Synthetic Data RL: Task Definition Is All You Need

Yiduo Guo
Peking University

Zhen Guo
MIT

Chuanwei Huang
Peking University

Zi-Ang Wang
Peking University

Zekai Zhang
Peking University

Haofei Yu
UIUC

Huishuai Zhang
Peking University

Yikang Shen
MIT-IBM

Abstract

Reinforcement learning (RL) is a powerful way to adapt foundation models to specialized tasks, but its reliance on large-scale human-labeled data limits broad adoption. We introduce **Synthetic Data RL**, a simple and general framework that reinforcement fine-tunes models using **only** synthetic data generated from a task definition. Our method first generates question and answer pairs from the task definition and retrieved documents, then adapts the difficulty of the question based on model solvability, and selects questions using the average pass rate of the model across samples for RL training. On Qwen-2.5-7B, our method achieves a 29.2% absolute improvement over the base model on GSM8K (+2.9 pp vs. instruction-tuned, +6.6 pp vs. Self-Instruct), 8.7% on MATH, 13.1% on GPQA (+7.0 pp vs. SynthLLM), 8.9% on MedQA, 17.7 % on CQA (law) and 13.7% on CFA (finance). It surpasses supervised fine-tuning under the same data budget and nearly matches RL with full human data across datasets (e.g., +17.2 pp on GSM8K). Adding 100 human demonstrations improves the performance of GSM8K only by 0.4 pp, showing a limited added value. By reducing human data annotation, Synthetic Data RL enables scalable and efficient RL-based model adaptation. Code and demos are available at `https://github.com/gydpku/Data_Synthesis_RL/`.

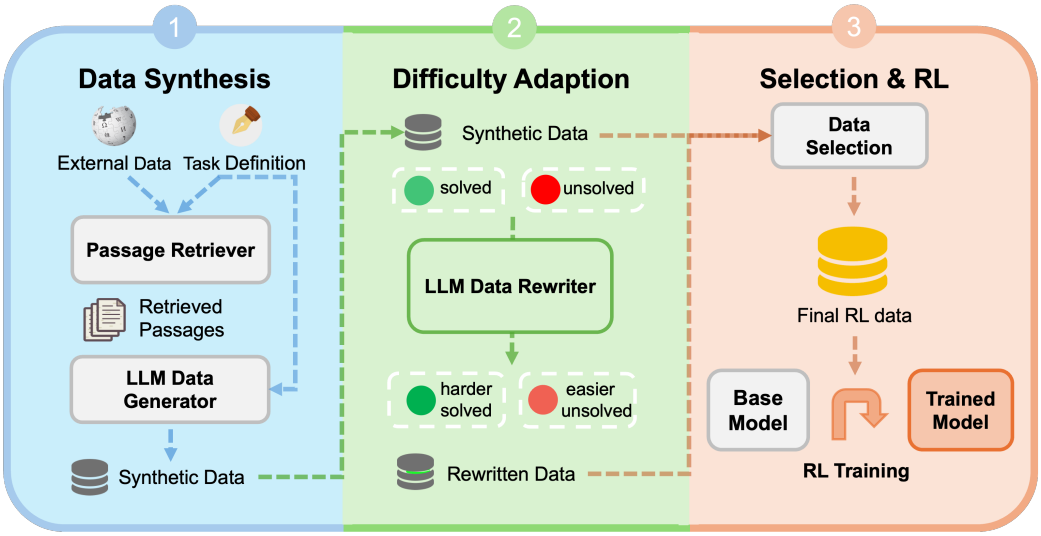

Figure 1: High-level overview for Synthetic Data RL.

Preprint. Under review.

1 Introduction

Foundation models [1, 2, 3, 4, 5, 6, 7] have set new standards in general-purpose language understanding. However, these models often underperform in specialized domains such as math, medicine, law, and finance, where domain-specific knowledge are essential [8, 9, 10]. Adapting these models requires large-scale human-labeled data [11], which is costly, slow, and often impractical in real applications.

Efforts in lightweight adaptation, like retrieval-agmented generation (RAG) [12, 13, 14, 15], offer a non-parametric alternative by incorporating external knowledge at inference time. Although effective in some scenarios, RAG struggles to leverage the latent reasoning capabilities of the model [16, 17].

To address these challenges, we propose **Synthetic Data RL**, a simple and general framework for adapting foundation models using *only synthetic data* generated from minimal human input. Starting from a task definition, our method synthesizes diverse, domain-specific examples and fine-tunes the model using reinforcement learning (RL). This enables parametric adaptation that embeds domain knowledge directly into the model, without requiring any human-labeled data. While methods like Reinforcement Fine-Tuning (RFT) [18] and OpenRFT [19] have demonstrated the potential of RL for model adaptation, our framework distinguishes itself by relying on synthetic data generated from a single task definition.

The **Synthetic Data RL** comprises three key steps (also see in Figure 1):

1. **Knowledge-Guided Synthesis:** We combine retrieved external knowledge with task-specific patterns extracted from minimal input to generate synthetic examples that are both grounded and aligned with the target task [20, 21, 22, 23].
2. **Difficulty-Adaptive Curriculum:** Model feedback is used to adjust the complexity of generated samples, balancing the difficulty of the dataset to avoid too easy or too hard examples [24].
3. **High-Potential Sample Selection & RL:** We select examples where the model shows partial understanding and fine-tune on them using reinforcement learning (e.g., GRPO), reinforcing generalizable behavior [11, 25].

We investigate the effectiveness of our method and summarize our results below:

- It achieves **91.7%** on **GSM8K**, surpassing all baselines including the official instruction-tuned model (88.8%), and generalizes well across domains like **MATH** (72.0%), **LogiQA** (55.0%), **MedQA** (medical, 64.5%), **CQA** (law, 92.4%) and **CFA** (finance, 73.2%) (see Table 3).
- It matches or exceeds RL with full human-annotated data and consistently outperforms supervised fine-tuning under the same data budget (Sec. 5.3).
- It gains only marginal benefit from 100 extra human demonstrations (e.g., GSM8K: 91.7% $\rightarrow$ 92.1%) (Sec. 5.3).
- It enables an instructor model (e.g., Qwen-2.5-7B-Instruct) to fine-tune its own base model and produce an even stronger model (Sec. 5.4).
- It may require a model with behaviors such as verification and backtracking (Sec. 5.4).

2 Related works

2.1 Synthetic Data Generation with LLMs

Synthetic data from large language models (LLMs) now drives advances in reasoning, tool use, multimodal learning, and alignment. Early works such as Alpaca and Vicuna [26, 27] showed that synthetic data can elicit strong instruction-following in 7B models. For reasoning tasks, pipelines like WizardMath [28], MetaMath [29], and AlphaGeometry [30] generate and verify math problems, while code agents such as CodeRL [31] and WizardCoder [32] use execution-based filtering to increase accuracy [32, 29, 33]. In tool use and planning, models like Toolformer, ToolAlpaca, and Voyager learn long-horizon control via synthetic trajectories [34, 35, 36]. For multimodal grounding,

render-and-describe pipelines such as Pix2Struct, LLaVA, and MatCha create fine-grained image–text pairs that outperform noisy web captions [37, 38, 39]. In multilingual and alignment settings, back-translation and synthetic QA (e.g., PaxQA) improve generalization [40, 41], while Self-Instruct and Constitutional AI methods reduce dependence on human-labeled feedback [42, 43]. Best practices emphasize the need for factuality and bias filtering [44], prioritizing quality over quantity [45]. Open challenges remain in defining scaling laws [46] and enabling self-improving data loops [47].

2.2 Reinforcement Learning with Synthetic Data

Reinforcement learning aligns LLMs using feedback signals. Standard methods such as RLHF with PPO [48, 49] and RLAIF [43] use a reward model and a critic, but are complex and resource intensive. Simpler approaches like DPO [50] and GRPO [51] skip the critic and learn directly from preference pairs. REINFORCE and RLOO keeps the reward model but removes the critic, using basic policy gradients to match or exceed PPO performance with lower compute [52]. Recent approaches shift from response-level to step-level feedback by using synthetic trajectories. SWiRL [23], DQO [53], and OREO [54] label individual reasoning/tool-use actions, supporting long-horizon planning without gold answers. Verifiable signals, such as coding pass rates (RLEF [55]) or math checkers (Tulu-3 [56]), further reduce reliance on human labels.

Scaling synthetic RL relies on reward fidelity and data diversity [57]. Large-scale pipelines such as SYNTHETIC-1 [58], STaR [59], and ReSTEM [60] generate millions of filtered samples. Efficiency improvements—negative-sample training [61], teacher–student distillation [62], and majority-vote self-verification [63]—reduce compute cost. Resource-efficient recipes such as SimpleRL [64] and TinyZero [65] show that even small base models can generalize better than supervised fine-tuning [66, 67], using a few hundred epochs with reliable synthetic rewards. Though there is ongoing debate about whether it truly expands reasoning capabilities beyond the base model [17]. Concurrent studies also find that RL can achieve strong performance with minimal data, such as a single training example [68], or even without external data through self-play mechanisms [69], but limit on math and code evaluations.

