi-step optimization (Liu et al., 2024; Wang et al., 2024). In SWiRL, we focus on scenarios where multiple steps of reasoning and tool calls are necessary prior to generating a response. Unlike the above methods, SWiRL enables the model to receive feedback on its granular stepwise actions which leads to better multi-step reasoning and tool use across longer horizons.

Multi-step Optimization with RL. Recent work including DQO (Liu et al., 2024) and OREO (Wang et al., 2024) propose offline reinforcement learning to improve multi-step reasoning for LLMs. However, neither focuses on enhancing a model’s ability to use tools or interact with an external environment. Additionally, unlike our approach, which optimizes at the (reasoning) step level, DQO relies on token-level actions, which as shown in (Wang et al., 2024), are generally less effective than step-level actions. Moreover, OREO requires training a separate value network and policy, and relies on iterative co-optimization of both models. The process of maintaining, training, and serving these two models can be prohibitively expensive, particularly for larger models. PRIME (Cui et al., 2025) proposes an online approach to improve multi-step reasoning, but does not enable tool use or offline training. Tulu-3 (Lambert et al., 2025) uses verifiable rewards to train a language model to do better at math, but requires access to golden labels.

Reasoning Improvement with Synthetic Data. Several approaches have been proposed for generating synthetic reasoning data. These methods either rely on golden labels to filter the data or use a combination of golden labels and process or outcome reward models (Zelikman et al., 2022; Singh et al., 2024). For example, STaR (Zelikman et al., 2022) generates chain-of-thoughts (CoT) for reasoning questions, filters for those that result in correct answers, and performs Supervised Fine-Tuning (SFT) on those reasoning traces. The paper also proposes an augmentation technique called “rationalization”, in which for each question the model answered incorrectly, the model is provided with the correct answer and prompted to generate a CoT that leads to that answer. Rejection finetuning (RFT) (Yuan et al., 2023) is another method that relies on collecting reasoning traces from the model and using those with correct outcomes for SFT. ReST (Gulcehre et al., 2023) demonstrates strong performance on machine translation by iteratively generating data and then finetuning on that data using either a supervised or reinforcement learning objective. $ReST^{EM}$ (Singh et al., 2024) is an extension of ReST which outperforms training on human data alone for math and coding evaluations, but which plateaus after a few iterations, presumably due to overfitting. Our method also uses a model-based approach to generate multi-step trajectories. However, we show that using a model to label the steps within each reasoning trajectory leads to higher out-of-domain generalization than using only the trajectories which contain correct final answers, meaning that we do not require golden labels. In addition, we enable the model to use tools iteratively to perform multi-hop question answering and mathematical reasoning.

Process vs. Outcome Based Optimization. There have been a number of attempts to compare the effectiveness of process and outcome-based approaches in the domain of math and reasoning (Lightman et al., 2023; Uesato et al., 2022; Snell et al., 2024). For example, (Lightman et al., 2023) showed that (Outcome Reward Models) ORMs are more effective than (Process Reward Models) PRMs at the task of ranking samples from a fixed generator model, whereas Uesato et al. (2022) demonstrated that outcome supervision yields comparable accuracy to process supervision at lower cost, but that the reasoning traces from the resulting model exhibit lower fidelity. Both rely on expensive human annotations and golden labels,

and do not explore the impact of PRMs and ORMs in reinforcement learning optimization, or the differential effect of data filtering on supervised vs. RL optimization objectives.

4 Experiments

Datasets Metrics	HotpotQA PM†	CofCA (Avg) PM†	MuSiQue PM†
Proprietary LLMs			
GPT-4	74.8	51.9	63.9
GPT-3.5	62.8	40.7	53.1
Gemini 1.0 Pro	63.5	33.3	46.9
Bing Chat	72.1	41.6	52.3
O1-preview	76.9	58.5	67.9
Open Source LLMs			
Llama 2-7b	38.5	28.9	34.2
Mistral-7b	34.9	25.6	29.2
Qwen 2-7b	39.3	30.7	33.5
Base Gemma 2-27b	58.6	31.7	35.4
SWiRL Gemma 2-27b (Ours)	67.8	39.3	43.6

Table 1: Comparison of Accuracy (PM†: Partial Match) across Multiple Datasets: **HotpotQA**, **CofCA** (Average of 2-hop, 3-hop, and 4-hop), and **MuSiQue**. Baseline results were drawn from Wu et al. (2024). The Gemma-2 models, both SWiRL and the base model, were not given access to the context documents, but were allowed to sequentially query a vector database. The SWiRL model was trained on HotPotQA using process-filtered data, and for consistency with baseline results, evaluated on GPT-4o with the same prompts as Wu et al. (2024) on 300 randomly subsampled questions. See Appendix F for example ids.

4.1 Evaluation Datasets

To evaluate performance on multi-step search tool use, we selected five challenging multi-hop question-answering and mathematical reasoning datasets:

- **HotPotQA** (Yang et al., 2018) is comprised of multi-hop questions from a variety of domains. Human annotators constructed the questions to be answerable only by combining information from multiple paragraphs of Wikipedia.
- **MuSiQue** (Trivedi et al., 2022) is a multi-hop question-answering dataset constructed by chaining together multiple single-hop questions.
- **CofCA** (Wu et al., 2024) is a multi-hop dataset constructed to be answerable only by querying a counterfactual version of Wikipedia. It contains 2- to 4-hop questions.
- **BeerQA** (Qi et al., 2021a) is an extension of HotPotQA designed to include an even greater number of hops than the original dataset.
- **GSM8K** (Cobbe et al., 2021) is a dataset composed of grade school math word problems, which typically take 2-8 steps to solve.

For question-answering datasets, we set up a vector database containing all articles from each data split using Gecko-1B with 768-dimensional embeddings (English) (Lee et al., 2024).

For the experiments in Table 1, we follow the same procedure as Wu et al. (2024), evaluating performance on 300 randomly subsampled examples from the target dataset, using the same language model as a judge (GPT4o) and the same prompt. For every other experiment in this paper, we used Gemma-2-27b as our judge, as this was more cost effective, with the exception of GSM8K for which we used Gemini 1.5 Pro as it exhibited noticeably better numeric evaluation. Model-based evaluation is emerging as a scalable and less brittle alternative to exact match and F1 metrics (Zheng et al., 2023; Gu et al., 2025), but does

introduce a new source of stochasticity into the evaluation. See Appendix D for our own manual inspection and error analysis of three different model judges.

As described in Section 2.3, for each question, we iteratively prompt the model to either call a tool or produce a final answer, and limit the maximum number of queries to 5 for question-answering datasets, and 10 for mathematical reasoning datasets.

4.2 Results and Discussion

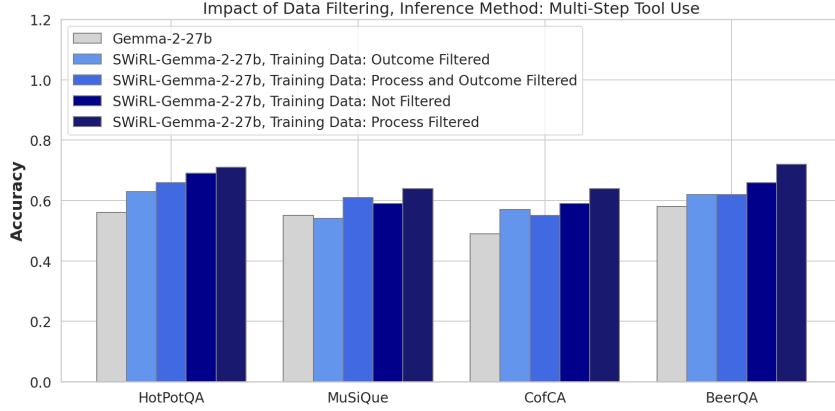


Figure 4: Impact of Data Filtering on Model Performance. Synthetic data for training is derived from HotPotQA. SWiRL learns to perform multi-hop question answering even when trained on unfiltered synthetic data. SWiRL’s best performance comes from training on process-only filtered data, where the data is selected based on the soundness of each step within its reasoning traces, but which includes both correct and incorrect responses.

Impact of Data Filtering on Model Performance: We evaluated the influence of various filtering mechanisms on downstream task accuracy, as shown in Figure 4. Concretely, we consider 4 different types of filtering: no filtering, outcome-based filtering that ensures correct final answers, process-based filtering that ensure that each step is correct as judged by a model, and both process and outcome-based filtering.