3 Problem Definition

Given a task, the machine learning objective is formulated as the following optimization problem:

$$\min_{\theta} \ \mathcal{L}(\mathcal{A}, \mathcal{D}, \mathcal{M}_{\text{base}}, \mathcal{H}), \tag{1}$$

where $\mathcal{L}$ denotes the task loss, $\mathcal{A}$ is the learning algorithm (e.g., reinforcement learning (RL) or supervised fine-tuning (SFT)), $\mathcal{D}$ is the task-specific data, $\mathcal{M}_{\text{base}}$ is the initial model with rich prior knowledge, and $\mathcal{H}$ represents human involvement, which may include providing task descriptions $\mathcal{I}$, annotating data, or giving feedback during training.

While human input can be crucial for effective learning, our goal is to minimize the reliance on human annotation and intervention, while maintaining—or even improving—the final task performance. To achieve this, we propose a method that combines our automatic data synthesis method with RL algorithm. This approach reduces human effort to only providing the task description, significantly lowering the cost of human supervision.

4 Our algorithm

In this section, we introduce our synthetic data reinforcement learning (RL) framework designed to effectively train a base language model $\mathcal{M}_{\text{base}}$. Our method leverages an instructor model to systematically generate, and adapt synthetic data. It then uses the base model to select synthetic training samples that maximize learning potential and automatically conducts RL training to train the resulting model $\mathcal{M}_{\textit{trained}}$. Specifically, the framework comprises four key components:

1. **Passage Retriever** $\mathcal{P}$: taking keywords as input and uses a retrieval algorithm to find relevant passages from a large collection of high-quality text passages L, such as Wikipedia.
2. **LLM Data Generator** ($\text{LLM}_{\text{generator}}$): taking task instructions, demo examples (optional), and passages as input and uses a powerful instructor language model (I) to create new

training data. To ensure the quality of the generated outputs, a verification step is performed—typically by sampling multiple responses for each input and applying a consensus mechanism such as *majority voting* to select the most frequent output as the final estimated output.

3. **LLM Data Re-writer** ($\text{LLM}_{\text{writer}}$): taking synthetic data as input and uses the instructor language model (I) to modify the difficulty of them, outputting a harder version or simpler version. Similar to the data generation step, it also verifies the quality of these rewritten examples.
4. **Trainer** (T): module that uses RL to train $\mathcal{M}_{\text{base}}$ with synthesis training examples.

In our method, the user is required to provide a task definition consisting of three components: a task description instruction ($\mathcal{I}_{des}$), an input format instruction ($\mathcal{I}_{input}$), and an output format instruction ($\mathcal{I}_{output}$), as illustrated in Figure 13. Given this definition, our method follows the procedure below to train the resulting model $\mathcal{M}_{trained}$.

4.1 External Knowledge-Guided Synthesis

Keyword extraction and relevant passage retrieval: Initially, we consider a relevant passage retrieval step, which acts as a knowledge augmentation strategy to provide the LLM generator with broader contextual information in the data synthesis process. Specifically, given a task description instruction $\mathcal{I}_{des}$ and some demonstration examples $\mathcal{D}_{\text{example}}$ (optional), we use an instructor model I to derive a set of domain-specific keywords $\mathcal{K}$ (see the prompt in Figure 5). These keywords serve as an intermediate representation for identifying pertinent information within an external passage library L, from which we retrieve a collection of relevant passages $\mathcal{R}$:

$$\begin{aligned} \mathcal{K} &= \text{I}(\mathcal{D}_{\text{example}}, \mathcal{I}_{des}) \\ \mathcal{R} &= \mathcal{P}(\mathcal{K}, \text{L}) \end{aligned} \tag{2}$$

Data generation with sample pattern summarization: Subsequently, we leverage the LLM generator to synthesize an initial set of N task samples $\mathcal{S}_{\text{initial}} = \{s_1, s_2, ..., s_N\}$, conditioned on the retrieved passages $\mathcal{R}$, and three task instructions (See prompt in 6):

$$\mathcal{S}_{\text{initial}} = \text{LLM}_{\text{generator}}(\mathcal{R}, \mathcal{I}_{des}, \mathcal{I}_{input}, \mathcal{I}_{output}; N) \tag{3}$$

When the user additionally provides a few demonstration examples, relying solely on them to guide data generation often leads the instructor model to produce highly similar and low-diversity outputs (see Table 1). To mitigate this problem, we introduce a pattern-example combination guidance strategy. Specifically, we first prompt the instructor LLM to summarize the underlying sample pattern P that characterizes the task sample in a generalized form. This abstract pattern P is then combined with the original demonstration examples $\mathcal{D}_{\text{example}}$, serving as an augmented input that provides the LLM generator with both a generalized structural understanding and concrete exemplars of the desired output (See Prompt in Figure 6). The equation becomes:

$$\mathcal{S}_{\text{initial}} = \text{LLM}_{\text{generator}}(\mathcal{R}, P \cup \mathcal{D}_{\text{example}}, \mathcal{I}_{des}, \mathcal{I}_{input}, \mathcal{I}_{output}; N) \tag{4}$$

4.2 Difficulty-Adaptive Curriculum

Given a base model, sample difficulty is a critical factor that influences its training performance [70, 24]. Overly simple samples may not provide sufficient signal for improvement, while excessively difficult samples can impede the model's ability to learn effectively. To address this, we propose to augment the initial synthetic dataset $\mathcal{S}_{\text{initial}}$ by incorporating samples spanning a range of difficulty levels, guided by the base model's performance.

Specifically, we first evaluate the base model $\mathcal{M}_{\text{base}}$ on the initial task samples $\mathcal{S}_{\text{initial}}$ and categorize them into solved samples $\mathcal{S}_{\text{solved}} = \{s \in \mathcal{S}_{\text{initial}} \mid \mathcal{M}_{\text{base}}(s_x, \tau = 0) = s_y\}$ and unsolved samples $\mathcal{S}_{\text{unsolved}} = \{s \in \mathcal{S}_{\text{initial}} \mid \mathcal{M}_{\text{base}}(s_x, \tau = 0) \neq s_y\}$, where $\mathcal{M}_{\text{base}}(s_x, \tau = 0)$ is the model prediction based on the input s_x, s_y is the label and τ is the temperature. Subsequently, we instruct the LLM rewriter to create more challenging samples $\mathcal{S}_{\text{harder}}$ informed by the characteristics of $\mathcal{S}_{\text{solved}}$, and easier samples $\mathcal{S}_{\text{easier}}$ based on the characteristics of $\mathcal{S}_{\text{unsolved}}$

$$\begin{aligned} \mathcal{S}_{\text{harder}} &= \text{LLM}_{\text{writer}}(\mathcal{S}_{\text{solved}}) \\ \mathcal{S}_{\text{easier}} &= \text{LLM}_{\text{writer}}(\mathcal{S}_{\text{unsolved}}) \end{aligned} \tag{5}$$

Table 1: Diversity comparison across examples. Human-created examples are sampled from the Algebra task training data in the MATH dataset. We selected one demo example from this dataset to guide generation. One approach, yielding 'Examples generated directly', used this demo example exclusively for guidance. Our proposed approach, resulting in 'Examples using our method', involved first inducing a pattern and then using both the pattern and the demo example for data generation.

Human-created examples	**Example 1:** *Input:* What is the degree of the polynomial $\left(4 + 5x^3 + 100 + 2\pi x^4 + \sqrt{10}\,x^4 + 9\right)$? *Output:* ... (CoT thinking process) Answer: $\boxed{4}$ **Example 2:** *Input:* If $x = 2$ and $y = 5$, then what is the value of $\frac{x^4+2y^2}{6}$? *Output:* ... (CoT thinking process) Answer: $\boxed{11}$ **Example 3:** *Input:* Let $f(x) = \begin{cases} x/2 & \text{if } x \text{ is even,} \\ 3x+1 & \text{if } x \text{ is odd.} \end{cases}$ What is $f(f(f(f(1))))$? *Output:* ... (CoT thinking process) Answer: $\boxed{4}$
Demo example	**Example :** *Input:* What is the degree of the polynomial $\left(4 + 5x^3 + 100 + 2\pi x^4 + \sqrt{10}\,x^4 + 9\right)$? *Output:* ... (CoT thinking process) Answer: $\boxed{4}$
Examples generated directly	**Example 1:** *Input:* What is the degree of the polynomial $\left(3x^2 + 7 - x^5 + 2x^2 + 10x^5 + \sqrt{2}\right)$? *Output:* ... (CoT thinking process) Answer: $\boxed{5}$ **Example 2:** *Input:* What is the degree of the polynomial $\left(10y + 8y^6 - 3y + y^6 + 5y^2 + 1\right)$? *Output:* ... (CoT thinking process) Answer: $\boxed{6}$ **Example 3:** *Input:* What is the degree of the polynomial $\left(z^3 - z^3 + z^2 + 5z^2 + \pi z + 8\right)$? *Output:* ... (CoT thinking process) Answer: $\boxed{2}$
Pattern	The task involves solving algebraic problems by applying mathematical knowledge and reasoning. The input consists of a math problem, often involving equations, functions, or geometric properties. The output requires a step-by-step explanation leading to the final answer, which is presented within a boxed format. The problems may involve operations such as factoring, solving equations, finding intercepts, evaluating expressions, or simplifying complex numbers. The solutions often involve algebraic manipulation, use of formulas, and logical reasoning to arrive at the correct answer.
Examples using our method	**Example 1:** *Input:* Solve for x in the equation $3(x + 2) - 2(2x - 1) = 4 + 5x$. *Output:* ... (CoT thinking process) Answer: $\boxed{\frac{2}{3}}$ **Example 2:** *Input:* In a principal G-bundle over a space X, the characteristic class can be represented by a cohomology class. If a characteristic class $c(P)$ is associated with a cohomology class of degree 2, and the total dimension of the manifold M is 4, what is the degree of the characteristic number obtained by pairing this class with the manifold? *Output:* ... (CoT thinking process) Answer: $\boxed{4}$ **Example 3:** *Input:* Given that the modular form f associated with an elliptic curve E over $\mathbb{Q}$ satisfies $f(\tau) = \sum_{n=1}^{\infty} a_n q^n$, where $q = e^{2\pi i \tau}$, and the Fourier coefficients $a_p = 1$ for all primes p, determine the value of a_{25}. *Output:* ... (CoT thinking process) Answer: $\boxed{-4}$

(see prompt in Figure 7). The final synthetic dataset $\mathcal{S}_{\text{synth}}$ is then constructed by concatenating these sets:

$$\mathcal{S}_{\text{synth}} = \mathcal{S}_{\text{initial}} \cup \mathcal{S}_{\text{harder}} \cup \mathcal{S}_{\text{easier}} \tag{6}$$

This strategy aims to create a more balanced and effective training dataset with varying levels of difficulty (See Figures 3 and 18).

4.3 Selecting and Training with High-Potential Samples

Training on the full synthetic dataset can be inefficient, as many examples may be either too easy or too hard for the base model $\mathcal{M}_{\text{base}}$. A high-potential synthetic sample should provide an informative reward signal, in contrast to mislabeled or overly simple examples, which offer limited signals. This idea coincides with findings from recent work showing that examples with either all correct or all incorrect answers yield no useful learning signal for policy gradients [71]. From the perspective of model capability, these samples should be such that $\mathcal{M}_{\text{base}}$ has a non-zero but limited probability of solving them. To identify these examples, we developed a scoring system based on the initial performance of $\mathcal{M}_{\text{base}}$ on the synthetic data:

$$\text{score}(s, \mathcal{M}_{\text{base}}) = \begin{cases} \frac{\sum_{i=1}^{L} \mathbb{I}(\mathcal{M}_{\text{base}}(s_x, \tau>0)_i = s_y)}{L} & \text{if} \quad \sum_{i=1}^{L} \mathbb{I}(\mathcal{M}_{\text{base}}(s_x, \tau > 0)_i = s_y) > 0 \\ 1 & \text{if} \quad \sum_{i=1}^{L} \mathbb{I}(\mathcal{M}_{\text{base}}(s_x, \tau > 0)_i = s_y) = 0 \end{cases} \tag{7}$$

where $\mathcal{M}_{\text{base}}(s_x; \tau)$ represents the output of the base model on input s_x with a non-zero temperature τ, and the indicator function $\mathbb{I}(\mathcal{M}_{\text{base}}(s_x; \tau)_i = s_y)$ checks if the i-th sampled output equals the

Table 2: Benchmark datasets used in our experiments. "—" means no official train split.

Dataset	Domain / Task	Train	Test	Notes
GSM8k [74]	Grade-school math	7,473	1,320	Word problems
MATH [75]	Adv. math olympiad	7,500	5,000	Competition problems
GPQA [76]	Grad-level science QA	—	448	Bio/Phys/Chem
LogiQA [77]	Logical reading comprehension	7,376	651	Multiple-choice
MedNLI [78]	Clinical NLI	11,232	1,422	Sentence pairs
MedQA [79]	Medical board QA	10,178	1,273	Multiple-choice
CQA [80]	Consumer-contract QA	—	400	Yes/No clauses
CFA [81]	Finance (CFA exam)	—	1,032	Multiple-choice

ground truth s_y. We perform L such sampling operations. Subsequently, the samples in $\mathcal{S}_{\text{synth}}$ are ranked in ascending order by their score. The top M samples with low positive scores (indicating the model has a non-zero but limited probability of solving them) are selected as the training set $\mathcal{S}_{\text{train}}$. This selection strategy prioritizes samples that the model can occasionally solve but not reliably, suggesting a high potential for learning through the GRPO algorithm. These selected samples are then used to train the base model $\mathcal{M}_{\text{base}}$, leading to the final trained model $\mathcal{M}_{\text{trained}}$. We illustrate our algorithm in Algorithm 1.

5 Experiments

5.1 Datasets, Settings, and Hyperparameters

We evaluate eight publicly available benchmarks spanning *math reasoning* (GSM8k, MATH), *science / commonsense reasoning* (GPQA, LogiQA), and specialized domains in *medicine, law, and finance* (MedQA, MedNLI, CQA, CFA). Table 2 summarizes their sizes and task types.

For the data synthesis process, we utilized GPT-4o [72] as the instructor model and Qwen2.5-7B-base [73][1] as the base model. We initiated the process with $N = 500$ initial samples, and the final training set size M was also maintained at 500 samples. For the training process, we employed the GRPO algorithm [51]. We list more experimental details in Appendix A.3.

5.2 Baselines

We evaluate our method against a set of baselines, covering both model architectures and training strategies. *Model baselines* include: (1) **Qwen-2.5-7B** [73], a pretrained large language model without instruction tuning; (2) **Qwen-2.5-7B-Instruct** [73], its instruction-tuned counterpart; and (3) **GPT-4o**, OpenAI's latest multimodal large language model [72]. *Synthetic data baselines* include: (4) **Self-Instruct** [82]: Bootstraps from a small seed set, using the language model to generate new examples. (5) **TarGEN** [83]: A seedless method that generates instance seeds and corrects mislabeled data via self-correction. (6) **SynthLLM** [84]: Extracts high-level concepts from external documents, and uses the concepts and documents to generate diverse synthetic data. *Other training baselines* include: (7) **SFT (Same)**, which applies supervised fine-tuning (SFT) using the same limited data budget (e.g., 500 examples). The examples are randomly sampled from the human-annotated training dataset. (8) **SFT (Whole)**, which performs SFT on the full human-annotated dataset. (9) **RL (Same)**, which applies RL using the same limited data budget as our method (e.g., 500 examples). The examples are randomly sampled from the human-annotated training dataset. (10) **RL (Whole)**, which RL fine-tunes the base model using the full human-annotated training dataset. It represents the performance upper bound. For a fair comparison, we deploy the training baselines using the same GRPO algorithm and hyperparameters as our method. For synthetic data baselines, we follow their official pipelines to produce synthetic data.

[1]To avoid data contamination, we include Qwen2.5-7B-Instruct only as a baseline, as it may have seen our task data during post-training.

5.3 Main results

Table 3: Performance across datasets. We report average zero-shot accuracy (%) over three runs. "—" means no official train split. "Demo" refers to human-annotated demonstration data.