In all experiments, we fix the number of trajectories used for finetuning (with the exception of our ablation study on the impact of scaling dataset size), and we provided all models with access to an appropriate tool. Notably, process-only filtering consistently yields the highest accuracy, suggesting that focusing on the procedural aspects of data refinement is more important than the correctness of a training trajectory. While both unfiltered and filtered data demonstrated an improvement over the baseline model, filtering for correctness usually harms performance; with the exception of MuSiQue, outcome-filtered or outcome and process-filtered data is less effective than unfiltered data. We hypothesize that this is because SWiRL actually benefits from having access to both positive and negative examples. These results underscore the relative unimportance of outcome-based filtering, which requires golden labels. They also demonstrate that our process RL method can effectively learn from even trajectories with incorrect final answers.

Generalization Across Disparate Tasks: To measure generalization across training tasks, we evaluated the mathematical reasoning capabilities of a model trained on multi-hop question-answering with search tool use (HotPotQA). Specifically, we evaluated the performance of this model on GSM8K, a mathematical reasoning task, providing the model with a SymPy interpreter to use as a calculator. This experiment was run on a different random subsample of 300 examples. As shown in Table 2, applying SWiRL on out-of-distribution data and tasks still improves performance.

Effect of Tool Use: As discussed in Section 2.3, at inference time, we use the proposed multi-step eval as shown in Figure 3 and we iteratively prompt the model to make tool calls as necessary to answer the question. As shown in Figure 5, both base and SWiRL models

	GSM8K (math)	HotPotQA (qa)	CofCA (qa)	BeerQA (qa)	MuSiQue (qa)
Base Model	0.65	0.65	0.54	0.59	0.45
SWiRL on GSM8K (math)	0.79	0.71	0.56	0.68	0.49
SWiRL on HotPotQA (qa)	0.76	0.73	0.62	0.68	0.50

Table 2: SWiRL Generalization Performance. Finetuning on synthetic traces from HotPotQA or GSM8K improves performance on both in-distribution and out-of-distribution tasks. Interestingly, training on a different domain and tool (e.g. math and a calculator) improves performance on question-answering with a search engine and vice versa, suggesting the effectiveness of SWiRL in improving general multi-step reasoning and tool use capability.

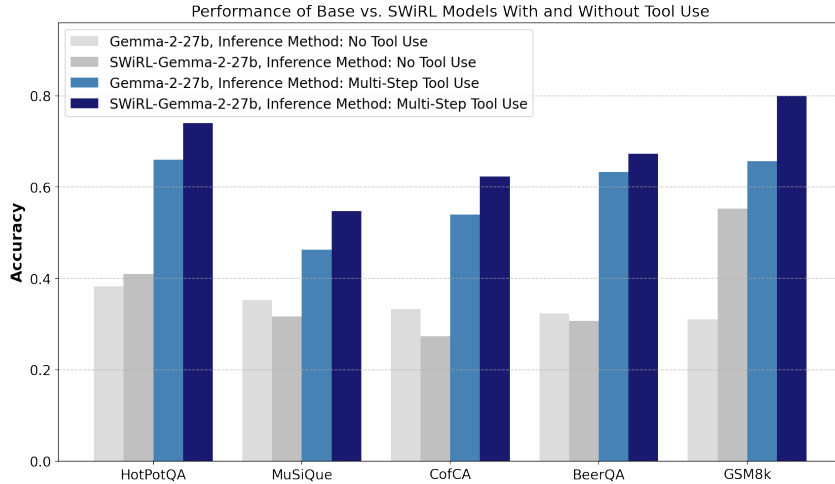


Figure 5: Performance of SWiRL With and Without Multi-Step Tool Use. SWiRL’s multi-step tool use inference improves the performance of both the base model and the SWiRL-finetuned model, but benefits the latter substantially more. Even without access to a tool, the SWiRL model outperforms, suggesting that SWiRL training improves the model’s ability to break down complex problems into multiple manageable subtasks.

improve with SWiRL’s multi-step tool use inference, but SWiRL-training offers even further improvements. Notably, the SWiRL model exhibits substantial improvements, even without access to a tool, suggesting that SWiRL training improves the model’s ability to break down complex problems into multiple manageable subtasks.

Impact of Scaling Finetuning Dataset and Model Size: Our experiments on scaling the fine-tuning dataset size reveal a clear trend: SWiRL has the ability to leverage larger datasets, even when using only process-filtered data, as shown in Figure 6. As the fine-tuning dataset size increases, a consistent enhancement in model performance is observed across our target multi-step reasoning tasks. While a limited dataset of 100 data points appears insufficient for the model to effectively generalize, a significant improvement is evident with 1,000 data points, showing solid gains across all datasets. Furthermore, scaling up to 10,000 data points continues to yield further performance enhancements, confirming the efficacy of our method in capitalizing on larger datasets for improved reasoning capabilities.

We also varied model size, observing that smaller models (2b and 9b) may benefit from in-domain SWiRL, but do not display the same generalization as their larger counterpart, Gemma 2 27b. See results in Appendix B.

Effect on Mean Process Label Accuracy: In the previous subsections, we evaluated the effect of SWiRL on downstream task accuracy. Here, we take a deeper look to understand how SWiRL achieves these performance improvements. In Table 3, we show the average process

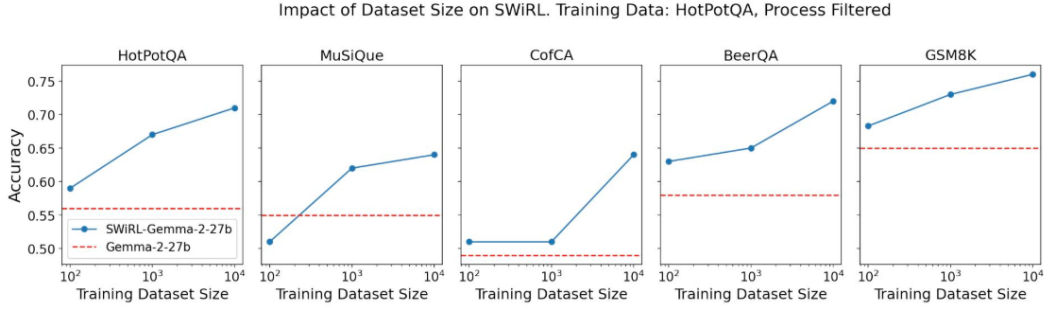


Figure 6: Performance as a Function of Synthetic Dataset Size. Synthetic training data is derived from HotPotQA, and accuracy is evaluated by Gemma 2 27b. As we scale the dataset size, we observe consistent improvements in model performance. With only 1000 data points, the model robustly improves both on in- and out- of distribution datasets.

label accuracy for the baseline model vs. a SWiRL finetuned model on 500 trajectories (seeded by 100 questions) for both HotPotQA and GSM8K. To calculate the score per step, we use the same model and prompt as we used for process filtering, as described in Section 4. We take a macro-average of the process label scores within and then across trajectories. We observe that both for in-distribution and out-of-distribution tasks, the SWiRL model generates trajectories with higher average process labels, suggesting that the higher final accuracies are driven by better multi-step reasoning.

	HotPotQA (in distribution)	GSM8K (out of distribution)
Base (Mean Process Label)	82.5%	87.5%
SWiRL on HotPotQA (Mean Process Label)	91.0%	91.6%

Table 3: Impact of SWiRL on Process Correctness. After our multi-step RL optimization, we observe that the average correctness of each step improves over the base model on both in- and out- of distribution tasks.

5 Conclusion

In this work, we propose a synthetic data generation and offline reinforcement learning approach to multi-step reasoning and tool use. This approach outperforms baselines by an average 15% across challenging multi-hop question-answering and mathematical reasoning tasks. We explore the effect of different data filtering strategies in a multi-step, tool use setting, and find that our RL approach is effective even on unfiltered data, but performs best on process-filtered data. Unlike supervised finetuning, our RL approach can learn from trajectories with incorrect final answers and actually benefits from the presence of a mixture of both correct and incorrect final answers. SWiRL demonstrates strong generalization properties, improving performance on mathematical reasoning (GSM8K) by 16.9% when trained on multi-hop question-answering (HotPotQA) and 9.2% vice versa.

References

- Anthropic. The Claude 3 Model Family: Opus, Sonnet, Haiku, 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card.Claude_3.pdf.
- Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A General Theoretical Paradigm to Understand Learning from Human Preferences, 2023. URL <https://arxiv.org/abs/2310.12036>.

- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI Feedback, 2022. URL <https://arxiv.org/abs/2212.08073>.
- Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2023. URL <https://arxiv.org/abs/1706.03741>.
- Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training, 2025. URL <https://arxiv.org/abs/2501.17161>.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process Reinforcement through Implicit Rewards, 2025. URL <https://arxiv.org/abs/2502.01456>.
- DeepSeek-AI and others. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. doi: 10.48550/arXiv.2501.12948. URL <https://arxiv.org/abs/2501.12948>.
- Ryan Ehrlich, Bradley Brown, Jordan Juravsky, Ronald Clark, Christopher Ré, and Azalia Mirhoseini. CodeMonkeys: Scaling Test-Time Compute for Software Engineering, 2025. URL <https://arxiv.org/abs/2501.14723>.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model Alignment as Prospect Theoretic Optimization, 2024. URL <https://arxiv.org/abs/2402.01306>.
- Jonas Gehring, Kunhao Zheng, Jade Copet, Vegard Mella, Quentin Carbonneaux, Taco Cohen, and Gabriel Synnaeve. RLEF: Grounding Code LLMs in Execution Feedback with Reinforcement Learning, 2025. URL <https://arxiv.org/abs/2410.02089>.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdih, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezar, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundarajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo

Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snaider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurumurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalance Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayanan Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tsihlias, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati,

Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rrustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodgkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiujia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Helendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khushen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xiang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Faret, Izhak Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirschall, Weiye Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C. Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhu Yun Xiao, Rupert Kemp, Huaixiu Steven Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Wojciech Fica,

Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard, Achintya Singhal, Thang Luong, Boyu Wang, Sujeewan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Kopparapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturel, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Vilella, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Tsendsuren Munkhdalai, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Ruddock, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnappalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Bartek Perz, Wooyeol Kim, Nandita Dukkupati, Anthony Baryshnikov, Christos Kaplanis, Xiang-Hai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Chao Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Kio-Jui Hsieh, Irene Cai, Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadisy, Prakash Shroff, Inderjit Dhillon, Tejas Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis

Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kepa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeff Dean, and Oriol Vinyals. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL <https://arxiv.org/abs/2403.05530>.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikula, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open Models Based on Gemini Research and Technology, 2024a. URL <https://arxiv.org/abs/2403.08295>.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshv, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo

Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahlteiz, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kupala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshtir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving Open Language Models at a Practical Size, 2024b. URL <https://arxiv.org/abs/2408.00118>.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelfer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yearly, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoqiang Nie, Sharan Narang, Sharath R-parthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyan Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert,

Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenber, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippas Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshchev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaoqian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The Llama 3 Herd of Models, 2024. URL <https://arxiv.org/abs/2407.21783>.

- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on llm-as-a-judge, 2025. URL <https://arxiv.org/abs/2411.15594>.
- Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced Self-Training (ReST) for Language Modeling, 2023. URL <https://arxiv.org/abs/2308.08998>.
- Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can Language Models Resolve Real-World GitHub Issues?, 2024. URL <https://arxiv.org/abs/2310.06770>.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing Frontiers in Open Language Model Post-Training, 2025. URL <https://arxiv.org/abs/2411.15124>.
- Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse Preference Optimization, 2025. URL <https://arxiv.org/abs/2501.18101>.
- Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftexhar Naim. Gecko: Versatile Text Embeddings Distilled from Large Language Models, 2024. URL <https://arxiv.org/abs/2403.20327>.
- Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d’Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-Level Code Generation with AlphaCode. *arXiv preprint arXiv:2203.07814*, 2022.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s Verify Step by Step, 2023. URL <https://arxiv.org/abs/2305.20050>.
- Guanlin Liu, Kaixuan Ji, Renjie Zheng, Zheng Wu, Chen Dun, Quanquan Gu, and Lin Yan. Enhancing Multi-Step Reasoning Abilities of Language Models through Direct Q-Function Optimization, 2024. URL <https://arxiv.org/abs/2410.09302>.
- Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple Preference Optimization with a Reference-Free Reward, 2024. URL <https://arxiv.org/abs/2405.14734>.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Floren-
cia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat,
Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao,
Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro,
Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brak-
man, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie
Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke
Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen,
Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings,
Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien
Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty

- Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rameev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 Technical Report, 2024. URL <https://arxiv.org/abs/2303.08774>.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL <https://arxiv.org/abs/2203.02155>.
- Peng Qi, Haejun Lee, Oghenetegiri “TG” Sido, and Christopher D. Manning. Answering Open-Domain Questions of Varying Reasoning Steps from Text, 2021a. URL <https://arxiv.org/abs/2010.12527>.
- Peng Qi, Haejun Lee, Oghenetegiri “TG” Sido, and Christopher D. Manning. Answering Open-Domain Questions of Varying Reasoning Steps from Text, 2021b. URL <https://arxiv.org/abs/2010.12527>.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv preprint arXiv:2305.18290*, 2023.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.

- Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. RL on Incorrect Synthetic Data Scales the Efficiency of LLM Math Reasoning by Eight-Fold, 2024. URL <https://arxiv.org/abs/2406.14532>.
- Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. Compute Trends Across Three Eras of Machine Learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, July 2022. doi: 10.1109/ijcnn55064.2022.9891914. URL <http://dx.doi.org/10.1109/IJCNN55064.2022.9891914>.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models, 2024. URL <https://arxiv.org/abs/2312.06585>.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL <https://arxiv.org/abs/2408.03314>.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop Questions via Single-hop Question Composition, 2022. URL <https://arxiv.org/abs/2108.00573>.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022. URL <https://arxiv.org/abs/2211.14275>.
- Huajie Wang, Shibo Hao, Hanze Dong, Shenao Zhang, Yilin Bao, Ziran Yang, and Yi Wu. Offline Reinforcement Learning for LLM Multi-Step Reasoning, 2024. URL <https://arxiv.org/abs/2412.16145>.
- Jian Wu, Linyi Yang, Zhen Wang, Manabu Okumura, and Yue Zhang. CofCA: A Step-Wise Counterfactual Multi-hop QA benchmark, 2024. URL <https://arxiv.org/abs/2402.11924>.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering, 2018. URL <https://arxiv.org/abs/1809.09600>.
- Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling Relationship on Learning Mathematical Reasoning with Large Language Models, 2023. URL <https://arxiv.org/abs/2308.01825>.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping Reasoning With Reasoning, 2022. URL <https://arxiv.org/abs/2203.14465>.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL <https://arxiv.org/abs/2306.05685>.

A Prompts for Synthetic Data Generation, Filtering, and Evaluation

In this work, we use the following prompts for data generation, filtering, and evaluation.

Prompt Type	Prompt Text
Prompt for Multi-Step Synthetic Data Generation for Question-Answering with Search Tool Use	<p><start_of_turn>user</p> <p>Please help me answer the following question in just a few words. If you think it would help to do a search, please generate a search query enclosed by <search_query> QUERY </search_query> tags.</p> <p>Some questions may require multiple searches in order to answer, so I will allow you to make up to {} sequential queries before answering the question.</p> <p>Please do not repeat queries you have already issued, as this is a waste of time.</p> <p>I will provide search results in the following format: QUERY → RESULT.</p> <p>Once you have enough information, generate an answer enclosed by <answer>ANSWER</answer> tags.</p> <p>Please either issue a search query or answer the question, but not both.</p> <p>The question is: {}</p> <p><end_of_turn></p>

Prompt Type	Prompt Text
Prompt for Multi-Step Synthetic Data Generation for Mathematical Reasoning with Calculator Tool Use	<p><start_of_turn>user</p> <p>Please help me answer the following question in just a few words. If you think it would help to use a calculator, please generate a mathematical query enclosed by <math_exp> MATH EXP </math_exp> tags.</p> <p>Some questions may benefit from using a calculator multiple times in order to answer, so I will allow you to make up to {} sequential queries before answering the question.</p> <p>Please do not repeat queries you have already issued, as this is a waste of time.</p> <p>I will provide results in the following format: QUERY → RESULT.</p> <p>Once you have enough information, generate an answer enclosed by <answer>ANSWER</answer> tags.</p> <p>Please either issue a search query or answer the question, but not both.</p> <p>The question is: {}</p> <p><end_of_turn></p>

Prompt Type	Prompt Text
Prompt for Process-Filtering on Multi-Step Search Tool Use Trajectories	<p><start_of_turn>user</p> <p>My boss asked me to answer the following question with the help of a search engine: {}</p> <p>This means that I might need to decompose the question into a sequence of searches before being able to answer the question.</p> <p>I am trying to learn how to do this more effectively, so please provide feedback on my last message.</p> <p>Please take a look at our conversation so far: {}</p> <p>When evaluating a message, please only consider the last message and do not penalize or reward me for previous messages.</p> <p>When evaluating an answer, please consider only whether the answer follows from the search results, and not whether you believe the answer to be correct.</p> <p>If there is not enough information from the search results to answer the question, you should rate any answer as "BAD". Pay close attention as it may initially seem like the answer is present when it is not.</p> <p>When evaluating a search query, please consider whether it is likely to help me answer the original question.</p> <p>Explain your reasoning and then answer with either "GOOD" or "BAD".</p> <p><end_of_turn></p>

Prompt Type	Prompt Text
Prompt for Evaluation / Outcome-Filtering on Multi-Step Trajectories with Search Tool Use	<p><start_of_turn>user</p> <p>I need you to help me grade the answer to the following question: "{}".</p> <p>The answer key says: {}, and my answer is {}. Am I correct?</p> <p>Please explain your reasoning and then answer "YES" or "NO".</p> <p>Do not use your own knowledge to decide, but simply check whether I gave the answer in the answer key.</p> <p><end_of_turn></p>