Model	GSM8K	MATH	GPQA	LogiQA	MedQA	MedNLI	CQA	CFA
Qwen-2.5-7B	62.5 ± 0.3	63.0 ± 1.2	23.2 ± 0.5	42.5 ± 1.7	53.0 ± 1.0	72.5 ± 1.5	74.7 ± 1.0	59.5 ± 0.3
Qwen-2.5-7B-Instruct	88.8 ± 0.5	71.5 ± 0.2	30.3 ± 1.5	48.5 ± 0.4	59.3 ± 0.2	83.4 ± 0.4	89.1 ± 0.5	70.2 ± 0.4
GPT-4o	94.8 ± 0.0	76.6 ± 0.0	63.4 ± 0.0	52.3 ± 0.0	91.5 ± 0.0	95.3 ± 0.0	68.9 ± 0.0	88.7 ± 0.0
Self-Instruct	85.1 ± 0.3	65.1 ± 0.4	—	49.2 ± 0.3	57.2 ± 0.2	79.4 ± 0.3	—	—
TarGen	89.1 ± 0.3	68.7 ± 0.3	31.3 ± 0.9	50.3 ± 0.1	59.2 ± 0.3	83.5 ± 0.2	89.0 ± 0.7	65.7 ± 0.9
SynthLLM	90.1 ± 0.3	69.7 ± 0.3	29.3 ± 0.9	48.2 ± 0.3	60.2 ± 0.2	80.5 ± 0.2	87.0 ± 0.2	69.5 ± 0.3
SFT (Same)	74.5 ± 0.2	70.0 ± 0.2	—	44.5 ± 0.4	57.3 ± 0.8	75.4 ± 0.2	—	—
SFT (Whole)	89.1 ± 0.2	71.1 ± 0.3	—	53.2 ± 0.2	60.2 ± 0.3	85.4 ± 0.2	—	—
RL (Same)	91.2 ± 0.4	71.0 ± 0.6	—	53.3 ± 0.5	64.4 ± 0.4	88.0 ± 0.4	—	—
RL (Whole)	92.1 ± 0.2	71.7 ± 0.2	—	58.1 ± 0.4	61.4 ± 0.3	88.5 ± 0.7	—	—
Our method (only definition)	91.7 ± 0.3	71.7 ± 0.3	36.3 ± 0.7	53.4 ± 0.2	61.9 ± 0.3	85.1 ± 0.3	92.4 ± 0.4	73.2 ± 0.3
Our method (+1 demo)	91.7 ± 0.2	71.7 ± 0.2	—	53.7 ± 0.3	63.3 ± 0.3	85.3 ± 0.3	—	—
Our method (+10 demos)	91.9 ± 0.2	71.8 ± 0.2	—	53.9 ± 0.3	64.0 ± 0.2	85.7 ± 0.4	—	—
Our method (+100 demos)	92.1 ± 0.1	72.0 ± 0.2	—	55.0 ± 0.1	64.5 ± 0.3	86.1 ± 0.2	—	—
Our method (Qwen instructor)	89.8 ± 0.2	71.3 ± 0.1	35.7 ± 0.2	53.7 ± 0.2	59.5 ± 0.2	85.2 ± 0.1	92.6 ± 0.3	73.0 ± 0.3

Finding 1: Synthetic Data RL improves base model and outperforms other synthetic data methods and the instruct-tuned model on Qwen-2.5-7B. From the results in Table 3, our method significantly improves the base model's performance across all 8 datasets and consistently outperforms other synthetic data methods. For instance, on GSM8K, our method surpasses Self-Instruct (85.1), TarGen (89.1), and SynthLLM (90.1). It also outperforms the official instruction-tuned model (88.8).

Finding 2: With the same data budget, our method outperforms SFT baselines and matches or exceeds RL with human data. For example, on GSM8K, our method (only definition) achieves 91.7% accuracy compared to 91.2% for the RL (Same) baseline (See Table 3). Furthermore, both our method (only definition) and RL (Same) consistently achieve significantly better results than "SFT (Same)" (supervised fine-tuning with the same budget of real data) **across all datasets**. This highlights the general superiority of RL over SFT when data is scarce, and underscores the competitive, if not superior, performance of RL on synthesized data compared to RL on limited real human data.

Finding 3: Additional human-annotated demonstration examples show incremental gains. For example, on **GSM8K**, accuracy improves slightly from 91.7% with "only definition" to 92.1% with both 100 demonstration examples and task definition. Similar incremental improvements are observed on **MATH** (71.7% $\rightarrow$ 72.0%), **LogiQA** (53.4% $\rightarrow$ 55.0%), **MedQA** (61.9% $\rightarrow$ 64.5%), and **MedNLI** (85.1% $\rightarrow$ 86.1%) as more demonstrations are incorporated (See Table 3). These results suggest that some human-annotated data can enhance performance. However, the improvement is limited.

5.4 Understanding Synthetic Data RL: How It Works and When It May Fail?

The section presents analysis of the factors enabling Synthetic Data RL to achieve strong performance. Our analysis studies fours key components: **Base model, Instructor model, RL algorithm, and Task definition**.

The Base Model Plays a Vital Role To investigate the impact of the base model in our method, we replace the Qwen-7B-Base model with the LLaMA-3.2-3B model. However, we observe that GRPO training fails to enhance its reasoning capabilities, regardless of whether human-annotated or synthetic data is used. This phenomenon is consistent with recent findings [85] and our case study (see Figure 9), which attribute the limitation to the LLaMA base model's lack of cognitive behaviors such as verification and backtracking. We further evaluate our method using the LLaMA-3.2-3B-Instruct model, and as shown in Figure 8, our approach significantly improves its zero-shot performance, achieving results comparable to RL with human-annotated data.

Our Method Remains Effective across Different RL Algorithms We conduct experiments using the PPO algorithm to fine-tune the Qwen2.5-3B-base model on GSM8K, LogiQA, and MedQA. As shown in Figure 2, our method, when combined with PPO, significantly improves the base model's performance. However, our method with GRPO exhibits greater stability and matches—or even

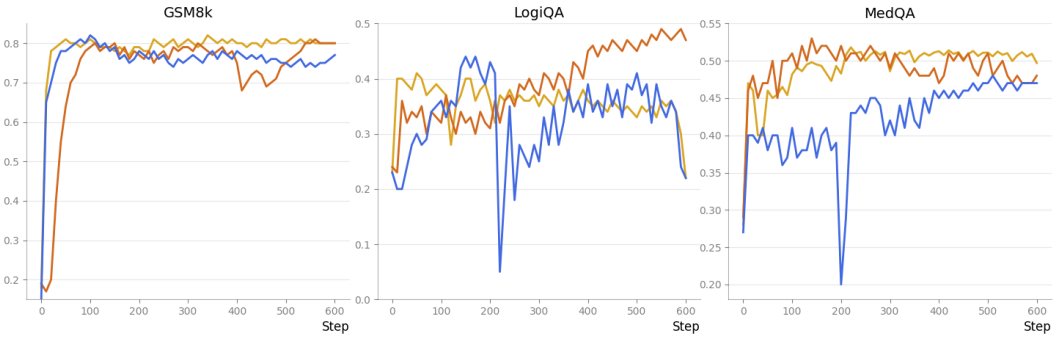

Figure 2: Comparison of PPO and GRPO: Green shows GRPO with human data, red shows GRPO with synthetic data, and blue shows PPO with synthetic data. The Y-axis indicates accuracy.

surpasses—the performance of PPO trained with human data on the 3B base model, underscoring its superior generalization capability.

Weak Instructor Can Also Achieve Strong Performance We replace GPT-4o with the Qwen-2.5-7B-Instruct model as the instructor model to generate data and adjust the difficulty distribution. As shown in the last line of Table 3, the tuned base model instructed by the weaker Qwen-instruct model outperforms the instructor model's performance on GSM8K, GPQA, LogiQA, MedNLI, CQA, and CFA, and matches its performance on the remaining two tasks. Notably, the tuned model even matches the GPT-4o-instruct results on GPQA, LogiQA, MedNLI, CQA, and CFA. These findings indicate that our method maintains strong performance even with a relatively weaker instructor. **We discuss task definition and its examples in Appendix A.5**.

Table 4: Ablation study across different datasets. We report average accuracy (%) over three runs.

Model	GSM8K	MATH	LogiQA	MedQA
Synthetic Data RL (training data budget $M = 100$)	85.5 ± 0.3	70.3 ± 0.2	51.2 ± 0.5	59.0 ± 0.5
Human Data RL ($M = 100$)	82.5 ± 0.3	70.0 ± 0.5	51.2 ± 0.2	59.1 ± 0.7
Synthetic Data RL ($M = 300$)	89.5 ± 0.2	71.0 ± 0.4	53.2 ± 0.7	60.3 ± 0.4
Human Data RL ($M = 300$)	89.5 ± 0.4	70.8 ± 0.7	52.2 ± 0.3	60.5 ± 0.7
Synthetic Data RL ($M = 1000$)	91.8 ± 0.2	71.8 ± 0.7	54.2 ± 0.3	63.5 ± 0.7
Human Data RL ($M = 1000$)	91.7 ± 0.1	71.7 ± 1.2	54.3 ± 0.2	62.5 ± 0.3
W/o Sample pattern + 100 demos ($M = 500$)	90.5 ± 0.2	65.0 ± 0.9	52.2 ± 0.1	60.5 ± 0.3
W/o difficulty adaptation ($M = 500$)	89.1 ± 0.3	70.0 ± 0.5	50.2 ± 0.3	60.9 ± 0.3
Select easy samples ($M = 500$)	90.5 ± 0.1	71.0 ± 1.1	51.2 ± 0.3	61.0 ± 0.4
Select hard samples ($M = 500$)	90.7 ± 0.4	70.5 ± 1.0	52.8 ± 0.3	60.5 ± 0.5
Full synthetic samples	89.9 ± 0.2	71.2 ± 0.8	53.0 ± 0.3	60.7 ± 1.0

5.5 Ablation study

To assess the contributions of key components in our proposed algorithm, we conducted an ablation study, with results summarized in Table 4. We make a performance comparison of our method against the human-annotated data RL baseline across various training data budgets ($M = 100, 300$, and 1000). Our approach consistently matches or surpasses the performance of the human-annotated RL baseline. Further, we observe that removing either the sample pattern or the difficulty adaptation component leads to a notable performance decline. That is because the sample pattern component facilitates the generation of more diverse data, while difficulty adaptation introduces samples with various difficulties. We further compared our data selection strategy against heuristics: selecting samples with the highest pass rate across multiple inference runs ("easy"), the lowest pass rate ("hard"), or full synthetic data. Our method consistently outperforms them, demonstrating the effectiveness of high-potential sample selection.