Prompt Type	Prompt Text
Prompt for Evaluation / Outcome-Filtering on Multi-Step Trajectories with Calculator Tool Use	<p><start_of_turn>user</p> <p>I need you to help me grade the answer to the following question: "{}".</p> <p>The answer key says: {}, and my answer is {}. Am I correct?</p> <p>Please explain your reasoning and then answer "YES" or "NO".</p> <p>There are multiple ways to write the same answer. For example, "10", "10.00", "\$10", and "\$10.00" are all equivalent.</p> <p><end_of_turn></p>

B Impact of Model Size on Effectiveness of SWiRL

The trend is that models are growing in parameter count over time (Sevilla et al., 2022), so measuring the impact of model size on the effectiveness of a method can provide insight into its longevity and future impact. It is also interesting to see whether larger models are able to learn more general patterns from the training process, and therefore exhibit greater transfer learning across datasets and even domains (e.g. math vs. question-answering). As shown in Figure 7, SWiRL demonstrates a clear performance boost over the baseline Gemma 2-27b model, showcasing consistent improvements across both in-domain (HotPotQA) and out-of-domain datasets (MuSiQue, COFCA, and BeerQA); while the 2b and 9b Gemma models also exhibit enhanced performance on in-domain data, their generalization performance on out-of-domain data is less consistent. This suggests that the effectiveness of SWiRL grows with increased model size, which is consistent with the observation that methods such as RLHF (Ouyang et al., 2022) and RLAIF (Bai et al., 2022) are more effective for larger models.

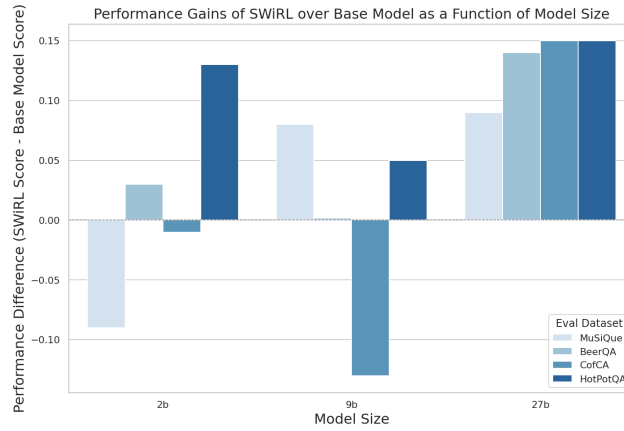


Figure 7: SWiRL Performance vs. Model Size. Synthetic data for training is derived from HotPotQA. Step-Wise RL finetuning robustly improves performance over baseline for the 27b model across both in-domain (HotPotQA) and out-of-domain datasets (MuSiQue, CoFCA, and BeerQA). However, while the in-domain improvements hold for smaller models, the out-of-domain performance is mixed, suggesting that the relative effectiveness of SWiRL is higher for larger models.

C Comparison of Supervised Finetuning and SWiRL

Figure 8 compares the performance of Supervised Fine-Tuning (SFT) and SWiRL on downstream tasks. The results show that SFT leads to worse overall performance when compared to SWiRL across all data filtering strategies. We observe that SFT performs better if we apply it to data that is both process and outcome-filtered, rather than only process-filtered. However, interestingly, SWiRL learns best from data that is only process-filtered. We attribute this to SFT’s tendency to memorize, rather than generalize (Chu et al., 2025; Setlur et al., 2024), which can hinder the model’s performance on new, unseen scenarios. In contrast, SWiRL has the ability to improve model performance by targeting per-step reward maximization. SWiRL enables the model to develop a deeper understanding of the necessary steps of query generation and retrieval, which leads to enhanced planning and generalization.

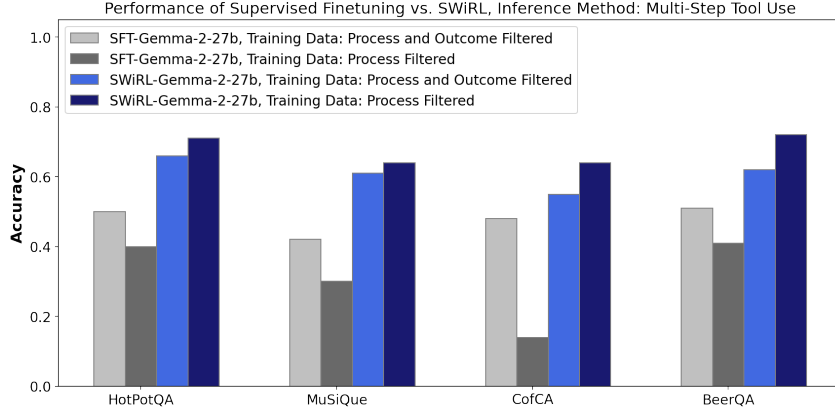


Figure 8: Comparison of SFT and SWiRL. Synthetic data for training is derived from HotPotQA, and to derive accuracy, Gemma 2 27b evaluates whether the model’s answer matches the golden answer. SWiRL greatly benefits from process-only filtered traces, and unlike SFT, is capable of learning from traces with both correct and incorrect outcomes.

D Error Analysis of Three LLM Judges

Table 4: Error Rates for Gemma-2-27b Judgments on HotPotQA (N=100)

Metric	Rate (%)
False Positive Rate (FPR)	4
False Negative Rate (FNR)	1

Table 5: Manual Analysis of LLM Math Grading Accuracy (N=100)

Model	FPR	FNR	Notes
Gemma-2-27b	15	0	Overly permissive (“nice”); all errors involved units.
GPT-4o	0	10	Overly harsh; all errors involved units.
Gemini 1.5 Pro	4	0	Accurate, slightly permissive; all errors involved units.

To evaluate the suitability of language models to serve as evaluators (i.e., check the correctness of a model answer, given a golden answer), we manually checked the correctness of 100 model judgments from Gemma-2-27b on HotPotQA questions. As shown in Table 4, we found that the error was relatively low (4% false positives and 1% false negatives), justifying the use of this low cost open-source model as our LLM judge.

However, we noticed that Gemma-2-27b made more errors when it came to numeric quantities, so we decided to run a separate analysis for GSM8K, manually evaluating 100 model judgments each for three language models (Gemma-2-27b, GPT-4o, and Gemini 1.5 Pro). Interestingly, we found that Gemma-2-27b tended to be overly “permissive” in its grading, but had zero false negatives, whereas GPT-4o had a relatively high false negative rate but no false positives. We also observed that relative results were consistent across model judges; if GPT-4o gave a higher accuracy score to a particular model, Gemma-2-27b did as well, even if the absolute scores differed. To reduce noise, we chose to use Gemini 1.5 Pro as the LLM judge for GSM8K, in spite of its higher cost.

E Example Trajectories

For clarity, we provide example trajectories from question-answering and mathematical reasoning datasets, which showcase both search and calculator tool use.

Trajectory Type	Trajectory Text
Example Multi-Hop Question-Answering Trajectory with Search Tool Use Generated by SWiRL model on HotPotQA	<p><start_of_turn>user Please help me answer the following question in just a few words. If you think it would help to do a search, please generate a search query enclosed by <search_query> QUERY </search_query> tags. Some questions may require multiple searches in order to answer, so I will allow you to make up to 5 sequential queries before answering the question. Please do not repeat queries you have already issued, as this is a waste of time. I will provide search results in the following format: QUERY → RESULT. Once you have enough information, generate an answer enclosed by <answer>ANSWER</answer> tags. Please either issue a search query or answer the question, but not both. The question is: What company published both The Scorch Trials and The Death Cure? <end_of_turn> <start_of_turn>model <search_query>the scorch trials publisher </search_query> <end_of_turn> <start_of_turn>user the scorch trials publisher -¿ The Scorch Trials is a 2010 young adult post-apocalyptic dystopian science fiction novel written by American author James Dashner and the second book, fourth chronologically, in "The Maze Runner" series. The novel was published on September 18, 2010 by Delacorte Press. It is preceded by "The Maze Runner", and followed by "The Death Cure". A was released on September 18, 2015 by 20th Century Fox. <end_of_turn> <start_of_turn>model <search_query>The Death cure publisher </search_query> <end_of_turn> <start_of_turn>user The Death cure publisher -¿ The Death Cure is a 2011 young adult dystopian science fiction novel written by American writer James Dashner and the third book, fifth chronologically, in the "Maze Runner" series. It was published on October 11, 2011 by Delacorte Press and was preceded by "The Maze Runner" and "The Scorch Trials" and followed by the series prequels, "The Kill Order and The Fever Code." <end_of_turn> <start_of_turn>model janswer¿Delacorte Press;/answer¿ <end_of_turn><eos></p>

Trajectory Type	Trajectory Text
Example Multi-Step Math Reasoning Trajectory with Calculator Tool Use Generated by SWiRL model on GSM8K	<p><start_of_turn>user Please help me answer the following question in just a few words. If you think it would help to use a calculator, please generate a mathematical query enclosed by <math_exp> QUERY </math_exp> tags. Some questions may benefit from using a calculator multiple times in order to answer, so I will allow you to make up to 10 sequential calls before answering the question. Please do not repeat queries you have already issued, as this is a waste of time. I will provide search results in the following format: QUERY → RESULT. Once you have enough information, generate an answer enclosed by <answer>ANSWER</answer> tags. Please either issue a math query or answer the question, but not both. The question is: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May? <end_of_turn> <start_of_turn>model <math_exp>48 / 2 </math_exp> <end_of_turn> <start_of_turn>user 48 / 2 -¿ 24.0 <end_of_turn> <start_of_turn>model <math_exp>48 + 24</math_exp> <end_of_turn> <start_of_turn>user 48 + 24 -¿ 72.0 <end_of_turn> <start_of_turn>model janswer¿72¿/answer¿ <end_of_turn><eos></p>

F Example Ids

For reproducibility, we provide the ids of the examples used for evaluation.

HotPotQA Example Ids			
5ab46a295542990594ba9c1c, 5a82a0fc55429954d2e2eb87, 5a83e4195542990548d0b243, 5a7f714c5542992097ad2f6e, 5aba5b2455429939ce03dc9c, 5adfbf3155429906c02daa29, 5a7ae77b554299042af8f6b0, 5a7746d025542995d83181301, 5a72321f55429971e9dc934a, 5abc3948554299700f9d782b, 5ab1e5975542997061209590, 5ac16eb355429964131be1f5, 5a8901d9554299515336125b, 5a8da5c355429941ae14dffe, 5a74fcb5542996c70cfac67, 5a7b65bf55429931da12ca86, 5a7b5d795542992d025e6825, 5a76d05542997175ce2062, 5a8658e4554299211dda2b02, 5a727f15542992359bc30c5, 5a864d835542994775f6073f, 5a8bd4375542997f31a41dd3, 5adfd4a55429906c02daa7c, 5ae4d41355429908b6326488, 5abdddeb5542991f66106083, 5ae142a4554299422ee9964a, 5adfa5405542992d7e9f93ca, 5ae18e37554299234fd0428f, 5abedcd105542993fe9a41d63, 5a7222dc55429971e9dc92c7, 5ae329e45542991a06ce993e, 5a7918ec554299148911f9ef, 5a82e7f25542995ce29dcd0a, 5ae535f5542993aec5ec17c, 5a8af82c55429950cd6afc31, 5ae69a455542996d980e7c48, 5abcb428955429959677d6a67, 5abce73b5542993a06ba9a2, 5a80901d5542996402f6a55c, 5a8a7bf5542996c9b8d5eff, 5ab5eab35542992aa134a3dd, 5a9cd955429948461cd6c, 5ae16d355429917b4a5bd18, 5ae7d44f554299540e5a5659, 5a77aa565542992a6e59df6a, 5ab5d12655429924427fcb4, 5ae24d1a5542992decdbcca6, 5adec955542992fa25da83f, 5a7c1c595542990527d55456, 5adcb8205542994ed6169bd2, 5abeaf8a5542997ec76fd346, 5a7a9b4755429941d65f26ef, 5abec3ac35542993f32c2a0ac, 5abae3eb5542996cc5e49ee2, 5ae0027b55429942ec259bda, 5ac5262755429924173fb60f, 5ab98fee554299131ca4237c, 5a7cbef0f55429909bec767ee, 5aba9cf5542994dbf01997e, 5abedbf5542993fe9a41d5f, 5adc375055429944faac246c, 5ae63c8f5542992663a4f27c, 5ac2312755429964131be2c3, 5ae1e4b554299068b959e63, 5a8053c5542992097ad2fe0, 5a8d48ff5542994ba4e3dc5a, 5a8c38235542995e66a4755f, 5a8e833d5542993f32c2a140, 5a8a73d5542996c9b8d5eee, 5a7a57935542990783324f1d, 5a7cfdda55429907fabef095, 5a7b93e05542995eb53be961, 5ac0ab335542992a796ded5d, 5ade5664554299728e26cd5, 5a82100955429926c1cdae1e,	5a85ad925542997b5ce40023, 5adc2c9355429947ff173904, 5ab3239b554299194fa93574, 5ab639c055429953192ad2aa, 5a7173b45542994082a3e83c, 5abf1fed5542990832d3a127, 5ae293bf5542996483e649fe, 5a89c2715542993b751ca990, 5ac40355542995c82c4ad0f8, 5a8a7cb255429930ff3c0df8, 5a74dca85542996c70cfae1f, 5ae5e12d55429929b08079e4, 5a89fd9e55429970aeb701e8, 5a8cad265542996e8ac88b19, 5adfa8ac55429942ec259add, 5a73870455429905862fe051, 5ab3185755429976abd1bc5f, 5a82dfa455429940e5e1a938, 5abd9fa55542996e802b4809, 5a8e2ba85542995a26add474, 5ab9bf3b554299743d22ebee6, 5ab277965542993be8fa9919, 5ae265bb5542992decdbcca, 5a76de0c5542994aec3b718d, 5a8218855542990a1d231f4e, 5a72d5155542991f9a20c5b4a, 5a7b8e3d55429927d897bfec, 5a84d29d5542994c784dd6a60, 5a80a7df554299485f59867f, 5ae221f15542994d89d5b366, 5a882caa55429975c09a596, 5a7e0bd25542997cc2e4750b, 5ab2a5fb554299545a2cf9ef, 5ac55c915542993c66e8234f, 5a8c564b554299240d9c2128, 5a8f8dfb5542997ba9cb32bb, 5ac263a25542992f1f2b38a3, 5adbfb672554299438c86cf0, 5adfdca36554299025d62a35e, 5ae73fae5542991bbc9761c9, 5ab667be55429954757d328a, 5ac1985e55429964131be248, 5a7289755542992359bc30d9, 5ae21595554299492dc91bc2, 5abff5e95542997d6429596a, 5a7ad0195542992d025e66fd, 5a7144df5542994082a3e72f, 5abfbfd00554299114383a0d4, 