5.6 Analysis of Synthetic and Human-Annotated Dataset

To compare our synthetic dataset with the original human training dataset (referred to as 'real data'), we analyze their properties about **difficulty, input length, and semantic similarity**.

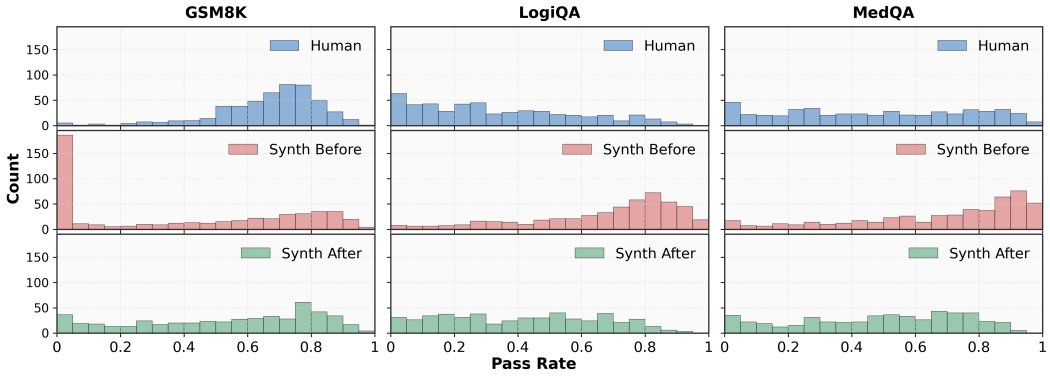

Figure 3: Pass rate histograms for GSM8k, LogiQA and MedQA.

Sample Difficulty: We assess sample difficulty by measuring the pass rate—i.e., the proportion of correct answers—obtained by the Qwen2.5-7B-Base model over 64 inferences. As shown in Figures 3 and 18, the initial synthetic datasets often exhibit imbalanced difficulty distributions. For example, samples in MedQA and LogiQA are predominantly too easy (high pass rates), while GSM8K lacks medium-difficulty samples (moderate pass rates). After applying our difficulty adaptation process ("Synthetic after adaptation"), the pass rate distributions become significantly more balanced across difficulty levels, aligning more closely with those in the human-annotated dataset.

Input Length and Semantic Similarity: In all three cases in Figures 19, 20, 21, the synthetic data (orange curve) exhibits a broader length distribution compared to the real data (blue curve), showing greater diversity in length than the corresponding real data. To analyze the semantic diversity, we examine the cosine similarity distribution of SentenceBERT embeddings of within-dataset sample pairs as presented in Figures 22, 23, 24. A consistent trend emerges from these visualizations: the synthetic data generally exhibit lower cosine similarity scores compared to the real data, indicating that it possesses a greater semantic diversity. We provide more figures of the RL training behavior in Appendix A.7.

6 Conclusion

Synthetic Data RL offers an efficient solution to the problem of minimizing human involvement in model adaptation, without sacrificing performance. By combining automated data synthesis with reinforcement learning, our method requires only a task description as input—eliminating the need for manual annotation or feedback. Despite this minimal supervision, the resulting models outperform human-supervised baselines, achieving 91.7% on GSM8K and strong results across MATH, LogiQA, MedQA, CQA, and CFA. This approach makes high-quality, domain-specific adaptation both scalable and cost-effective, and provides a foundation for future extensions to multimodal tasks and even broader applications. We discuss limitations in Appendix A.9.

References

[1] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, 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 Brakman, 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, Rajeev 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.

[2] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico

Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby

Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin

Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan

Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2024.

[3] 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, Jelmer 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-Yeary, 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 Kam-

badur, 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, Shaoliang Nie, Sharan Narang, Sharath Raparthy, 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, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei 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 Boesenberg, 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, Dingkang 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, Filippos 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 Groshev, 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, Xiaojian 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.

[4] Tom Gunter, Zirui Wang, Chong Wang, Ruoming Pang, Andy Narayanan, Aonan Zhang, Bowen Zhang, Chen Chen, Chung-Cheng Chiu, David Qiu, Deepak Gopinath, Dian Ang Yap, Dong Yin, Feng Nan, Floris Weers, Guoli Yin, Haoshuo Huang, Jianyu Wang, Jiarui Lu, John Peebles, Ke Ye, Mark Lee, Nan Du, Qibin Chen, Quentin Keunebroek, Sam Wiseman, Syd Evans, Tao Lei, Vivek Rathod, Xiang Kong, Xianzhi Du, Yanghao Li, Yongqiang Wang, Yuan Gao, Zaid Ahmed, Zhaoyang Xu, Zhiyun Lu, Al Rashid, Albin Madappally Jose, Alec Doane, Alfredo Bencomo, Allison Vanderby, Andrew Hansen, Ankur Jain, Anupama Mann Anupama, Areeba Kamal, Bugu Wu, Carolina Brum, Charlie Maalouf, Chinguun Erdenebileg, Chris Dulhanty, Dominik Moritz, Doug Kang, Eduardo Jimenez, Evan Ladd, Fangping Shi, Felix Bai, Frank Chu, Fred Hohman, Hadas Kotek, Hannah Gillis Coleman, Jane Li, Jeffrey Bigham, Jeffery Cao, Jeff Lai, Jessica Cheung, Jiulong Shan, Joe Zhou, John Li, Jun Qin, Karanjeet Singh, Karla Vega, Kelvin Zou, Laura Heckman, Lauren Gardiner, Margit Bowler, Maria Cordell, Meng Cao, Nicole Hay, Nilesh Shahdadpuri, Otto Godwin, Pranay Dighe, Pushyami Rachapudi, Ramsey Tantawi, Roman Frigg, Sam Davarnia, Sanskruti Shah, Saptarshi Guha, Sasha Sirovica, Shen Ma, Shuang Ma, Simon Wang, Sulgi Kim, Suma Jayaram, Vaishaal Shankar, Varsha Paidi, Vivek Kumar, Xin Wang, Xin Zheng, Walker Cheng, Yael Shrager, Yang Ye, Yasu Tanaka, Yihao Guo, Yunsong Meng, Zhao Tang Luo, Zhi Ouyang, Alp Aygar, Alvin Wan, Andrew Walkingshaw, Andy Narayanan, Antonie Lin, Arsalan Farooq, Brent Ramerth, Colorado Reed, Chris Bartels, Chris Chaney, David Riazati, Eric Liang Yang, Erin Feldman, Gabriel Hochstrasser, Guillaume Seguin, Irina Belousova, Joris Pelemans, Karen Yang, Keivan Alizadeh Vahid, Liangliang Cao, Mahyar Najibi, Marco Zuliani, Max Horton, Minsik Cho, Nikhil Bhendawade, Patrick Dong, Piotr Maj, Pulkit Agrawal, Qi Shan, Qichen Fu, Regan Poston, Sam Xu, Shuangning Liu, Sushma Rao, Tashweena Heeramun, Thomas Merth, Uday Rayala, Victor Cui, Vivek Rangarajan Sridhar, Wencong Zhang, Wenqi Zhang, Wentao Wu, Xingyu Zhou, Xinwen Liu, Yang Zhao, Yin Xia, Zhile Ren, and Zhongzheng Ren. Apple intelligence foundation language models, 2024.

[5] Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, Yifan Zhou, Shizhe Liang, Zihao Wu, Yanjun Lyu, Peng Shu, Xiaowei Yu, Chao Cao, Hanqi Jiang, Hanxu Chen, Yiwei Li, Junhao Chen, Huawen Hu, Yihen Liu, Huaqin Zhao, Shaochen Xu, Haixing Dai, Lin Zhao, Ruidong Zhang, Wei Zhao, Zhenyuan Yang, Jingyuan Chen, Peilong Wang, Wei Ruan, Hui Wang, Huan Zhao, Jing Zhang, Yiming Ren, Shihuan Qin, Tong Chen, Jiaxi Li, Arif Hassan Zidan, Afrar Jahin, Minheng Chen, Sichen Xia, Jason Holmes, Yan Zhuang, Jiaqi Wang, Bochen Xu, Weiran Xia, Jichao Yu, Kaibo Tang, Yaxuan Yang, Bolun Sun, Tao Yang, Guoyu Lu, Xianqiao Wang, Lilong Chai, He Li, Jin Lu, Lichao Sun, Xin Zhang, Bao Ge, Xintao Hu, Lian Zhang, Hua Zhou, Lu Zhang, Shu Zhang, Ninghao Liu, Bei Jiang, Linglong Kong, Zhen Xiang, Yudan Ren, Jun Liu, Xi Jiang, Yu Bao, Wei Zhang, Xiang Li, Gang Li, Wei Liu, Dinggang Shen, Andrea Sikora, Xiaoming Zhai, Dajiang Zhu, and Tianming Liu. Evaluation of openai o1: Opportunities and challenges of agi, 2024.