5a7c583e5542996dd594b910, 5a8cef7a554299441c6b9f8a, 5abbe67e5542993f40c73c05, 5a87c1ac5542997e5c09a565, 5a7639d55542992db9473748, 5adff38b55429925eb1afb7d, 5a85ec815542994775f606af, 5a8e72fe5542990e94052b14, 5ac0e564554299294b219045, 5a845bdd5542996488c2e524, 5ae11f0b5542997b2ef7d0e0, 5a792421554299148911fa09, 5ac14d9d55429964131be1ab, 5ae71816554299572ea546d1, 5ae36d325542992c3233c3f8, 5a7e6d325542991319bc94a7, 5a8db1b75542994ba4e3cd01, 5ab5f6945542994884d9a66, 5ab6ccf155429954757d3372, 5a8b002155429950cd6afc3e, 5a9063c955429933b8a2050f, 5abe225c5542991f661060ec, 5a8994505542993b751ca950, 5a88710b554299206df2b26b, 5ade469c5542992fa25da722, 5ae4a3b65542995ad6573dee, 5ac5138c5542994611c8b36a,	5a8353285542996488c2e40d, 5ab1e71b554299340b525421, 5ae1a460554299234fd042a8, 5a7c1fe4554299683c1c62cf, 5a90049d55429933b8a20468, 5addf6415542990dbb2f7f25, 5ae40a8b55429970de88d8a9, 5a7a4d845542990783324f04, 5a7dd8625542990b8f503ae8, 5ae1178e5542997b2ef7d0d6, 5ab8348d55429934fafe6d13, 5ade6bbf5542997c77adee24, 5a7917d955429974737f7982, 5add4ae25542992200553a88, 5adbff455542994650320c18, 5a8b009755429950cd6afce40, 5ac046475542996f0d89cb70, 5a8730355542991e7718170f, 5ab268aa5542993be8fa9908, 5a84f9465542991dd0999e36, 5a864dfc5542994775f6073f, 5abcea83554299114383a194, 5a84b3035542992a431d1a91, 5a7d2045554299452d57bb09, 5a732fbb5542992359bc3271, 5a722a4b55429971e9dc931f, 5a7c6ac25542996dd594b925, 5ae44eeb5542995dadf2430f, 5ab2f6b1554299545a2cfacae, 5a7f9cc25542995d8a8ddec2, 5ac1a94455429964131be262, 5ab8af3c55429916710eb0ac, 5ab3d4ae5542992ade7c6ec5, 5adfcf7655429906c02daa49, 5a89efb25542992e4fca8497, 5a811e1955429903bc27b931, 5ac5190d5542996feb3fe9f8, 5a75ddcd02554299109176e5aa, 5a7f9e0155429969796c1aee, 5a77b0795542992a6e59df89, 5a7a333f5542996a35c17130, 5a848c215542992a431d1a4f, 5a7d1dd055429909bec76960, 5a8935e6554299669944a506, 5ae07634554299603e418412, 5a7cf9b455429907fabef07c, 5ac0279c5542996f0d89cb3f, 5a7b9cac554299042af8f78f, 5a8e72f05542990e94052b13, 5a7fee435542994857a7685b, 5a8f4e8955429918e830d1f1, 5ab962ff554299131ca4231f, 5a7a2ec05542990198eaf0bc, 5ab7530b55429928e1fe3849, 5ac172a15542994d76dcce2e, 5a76133755429976ec32bcff, 5a72edeb5542992359bc31da, 5a8a28b55542996c9b8d5e23, 5abe16c655429976d4830a71, 5a80c5f6554299260e20a151, 5abf23a65542997ec76fd3d7, 5ae4bdeb55429913cc2044ee, 5a7d68045542995f4f40226d, 5ab96d865542996be20204df, 5a8d40c95542994ba4e3dc3b, 5a8f99bc55429918e830d28d, 5ae44fe75542995dadf24314, 5a76f3c65542994aec3b719a, 5a7b45c855429931da12ca4a, 5a72a6b65542994cfe9b3b7, 5ae308775542992decdbcdcd, 5ab6259855429953192ad272, 5ab318a0554299233954f07, 5ae40c3955429970de88d8c5,	5a79be835542994bb9457052, 5a7790ac5542992a6e59def9, 5a81e075554299676cce128, 5ab7cff355429928e1fe391e, 5a8e0d7e5542995085b373b4, 5a8138c155429938b6142300, 5ab457445542991751b4d748, 5ae0ef5e5542994adbac6df, 5ab48dd55542991779162cd9, 5abd08ae554299700f9d7980, 5a78f1e55429974737f7919, 5adf573c5542995534e8c798, 5adc1017554299438c868d20, 5ae026eb55429924de1b703a, 5ac31609554299741d48a1c0, 5ae62b2d5542992a0cd1625b, 5a89138255429951533612af, 5a85b3455542994c784dddb4d, 5ae5dcc755429929b08079d8, 5a87099455429960ec39b704, 5a871ce055429960ec39b704, 5a89756155429915336130b, 5a77280b5542994aec3b71ff, 5abc7af15542993a06baf8ed, 5a8024ad5542992097ad2fde, 5a7a9ca55429941d65f26f3, 5abae9cd5542996cc5e49f04, 5adb7b455429944fac23b0, 5ac29dcd554299657fa28fd, 5abe42aa55429967d4830ac2, 5a762e0f5542992d0ec06052, 5aba94465542994dbf019953, 5ac25882554299636651998c, 5a8a12555542992e4fca84f1, 5ab58009554299637185c5b2, 5a81f2955542990a1d231eee, 5a82bfbc55429954d2e2be5e, 5a8200d055429926c1cdade2, 5a7b5f6455429942af8f757, 5ac178655542994ab5c67d5a, 5ac262a055429951e9e6859a, 5a89a79c5542993b751ca970, 5ac152e755429964131be1bb, 5a831cb955429966c78a6b3f, 5ab4eb2b55429942dd415fa2, 5ae0fa52554299242ee9594, 5a88a93c5542994846c1cead, 5ab902d05542991b5579f0ca, 5a85a1015542991d0d0999eef, 5a7b4f2c55429931da12ca66, 5ac1a0e15542994ab5c67dab, 5a7b79c95542997c3e971b0, 5ac3d31a5542995ef918c249, 5a88dcef55429938390d3fe3, 5ac073eb5542996f0d89cb8d, 5ae6b38c5542992a0d16392, 5a7b663355429931da12ca87, 5ae5fb975542996de7b71aa8, 5abddcd355542992ccd8e7fc6, 5ab4136b5542996a3a969f18, 5a7e1d4255429965ceec5ea79, 5ae4a09e5542996836b02ced, 5aba88d555429901930fa811, 5ae4d2c255429960a22e01f6, 5ae5af10554299546bf82f23, 5add0ed35542990d50227dac, 5adcb67e5542994ed6169bca, 5ab5207c5542996a3a96a02b, 5a8e8b6c5542990e94052b43, 5ab7f3625542995d3e37ea06, 5ab72f32554299110f219ac3, 5ac29ca6554299218029dac0, 5ab1f75d554299340b525443, 5ab9025855429934f4e6e47,