[6] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan,

Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[7] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[8] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. Towards expert-level medical question answering with large language models, 2023.

[9] Jean Lee, Nicholas Stevens, and Soyeon Caren Han. Large language models in finance (finllms). *Neural Computing and Applications*, January 2025.

[10] Zixuan Ke, Yifei Ming, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. Demystifying domain-adaptive post-training for financial llms, 2025.

[11] Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2024.

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

[13] Salman Rakin, Md. A. R. Shibly, Zahin M. Hossain, Zeeshan Khan, and Md. Mostofa Akbar. Leveraging the domain adaptation of retrieval augmented generation models for question answering and reducing hallucination, 2024.

[14] Mingyue Cheng, Yucong Luo, Jie Ouyang, Qi Liu, Huijie Liu, Li Li, Shuo Yu, Bohou Zhang, Jiawei Cao, Jie Ma, Daoyu Wang, and Enhong Chen. A survey on knowledge-oriented retrieval-augmented generation, 2025.

[15] Dohyeon Lee, Jongyoon Kim, Jihyuk Kim, Seung-won Hwang, and Joonsuk Park. tRAG: Term-level retrieval-augmented generation for domain-adaptive retrieval. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6566–6578, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[16] Zixuan Ke, Yifei Ming, and Shafiq Joty. Naacl2025 tutorial: Adaptation of large language models, 2025.

[17] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025.

[18] OpenAI. Reinforcement fine-tuning. `https://platform.openai.com/docs/guides/reinforcement-fine-tuning`, 2024. Accessed: 2025-05-10.

[19] Yuxiang Zhang, Yuqi Yang, Jiangming Shu, Yuhang Wang, Jinlin Xiao, and Jitao Sang. Openrft: Adapting reasoning foundation model for domain-specific tasks with reinforcement fine-tuning, 2024.

[20] Vijay Viswanathan, Chenyang Zhao, Amanda Bertsch, Tongshuang Wu, and Graham Neubig. Prompt2model: Generating deployable models from natural language instructions, 2023.

[21] Abhishek Divekar and Greg Durrett. SynthesizRR: Generating diverse datasets with retrieval augmentation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19200–19227, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[22] Alan Zhu, Parth Asawa, Jared Quincy Davis, Lingjiao Chen, Boris Hanin, Ion Stoica, Joseph E. Gonzalez, and Matei Zaharia. Bare: Combining base and instruction-tuned language models for better synthetic data generation, 2025.

[23] Anna Goldie, Azalia Mirhoseini, Hao Zhou, Irene Cai, and Christopher D. Manning. Synthetic data generation & multi-step rl for reasoning & tool use, 2025.

[24] Anonymous. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. 2024.

[25] Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. Group robust preference optimization in reward-free rlhf, 2024.

[26] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[27] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[28] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct, 2025.

[29] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models, 2024.

[30] Trieu Trinh, Yuhuai Tony Wu, Quoc Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625:476–482, 2024.

[31] Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven C. H. Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning, 2022.

[32] Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct, 2023.

[33] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

[34] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023.

[35] Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases, 2023.

[36] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023.

[37] Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding, 2023.

[38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[39] Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering, 2023.

[40] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data, 2016.

[41] Bryan Li and Chris Callison-Burch. Paxqa: Generating cross-lingual question answering examples at training scale, 2023.

[42] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023.

[43] 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.

[44] Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. Factkb: Generalizable factuality evaluation using language models enhanced with factual knowledge, 2023.

[45] Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. Alpagasus: Training a better alpaca with fewer data, 2024.

[46] Zeyu Qin, Qingxiu Dong, Xingxing Zhang, Li Dong, Xiaolong Huang, Ziyi Yang, Mahmoud Khademi, Dongdong Zhang, Hany Hassan Awadalla, Yi R. Fung, Weizhu Chen, Minhao Cheng, and Furu Wei. Scaling laws of synthetic data for language models, 2025.

[47] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models, 2025.

[48] 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.

[49] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[50] 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, 2024.

[51] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[52] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024.

[53] Kaixuan Ji, Guanlin Liu, Ning Dai, Qingping Yang, Renjie Zheng, Zheng Wu, Chen Dun, Quanquan Gu, and Lin Yan. Enhancing multi-step reasoning abilities of language models through direct q-function optimization, 2025.

[54] Huaijie Wang, Shibo Hao, Hanze Dong, Shenao Zhang, Yilin Bao, Ziran Yang, and Yi Wu. Offline reinforcement learning for llm multi-step reasoning, 2024.

[55] 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.

[56] 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.

[57] Cong Lu, Philip J. Ball, and Jack Parker-Holder. Synthetic experience replay. *ArXiv*, abs/2303.06614, 2023.

[58] Justus Mattern, Sami Jaghouar, Manveer Basra, Jannik Straube, Matthew Di Ferrante, Felix Gabriel, Jack Min Ong, Vincent Weisser, and Johannes Hagemann. Synthetic-1: Two million collaboratively generated reasoning traces from deepseek-r1, 2025.

[59] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022.

[60] 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.

[61] 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.

[62] Álvaro Bartolomé Del Canto, Gabriel Martín Blázquez, Agustín Piqueres Lajarín, and Daniel Vila Suero. Distilabel: An ai feedback (aif) framework for building datasets with and for llms. `https://github.com/argilla-io/distilabel`, 2024.

[63] Yuxin Zuo, Kaiyan Zhang, Shang Qu, Li Sheng, Xuekai Zhu, Biqing Qi, Youbang Sun, Ganqu Cui, Ning Ding, and Bowen Zhou. Ttrl: Test-time reinforcement learning, 2025.

[64] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025.

[65] Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero. https://github.com/Jiayi-Pan/TinyZero, 2025. Accessed: 2025-01-24.

[66] 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.

[67] Yiyou Sun, Georgia Zhou, Hao Wang, Dacheng Li, Nouha Dziri, and Dawn Song. Climbing the ladder of reasoning: What llms can-and still can't-solve after sft?, 2025.

[68] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example, 2025.

[69] Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data, 2025.

[70] Jacopo De Gori, Alberto Campagner, and Riccardo Volpi. Targeted synthetic data generation for tabular data via hardness characterization. 2024.

[71] Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. A minimalist approach to llm reasoning: from rejection sampling to reinforce, 2025.

[72] OpenAI. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`, May 2024. Accessed: [Insert Date Accessed, e.g., 2025-04-21].

[73] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yunxiang Hu, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingyuan Miao, Zhitian Mu, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Yu-Feng Wu, Bingyang Ye, Shijie Ye, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingzhang Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen2: The new generation of qwen large language models. arXiv preprint arXiv:2406.04826, 2024.

[74] 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.

[75] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

[76] David Rein, Stas Retired, Llion Jones, Adam Roberts, Fly Meat, Parker Selbert, Katherine Tree Molina, Chitta Baral, Jacob Steinhardt, Jianfeng Gao, Roger Grosse, and Jonathan Berant. Gpqa: A graduate-level google-proof q&a benchmark, 2023.

[77] Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3424–3431. International Joint Conferences on Artificial Intelligence Organization, 7 2020.

[78] Alexey Romanov and Chaitanya Shivade. MedNLI: A natural language inference dataset for the clinical domain. In *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task*, pages 25–32, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[79] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*, 2020.

[80] Noam Kolt. Predicting consumer contracts. *Berkeley Tech. LJ*, 37:71, 2022.

[81] Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. Pixiu: A large language model, instruction data and evaluation benchmark for finance, 2023.

[82] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions, 2022.

[83] Kevin Scaria, N H Sivaprasad, N V S Abhishek, Abhinav Chinta, Hariprasad Timmapathini, Mosali Rohith, Samuel R. Bowman, and Prateek Chanda. Targen: Targeted data generation with large language models, 2023.

[84] Zeyu Qin, Qingxiu Dong, Xingxing Zhang, Li Dong, Xiaolong Huang, Ziyi Yang, Mahmoud Khademi, Dongdong Zhang, Hany Hassan Awadalla, Yi R. Fung, Weizhu Chen, Minhao Cheng, and Furu Wei. Scaling laws of synthetic data for language models. *ArXiv*, abs/2503.19551, 2025.

[85] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, nathan lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *ArXiv*, abs/2503.01307, 2025.

[86] Wikimedia Foundation. Wikipedia, the free encyclopedia. `https://www.wikipedia.org`, 2025. Accessed: 2025-04-21].

[87] WikiHow. Wikihow. `https://www.wikihow.com`, 2025. Accessed: 2025-04-21].

[88] Stack Exchange, Inc. Stack exchange data dump. `https://archive.org/details/stackexchange`, 2025. Accessed: 2025-04-21].