CofCA Example Ids

5a866fee5542991e77181657, 5ac4bfd05542997ea680caab, 5ab381b155429969a97a816b, 5abaa25155429901930fa868, 5add82fc5542997545bbbd57, 5a79c7f95542994bb9457099, 5ae136fe55429920d5234325, 5ae77a31554299540e5a55c7, 5a804fc45542992bc0c4a6f0, 5ae61be055429929b0807ace, 5ae7b03e5542993210983ef6, 5a866fee5542991e77181657, 5add28c85542992ae4cec4be, 5abba27f5542996606241708, 5a7336d05542991f9a20c68d, 5ae6038155429929b0807a55, 5a76cb6e5542994aec3b717a, 5ae0fa865542997b2ef7d0c6, 5ac91eb5542996feb3fe8d2, 5ac219df5542992f1f2b37fc, 5a8e171b554299068b959e5a, 5adce28f5542990d50227d52, 5a80043055429969796c1ba0, 5a8f7de3554299458435d657, 5adc318c5542996e685252d5, 5abaf9df5542996cc5e49f45, 5a83168855429966c78a6b2e, 5abbd1c135542993f40c73bf6, 5ab3dad4554299753aec59cb, 5ac143535542991316484aac, 5abd04f155429968e802b467e, 5ae061d5554299603e41840e, 5a7a358f5542990783324ec1, 5a8ee0a35542990e94052ba0, 5a7bb3cd9554299294a54aa0, 5add992c5542997545bbbd83, 5a8b58b955429950cd6afcc2, 5a8d0006554299441c6b9fa8, 5a81ff1d554299676cceb1c3, 5abbbd0f55429931dba144d5, 5ab5141a5542991779162d70, 5ac143535542991316484aac, 5ae377155542991a06ce99c7, 5ac1688655429964b5c67d14, 5ae27edc5542992decdbcd2d, 5ae3b4d05542992f92d82349, 5a88b7735542995e715ac079, 5ac538e15542994611c8b437, 5ae31a9c55429928c42395ef, 5ade52e85542997c77adedfa, 5a7fa53c5542995d8a8ddedc, 5ab55435554299488d4d9939, 5ab3d3de2554299753aec59d6, 5ae755665542997b22f6a6e9, 5a8099025542996402f6a588, 5ae13f525542997b2ef7d169, 5a760f6855429976ec32bcf9, 5a90c2b35542995651fb51df, 5ac168865542994ab5c67d14, 5ac219df5542992f1f2b37fc, 5a879ab055429964f30887e, 5ae5365d5542992663a4f16d, 5ab9121555429919ba4e238a, 5ae5365d5542992663a4f16d, 5ac31c9d554299741d48a203, 5a7cc5ae55429909bec767fc, 5a77ff5542992a6e59df86, 5ae77a31554299540e5a55c7, 5ae136fe55429920d5234325, 5ae1f61a5542994d89d5b2e1, 5a7336d05542991f9a20c68d, 5ae2aba15542996483e64a32, 5a736bfa5542991f29ee2e03, 5ab6e856554299710c8d1fac, 5ac4fa8c55429924173fb536, 5a8beddd5542995d1e6f1468, 5abd90545542996e802b47d7, 5a7e39515542995ed0d166da	5a7db2f75542990b8f503a34, 5ac4c61a5542996feb3fe93c, 5ac2a912554299218029dae8, 5ac39a1c554299657fa290f9, 5add117e5542990d50227db2, 5ab58ae15542992aa134a357, 5a80b4635542992bc0c4a7bd, 5ae0d91e55429924de1b7198, 5ac2ffa9554299218029dbb2, 5a8ae6c055429950cd6afbcc, 5a8eacc75542995085b37473, 5a888a8a5542997e5c09a603, 5abffcf58554299012d1db552, 5ab6ad2855429953192ad35e, 5ac3b0f15542995ef918c1fc, 5ab3dde2554299753aec59d6, 5a7524ca55429929fdd850a, 5a7336d05542991f9a20c68d, 5a7fe9975542994857a76847, 5a8a84775542996c9b8d5f19, 5a79e0445542994f819ef0e7, 5a8f7de3554299458435d657, 5ac39f2a554299391541382d, 5ab74412554299110f219ae8, 5a78cdcf7554299029c4b5e9f, 5adcf28c5542994ed6169c30, 5adc134b5542994650320c5c, 5add7d05542990dbb2f7e61, 5a8dc8de55429941ae14e060, 5ac31c9d554299741d48a203, 5a72b2695542991f9a20c56f, 5ae56d31554299546bf82ed7, 5a7f38ae5542992e7d278c99, 5ae3bdfa5542990afbd1e1c0, 5abb1f745542996cc5e49fb5, 5adc1309554299438c868d3b, 5ae22d035542996483e64925, 5add82fc5542997545bbbd57, 5ae755665542997b22f6a6e9, 5a7cce9554299452d57ba72, 5ae4c2145542995dadf243e7, 5a7f98e655429969796c1ad8, 5ae614055542996de7b71b2a, 5ac1944c5542996f0d89cc90, 5ab979da554299131ca4233a, 5ab39701554299233954ff5e, 5adfd35b55429906c02daa54, 5ab520565542996a3a96a02a, 5ab67b8f55429954757d32f0, 5a7607d7554299109176e61a, 5abae3f4554299660624169c, 5a904e725542995651fb5118, 5ac3ad225542995ef918c1da, 5ae316f355429928c42395e3, 5a74248855429929fdd83e5, 5ac1f7f355429964131be2ae, 5a7f38ae5542992e7d278c99, 5a78ed46554299148911f9a6, 5a88b7735542993e715ac079, 5ab67b8f55429954757d32f0, 5ae5365d5542992663a4f16d, 5a736bfa5542991f29ee2e03, 5a8dfbeb5542995085b3736e, 5add28065542990d50227e08, 5adf2b325542993a75d2640b, 5a8a84775542996c9b8d5f19, 5ae31a9c55429928c42395ef, 5ac2ad3f554299657fa290f9, 5ae525835542990ba0bbb1cd, 5add28c85542992ae4cec4be, 5a8eacc75542995085b37473, 5abba27f5542996606241708, 5a8f0e065542997ba9cb319c, 5aba0675542994dbf0198a0, 5a8ee0a35542990e94052ba0, 5abbbd0f55429931dba144d5, 5ae3345f55429928c4239682, 5a73332b5542992359bc3287, 5ab93287554299753720f78f, 5ae3bdf45542990afbd1e1c0, 5ab93287554299753720f78f, 5ae64cab5542991bbc9760be, 5ae7b03e5542993210983ef6, 5ae64cab5542991bbc9760be, 5ab9253c554299131ca4227f, 5abee7e5542993fe9a41da0, 5a8bab4e554299240d9c207c, 5aba6b2d55429901930fa7a9, 5ac3ad225542995ef918c1da, 5ab381b155429969a97a816b, 5ade025e5542997dc790711e, 5a8b560855429950cd6afcba, 5a72b2695542991f9a20c56f, 5abd7ca05542993062266cab, 5ae0d26455429945ae959473, 5adc1309554299438c868d3b, 5a72b1c25542992359bc3172, 5a904e725542995651fb5118, 5ade8f5e55429975fa854f11, 5a7e7bf455429949594199d6, 5a90c58255429916514e756c, 5ab34af554299753aec5969, 5ae377155542991a06ce99c7, 5ae5569255429908b63265e4, 5ab59bd05542997d4ad1f190, 5ae255db5542992decdbccc1, 5ab5c9c5554299494045f065, 5ab561d85542992aa134a2fc, 5adce28f5542990d50227d52, 5adfd35b55429906c02daa54, 5a7fa53c5542995d8a8ddedc, 5a80d30655429938b61421fe, 5a79e0445542994f819ef0e7, 5a7bb3cd9554299294a54aa0, 5ae2e27155429928c423952a, 5a77bd595542995d83181291, 5a823ae45542990a1d231f6d, 5a7cedca55429909bec7689c, 5ab345db55429969a97a8122, 5add992c5542997545bbbd83, 5a8514545542992a431d1ad2, 5a74fbc5542996c70cfae63, 5ae13f525542997b2ef7d169, 5a85603a5542997b5ce3ffff1, 5ae3d8dc5542992f92d8239c, 5a879c8e5542994846c1cddb3, 5ae11a6755429901ffe4ad8d, 5abfbb455542997ec76fd440, 5ac39a1c554299657fa290f9, 5a84322b5542996488c2e50d, 5ae655c855429908198fa599, 5a8454e85542992ef85e23be, 5a77aff5542992a6e59df86, 5a7a0d455542990783324e13, 5a7a0d455542990783324e13, 5abfbb455542997ec76fd440, 5a8a35625542992d82986efd, 5ae64cbf5542992ae0d162c1, 5ae755665542997b22f6a6e9, 5ae377a35542994393b9e6db, 5adf5ebd5542995c70e8fd8, 5ab5a2f85542997d4ad1f197, 5a7738dc554299373536021f, 5a8bdef85542997f31a41dea, 5a8cdc5255429941ae14df21, 5abd7ca05542993062266cab, 5a8a2d805542996c9b8d5e2e, 5ae3345f55429928c4239682,	5ae525835542990ba0bbb1cd, 5a89372855429951533612e6, 5a7bb3cd9554299294a54aa0, 5ae655c855429908198fa599, 5ab979da554299131ca4233a, 5add7d05542990dbb2f7e61, 5ae3b4d05542992f92d82349, 5ab865be5542990c739ec8e5, 5a77153355429937353601c8, 5a72a00d5542991f9a20c53c, 5a8ee0a35542990e94052ba0, 5ae4b3da55429913cc2044d6, 5abae52a5542996cc5e49eeaa, 5abc145b554299658360041f, 5a7a06935542990198eaf050, 5a77bd595542995d83181291, 5ac17f4f5542994ab5c67d70, 5adce28f5542990d50227d52, 5a89372855429951533612e6, 5ac07a585542996f0d89cbf0, 5a8f0e065542997ba9cb319c, 5ac219df5542992f1f2b37fc, 5abc394f55429976d4830aaa, 5a7a02235542996c55b2dcd3, 5ab865be5542990c739ec8e5, 5adbe1e755429947f173853, 5a8efd3c55429918e830d179, 5a8a35625542992d82986efd, 5a7cb48a5542996d594b9a1, 5ab93287554299753720f78f, 5a7f3d325542992e7d278cb5, 5ab6e856554299710c8d1fac, 5ac061ab554299294b218fac, 5ae3d8dc5542992f92d8239c, 5a904e725542995651fb5118, 5ab39701554299233954ff5e, 5a84322b5542996488c2e50d, 5a72b2695542991f9a20c56f, 5ae4c2145542995dadf243e7, 5a8355f9554299123d8c20f3, 5abee5e2554299451645473, 5a78ed46554299148911f9a6, 5ab520565542996a3a96a02a, 5ab707c05542991d32237060, 5a88fea05542997e5c09a6e9, 5ab5e6d65542997d4ad1f232, 5adfd35b55429906c02daa54, 5ab55435554299488d4d9939, 5a7d1f605542995ab5c67d70, 5ac17f4f5542994ab5c67d70, 5ae0fa865542997b2ef7d0c6, 5a870d0255429960ec39b710, 5ab9116f5542991b5579f0db, 5a88377c5542997e5c09a5a7, 5abbc70d5542992ccd8e7f9b, 5a7738dc554299373536021f, 5a821ffa5542990a1d231f5c, 5a8514545542992a431d1ad2, 5ab39701554299233954ff5e, 5a8461d55542990548d0b29b, 5a7f9e855429969796c1af3, 5a8f8f345542997ba9cb32c2, 5ac31c9d554299741d48a203, 5adc134b5542994650320c5c, 5a8454e85542992ef85e23be, 5ac4fa8c5542992ae0d1631b, 5a84abdc55429930ff3c0d8c, 5abd7cb855429924427fd00a, 5a7a52745542996c55b2dd4f, 5ae614055542996de7b71b2a, 5ae664955542992ae0d1631b, 5ac1a5cd5542994d76dccc94, 5ae546e85542992663a4f1b5, 5a7a02235542996c55b2dcd3,
--	--	--

MuSiQue Example Ids

2hop...376129.44537, 2hop...764465.126539, 3hop1...434518.136629.55288, 2hop...353084.36340, 2hop...344450.160798,
 2hop...637856.351187, 2hop...760990.44191, 3hop1...162325.11248.3752, 2hop...326799.278127, 2hop...239927.62031,
 2hop...153813.69936, 3hop1...213491.782843.75255, 2hop...2846.2741, 2hop...3880.909, 2hop...347735.36735, 2hop...144393.87372,
 4hop1...709382.146811.31223.45305, 2hop...143434.20122, 2hop...21457.74218, 3hop1...129597.517267.451901, 2hop...469317.776926,
 2hop...27032.5400, 3hop2...83954.32417.24628, 3hop2...14790.57411.86234, 2hop...78490.49700, 3hop1...228008.354329.5303,
 2hop...631861.160851, 3hop1...662283.507729.351187, 2hop...482727.20661, 3hop1...858308.102146.84004, 2hop...565717.77346,
 3hop1...470555.668347.492654, 2hop...25478.65517, 2hop...129389.31248, 2hop...527889.5365, 2hop...20857.20779,
 2hop...770.919, 2hop...375649.80178, 3hop1...332614.131794.17114, 2hop...144295.211364, 2hop...108160.159045,
 2hop...46545.88521, 2hop...518906.44191, 2hop...733628.131886, 4hop1...28235.74795.84660.15312, 2hop...104341.92821,
 2hop...445544.127008, 2hop...46766.79233, 2hop...342213.185893, 2hop...528837.126102, 2hop...497897.541630,
 3hop1...48619.26424.581618, 2hop...87287.83906, 4hop1...411538.805015.475503.32631, 2hop...658198.72962, 2hop...42307.120207,
 2hop...30878.555599, 3hop1...8373.87072.45358, 3hop2...337255.48727.83343, 2hop...251450.8796, 3hop1...161080.639509.644660,
 2hop...558231.52667, 2hop...424189.49441, 3hop1...821692.74047.756423, 2hop...531731.79705, 3hop1...257981.259472.611044,
 2hop...370765.14904, 2hop...446352.14183, 2hop...81087.13292, 2hop...684971.333904, 2hop...234176.69926, 2hop...858097.121880,
 4hop2...724536.444580.75897.631997, 2hop...492509.70585, 4hop1...405751.4520.65397.49736, 2hop...128610.126060,
 3hop1...325154.786384.42990, 2hop...34130.56335, 2hop...145997.63766, 2hop...146446.690423, 2hop...225632.111125,
 2hop...856457.495, 2hop...129234.330515, 2hop...15674.42467, 3hop1...161946.84298.53741, 2hop...48959.83539,
 2hop...64650.20556, 3hop1...316518.395352.131877, 2hop...136618.92216, 2hop...199336.185893, 2hop...930.57555,
 3hop1...31942.48661.15069, 2hop...35105.160978, 2hop...128804.351187, 2hop...153004.86587, 2hop...715365.565667,
 2hop...401484.135138, 2hop...52622.67783, 2hop...713501.58946, 2hop...300786.39199, 2hop...5430.5348, 3hop2...29467.132027.73594,
 3hop1...225298.755188.480696, 2hop...367037.80178, 2hop...343473.53204, 2hop...848923.66214, 3hop1...369072.287321.161879,
 2hop...250315.64214, 3hop1...104311.833580.61459, 2hop...1835.322987, 3hop1...836616.291186.4303, 2hop...531924.1094,
 2hop...131831.84128, 2hop...328708.90697, 2hop...704691.82816, 2hop...80353.3001, 2hop...196785.61424, 2hop...130964.47336,
 3hop1...761109.548045.159613, 3hop1...4525.52205.55099, 3hop1...58522.787757.69397, 2hop...58284.37793, 2hop...487591.7672,
 2hop...250913.58115, 2hop...131095.85298, 2hop...144937.8600, 3hop2...625639.25582.21116, 3hop2...30023.63595.53125,
 2hop...584872.88978, 2hop...116643.351162, 2hop...826203.62031, 2hop...85036.909, 2hop...62996.299942, 2hop...236731.229413,
 2hop...15169.87091, 2hop...143791.75878, 2hop...658198.90536, 2hop...70321.15755, 2hop...131105.68117, 2hop...143162.438686,
 2hop...20771.65517, 2hop...65149.46180, 2hop...251426.88653, 3hop1...238983.403313.61770, 2hop...28291.709757,
 2hop...391909.3430, 3hop1...266733.291186.50964, 2hop...205685.160137, 2hop...343141.702969, 3hop1...383692.434040.59381,
 2hop...240975.736878, 2hop...507864.368521, 3hop1...723003.593059.76293, 2hop...109234.62766, 4hop1...16401.4520.65397.52251,
 2hop...140591.256194, 2hop...104757.74309, 2hop...194976.55566, 2hop...361127.140822, 3hop1...108774.104782.14771,
 4hop3...393686.620110.61746.261712, 2hop...324178.83854, 3hop1...849536.301867.127418, 2hop...24408.541630,
 2hop...54755.729624, 2hop...693650.61232, 3hop1...89787.49283.632017, 4hop1...104663.221169.833580.61459, 2hop...664573.36741,
 3hop1...702271.823374.26254, 2hop...129892.62851, 3hop1...659125.39490.23352, 2hop...222162.386543, 2hop...446009.412262,
 2hop...781841.77980, 3hop1...706183.20196.10585, 2hop...809948.162428, 3hop1...458602.681261.369731, 2hop...529082.114112,
 3hop1...388966.508834.145463, 2hop...582169.370960, 2hop...225632.52135, 2hop...302491.81463, 2hop...136889.52356,
 2hop...81363.42667, 3hop1...599980.544161.92922, 2hop...504710.513189, 2hop...145939.11443, 2hop...320353.4018,
 2hop...27033.85063, 2hop...145110.861627, 2hop...149891.44359, 2hop...376266.37939, 3hop2...10879.37094.161133,
 3hop2...159915.8509.19700, 4hop1...15118.31258.43153.32993, 3hop1...522518.132413.16066, 2hop...129782.517267,
 3hop1...252998.715836.26008, 4hop1...205937.144938.83779.44678, 2hop...131318.47465, 2hop...338405.68172,
 4hop3...3153.3356.11988.24628, 2hop...106465.54210, 2hop...397761.404718, 4hop1...632232.164954.6975.6891, 2hop...121872.708662,
 2hop...73501.31113, 2hop...378511.191233, 3hop1...85045.96305.25007, 3hop1...755950.592709.78102, 2hop...811421.377891,
 3hop2...63595.391767.53125, 2hop...131380.84859, 3hop1...158678.48408.37793, 3hop1...7312.830682.68600, 2hop...207212.21032,
 3hop1...10725.695397.74345, 2hop...445228.774871, 4hop1...603090.818753.783943.26110, 2hop...177131.646483,
 3hop1...801682.192919.16121, 2hop...243908.500443, 3hop2...89818.157704.4107, 2hop...160546.26427, 2hop...128772.745471,
 2hop...62588.20779, 2hop...661636.82027, 2hop...105388.89066, 2hop...368185.131944, 3hop1...153577.411195.8682,
 2hop...327451.90697, 2hop...647590.134798, 3hop2...30796.804098.24137, 2hop...146227.42328, 2hop...152881.620955,
 2hop...11693.42892, 2hop...753498.7606, 2hop...2795.2741, 3hop1...373317.533132.1660, 2hop...229374.333904,
 3hop1...370820.301867.127418, 3hop1...713250.4016.83854, 2hop...130414.68117, 4hop1...7312.84360.334118.41330,
 2hop...65149.68376, 2hop...182310.565529, 3hop1...136299.84467.89676, 2hop...454055.86874, 2hop...604878.40786,
 2hop...307569.51671, 2hop...854082.159115, 2hop...198557.55566, 3hop1...352446.506157.44678, 2hop...468848.44537,
 2hop...207571.126101, 4hop2...53235.18485.57802.311656, 2hop...451164.140822, 3hop1...37692.84298.53741,
 3hop1...672119.196807.760519, 3hop2...131210.661360.54023, 2hop...8531.24846, 3hop2...77886.64137.69951, 2hop...730762.8600,
 2hop...350323.45731, 2hop...131117.53519, 3hop1...157534.275705.81669, 2hop...185628.677577, 2hop...77119.20732,
 2hop...67755.82010, 3hop1...790278.593059.76293, 3hop2...162189.611045.73761, 2hop...568848.50788, 2hop...45625.61952,
 2hop...146207.30651, 2hop...57439.78714, 2hop...3756.52135, 3hop1...501828.348668.856982, 3hop1...106423.35178.686699,
 2hop...103203.23140, 3hop1...77985.66386.16350, 2hop...664921.579740, 2hop...106125.20644, 2hop...400998.61424,
 3hop1...35884.161545.16532, 2hop...584521.755188, 2hop...80508.400874, 2hop...664137.58115, 2hop...453207.80674,
 3hop1...29335.30907.24600, 2hop...144364.68900, 2hop...226817.482901, 4hop3...39198.75897.8509.19700, 2hop...713863.64008,
 2hop...71269.36735, 2hop...504228.64689, 2hop...604878.18657, 2hop...81372.303417, 3hop1...674688.707133.72062,
 2hop...157766.18657