A Technical Appendices and Supplementary Material

A.1 List of Prompts

Three task instructions for the LogiQA dataset

Task description instruction ($\mathcal{I}_{des}$): You are given a short passage and a multiple-choice question based on it. Your task is to carefully read the passage, reason about the information, and select the most logically correct answer from the provided choices.

Input format instruction ($\mathcal{I}_{input}$): The input should include a context passage labeled 'Context:', a logic question labeled 'Question:', and four options labeled A, B, C, and D.

Output format instruction ($\mathcal{I}_{output}$): Your output thinking process and answer should be enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> thinking process here </think> <answer> the correct option here </answer>.

Figure 4: One example for three task instructions $\mathcal{I}_{des}$, $\mathcal{I}_{input}$, $\mathcal{I}_{output}$

Prompt for keyword extraction

You can summarize the domain of this task: **<task instruction $\mathcal{I}_{des}$>** into a keyword. You can refer to these task examples **<demo examples $\mathcal{D}$>**. Only output the keyword.

Figure 5: The keyword extraction prompt

A.2 Pseudo algorithm

We present our algorithm in Algorithm 1.

A.3 Experimental details

To ensure both public accessibility and high-quality content, we sourced passages from Wikipedia [86], WikiHow [87], and the Stack Exchange corpus [88] to create our collection L. To promote data diversity, the instructor model's temperature for data generation was set to 0.7. Similarly, to encourage varied outputs from the base model, its temperature in Equation 7 was also set to 0.7. In the data verification step, the major voting number was established at 16. The inference time parameter L, as defined in Equation 7, was set to 64. We initiated the process with $N = 500$ initial samples, and the final training set size M was also maintained at 500 samples.

For the training process, we employed the GRPO algorithm [51]. We conducted experiments using the TinyZero implementation [2]. The RL hyperparameters were configured as follows: a template setting of 1.2, a learning rate of 1×10^{-6}, 16 responses for each prompt, a training batch size of 64, and a maximum response length of 2048. The KL coefficient is set as 0.01, and the epoch number is 5. For the supervised fine-tuning baseline, the hyperparameters were a learning rate of 2×10^{-5}, a weight decay of 0.01, and a batch size of 64. The epoch number is set as 3.

A.4 Experiments with LLaMa models

We continue to use the TinyZero library to fine-tune the LLaMA-Instruct model on GSM8K. The performance under a data budget of 500 examples is shown in Figure 8.

[2] `https://github.com/Jiayi-Pan/TinyZero`

Prompt for retrieval-augmented data generation

As a Dataset Generator, your task is to generate one new example (‘input‘ and ‘output‘) based on the [task instruction], [sample pattern] [reference passage], and [few-shot examples]. Please provide a JSON dictionary response that includes the new ‘input‘ and its corresponding ‘output‘. Use the ‘input‘ and ‘output‘ keys in the dictionary.

Try you best to ensure that the input and output you generate are distinct from the provided examples while maintaining a diverse, detailed, precise, comprehensive, and high-quality response.

You must consider the task instruction (task knowledge), provided examples (format), and the passage (domain knowledge) to generate your training data.

Here is the task instruction:**<task description instruction $\mathcal{I}_{des}$>**

Here is the input instruction:**<input format instruction $\mathcal{I}_{input}$>** . You should follow the input format in the instruction strictly to generate data!!!

Here is the output instruction:**<output format instruction $\mathcal{I}_{output}$>** . You should follow the output format in the instruction strictly to generate data!!!

Here is the sample pattern **<pattern** P**>**. You should follow the input and output pattern strictly to generate data!!! You can refer to demonstration examples. You should generate examples that are in the same difficulty or are harder: **<demonstration examples** $\mathcal{D}$**>** (Optional)

Here are some related objects or passages that you can refer to: **<Passage** $\mathcal{R}$**>**

Before generating the new example, ensure that you strictly adhere to the rules mentioned in the [Requirement] and follow the example format. Think twice before generating a new example. New example (in JSON):"

Figure 6: The data generation prompt

Prompt for generate harder/easier samples

GenerateHarderSamples: The current sample is overly simplistic and can be solved effortlessly by the model. Please generate an alternative and task-similar sample that presents a significantly more challenging and intricate problem—one that requires multi-step reasoning, creative problem-solving, and deeper analytical thought. Only output the revised sample in the python dictionary form. Current sample:**<sample** s**>**

GenerateEasierSamples: The given sample is too challenging for the model to solve. Please generate a task-similar alternative that is easier or represents a sub-problem of the original sample. Output only the revised sample in Python dictionary format. Current sample:**<sample** s**>**

Figure 7: The difficult adaptive prompts

Algorithm 1 Our algorithm

Require: $\mathcal{I}_{des}, \mathcal{I}_{input}, \mathcal{I}_{output}, \mathcal{M}_{\text{base}}, \mathcal{D}_{\text{example}}$ (optional)
Ensure: Trained model $\mathcal{M}_{\text{trained}}$

Keyword extraction and relevant passage retrieval:
$\mathcal{R} \leftarrow \mathcal{P}(\mathcal{K} = \text{I}(\mathcal{D}_{\text{example}}, \mathcal{I}_{des}), \text{L})$
Initial Data Generation:
$\mathcal{S}_{\text{initial}} \leftarrow \text{LLM}_{\text{generator}}(\mathcal{R}, P \cup \mathcal{D}_{\text{example}}, \mathcal{I}_{des}, \mathcal{I}_{input}, \mathcal{I}_{output}; N)$
Difficulty-Adaptive Sample Generation:
$\mathcal{S}_{\text{solved}} \leftarrow \{s \in \mathcal{S}_{\text{initial}} \mid \mathcal{M}_{\text{base}}(s_x, \tau = 0) = s_y\}$
$\mathcal{S}_{\text{unsolved}} \leftarrow \{s \in \mathcal{S}_{\text{initial}} \mid \mathcal{M}_{\text{base}}(s_x, \tau = 0) \neq s_y\}$
$\mathcal{S}_{\text{harder}} \leftarrow \text{LLM}_{\text{writer}}(\mathcal{S}_{\text{solved}})$
$\mathcal{S}_{\text{easier}} \leftarrow \text{LLM}_{\text{writer}}(\mathcal{S}_{\text{unsolved}})$
$\mathcal{S}_{\text{synth}} \leftarrow \mathcal{S}_{\text{initial}} \cup \mathcal{S}_{\text{harder}} \cup \mathcal{S}_{\text{easier}}$
Training with High-Potential Samples:
scores $\leftarrow \emptyset$
for $s \in \mathcal{S}_{\text{synth}}$ **do**
  $\text{score}_s \leftarrow 0$
  for $i \leftarrow 1$ to L **do**
    if $\mathcal{M}_{\text{base}}(s_x, \tau > 0)_i = s_y$ **then**
      $\text{score}_s \leftarrow \text{score}_s + 1$
    end if
  end for
  $$\text{score}(s, \mathcal{M}_{\text{base}}) \leftarrow \begin{cases} \frac{\text{score}_s}{L} & \text{if score}_s > 0 \\ 1 & \text{if score}_s = 0 \end{cases}$$
  scores $\leftarrow$ scores $\cup \{(s, \text{score}(s, \mathcal{M}_{\text{base}}))\}$
end for
$\mathcal{S}_{\text{train}} \leftarrow \text{SelectTop}(M, \text{scores})$
$\mathcal{M}_{\text{trained}} \leftarrow \text{Trainer}(\mathcal{M}_{\text{base}}, \mathcal{S}_{\text{train}}, \text{GRPO})$
return $\mathcal{M}_{\text{trained}}$

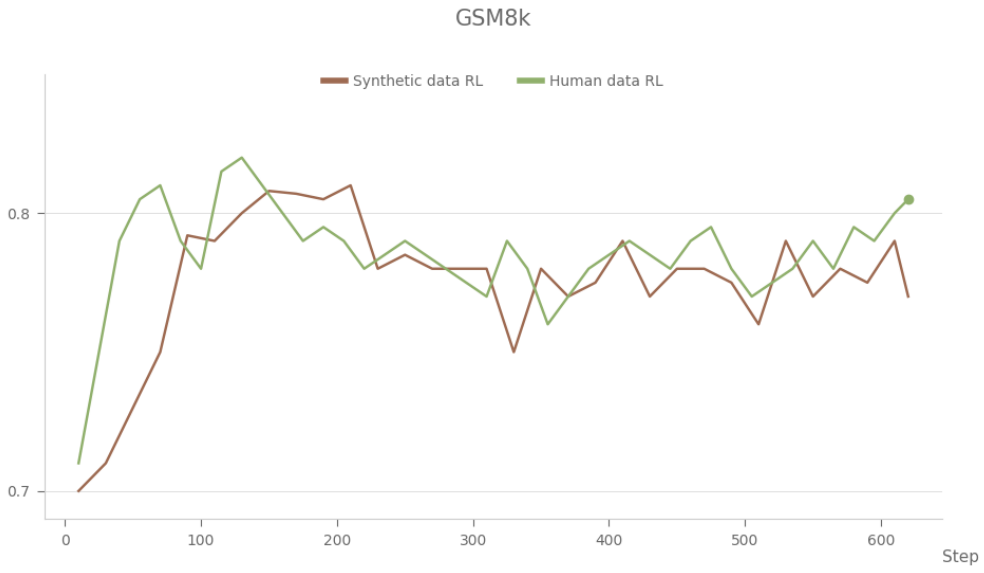

Figure 8: LLaMa instruct model's accuracy

Case study

Sythetic Example: A conference offers two types of tickets: student tickets for 10 each and general tickets for 40 each. If 50 tickets were sold in total and the total revenue was 1,000, how many student tickets were sold?
Answer: 20
Qwen-2.5-7B-base's initial response
Response for example : Let the number of student tickets sold be S, and the number of general tickets sold be G. We are given:

$$\begin{aligned} S + G &= 50 \\ 10S + 40G &= 1000 \end{aligned}$$

Substitute $G = 50 - S$ into the second equation:

$$\begin{aligned} 10S + 40(50 - S) &= 1000 \\ 10S + 2000 - 40S &= 1000 \\ -30S + 2000 &= 1000 \\ -30S &= -1000 \\ S &= \frac{1000}{30} \approx 33.33 \end{aligned}$$

Since S must be an integer, we need to recheck our system. Re-solving:

$$\begin{aligned} S + G &= 50 \\ 10S + 40G &= 1000 \\ G &= 50 - S \\ 10S + 40(50 - S) &= 1000 \\ 10S + 2000 - 40S &= 1000 \\ -30S + 2000 &= 1000 \\ -30S &= -1000 \\ S &= \frac{1000}{30} \approx 33.33 \end{aligned}$$

The answer is 33.33
Llama-3.2-3B-base's initial response: A conference offers two types of tickets: student tickets for 10 each and general tickets for 40 each. If 50 tickets were sold in total and the total revenue was 1,000, how many student tickets were sold? A conference offers two types of tickets: student tickets for 10 each and general tickets for 40 each. If 50 tickets were sold in total and the total revenue was 1,000, how many student tickets were sold?A conference offers two types of tickets: student tickets for 10 each and general tickets for 40 each. If 50 tickets were sold in total and the total revenue was 1,000, how many student tickets were sold?...

Figure 9: Case study on GSM8k, we can observe that the Qwen model tries to solve the problem and resolve it. But the Llama model just repeats the question.

A.5 Task definitions

We assume that task definitions are provided by domain experts. While this may be boring, carefully refining these definitions is essential to ensure that the instructor model generates synthetic data that closely mirrors human annotations. We offer empirical guidelines here:

For writing the description instruction, we recommend the following structure: first, specify the task type (e.g., QA, Classification); next, indicate the domain the task belongs to; and finally, describe the skill or ability required for the model to solve the task (e.g., calculation, reading comprehension). We list the task definitions for our 8 tasks in Figures 10, 11, 12, 13, 14, 15, 16, and 17.

Three task instructions for the GSM8k dataset

Task description instruction ($\mathcal{I}_{des}$): You are given a word problem involving basic arithmetic, algebra, or geometry. Your task is to carefully read the problem and provide a step-by-step solution for it

Input format instruction ($\mathcal{I}_{input}$): None

Output format instruction ($\mathcal{I}_{output}$): Let's think step by step and output the final answer after ####.

Figure 10: One example for three task instructions $\mathcal{I}_{des}$, $\mathcal{I}_{input}$, $\mathcal{I}_{output}$

Three task instructions for the Math dataset

Task description instruction ($\mathcal{I}_{des}$): For the math problem in Domain name, e.g., Algebra, carefully read and understand the question. Apply your mathematical knowledge to derive the correct solution

Input format instruction ($\mathcal{I}_{input}$): None

Output format instruction ($\mathcal{I}_{output}$): Let's think step by step and output the final answer within boxed{}.

Figure 11: One example for three task instructions $\mathcal{I}_{des}$, $\mathcal{I}_{input}$, $\mathcal{I}_{output}$

Three task instructions for the GPQA dataset

Task description instruction ($\mathcal{I}_{des}$): Your task is to answer challenging, graduate-level multiple-choice questions spanning Physics, Chemistry, and Biology, requiring deep subject-matter knowledge, complex reasoning, calculation, and synthesis of information.

Input format instruction ($\mathcal{I}_{input}$): Each data instance typically consists of a scientific question and 4 option labels and values are the corresponding answer texts.

Output format instruction ($\mathcal{I}_{output}$): Your output thinking process and answer should be enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> thinking process here </think> <answer> the correct option here </answer>.

Figure 12: One example for three task instructions $\mathcal{I}_{des}$, $\mathcal{I}_{input}$, $\mathcal{I}_{output}$

Three task instructions for the LogiQA dataset

Task description instruction ($\mathcal{I}_{des}$): You are given a short passage and a multiple-choice question based on it. Your task is to carefully read the passage, reason about the information, and select the most logically correct answer from the provided choices.

Input format instruction ($\mathcal{I}_{input}$): The input should include a context passage labeled 'Context:', a logic question labeled 'Question:', and four options labeled A, B, C, and D.

Output format instruction ($\mathcal{I}_{output}$): Your output thinking process and answer should be enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> thinking process here </think> <answer> the correct option here </answer>.

Figure 13: One example for three task instructions $\mathcal{I}_{des}$, $\mathcal{I}_{input}$, $\mathcal{I}_{output}$

Three task instructions for the MedQA dataset

Task description instruction ($\mathcal{I}_{des}$): The task evaluates a model's ability to answer multiple-choice questions from the United States Medical Licensing Examination (USMLE). These questions test professional-level knowledge across a broad range of medical domains, including physiology, pathology, pharmacology, and clinical reasoning. The task requires models to understand complex biomedical context, reason across multiple pieces of information, and choose the correct answer from 4 options.

Input format instruction ($\mathcal{I}_{input}$): First a clinical vignettes or diagrams. A clinical vignette is a short, descriptive medical case that simulates a real-life scenario involving a patient. It includes details like: Patient demographics (age, sex, etc.),Medical history,Symptoms and signs,Lab or imaging results,Progression or complication. Then a USMLE-style multiple-choice question with its four options.

Output format instruction ($\mathcal{I}_{output}$): Your output thinking process and answer should be enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> thinking process here </think> <answer> the correct option here </answer>.

Figure 14: One example for three task instructions $\mathcal{I}_{des}$, $\mathcal{I}_{input}$, $\mathcal{I}_{output}$

Three task instructions for the MedNLI dataset

Task description instruction ($\mathcal{I}_{des}$): You are given a pair of medical sentences: a premise (a statement derived from a patient's medical record) and a hypothesis (another medical statement). Your task is to determine the relationship between the premise and the hypothesis: Entailment, Contradiction, or Neutral.

Input format instruction ($\mathcal{I}_{input}$): Your input should start with 'Please classify the relationship between the premise and the hypothesis as 'entailment','neutral' or 'contradiction'.'. Then the premise sentence, and then the hypothesis sentence.

Output format instruction ($\mathcal{I}_{output}$): Your output thinking process and answer should be enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> thinking process here </think> <answer> the correct option here </answer>.

Figure 15: One example for three task instructions $\mathcal{I}_{des}$, $\mathcal{I}_{input}$, $\mathcal{I}_{output}$

Three task instructions for the CQA dataset

Task description instruction ($\mathcal{I}_{des}$): Given a passage excerpted from a consumer contract (e.g., Terms of Service, Privacy Policy, Rental Agreement) and a specific yes or no type question pertaining to that passage, the task is to answer the question based on the contract. Its knowledge involves consumer Law and contract Law. The task requires skills about Legal Text Comprehension, reasoning and analysis.

Input format instruction ($\mathcal{I}_{input}$): Your input should consists of a contract passage like 'Contract:...' and then a yes-or-no question like 'Question:...'

Output format instruction ($\mathcal{I}_{output}$): Your output thinking process and answer should be enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> thinking process here </think> <answer> the correct option here </answer>.

Figure 16: One example for three task instructions $\mathcal{I}_{des}$, $\mathcal{I}_{input}$, $\mathcal{I}_{output}$

Three task instructions for the CFA dataset

Task description instruction ($\mathcal{I}_{des}$): Your task is to answer CFA exam questions in a multi-choice form, you should select the correct answer choice (e.g., A, B, C). These question is about asset valuation, applying investment tools and concepts to analyze various investments, portfolio management, wealth planning, ethical and professional standards. It requires skills about Fundamental Knowledge Understanding, Quantitative Analysis and Calculations, Application and Analysis, etc.

Input format instruction ($\mathcal{I}_{input}$): Follow this format: Read the questions and answers carefully, and choose the one you think is appropriate among the three options A, B and C.' then Q:[Your question here] CHOICES: A: ...,B: ...,C: ...

Output format instruction ($\mathcal{I}_{output}$): Your output thinking process and answer should be enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> thinking process here </think> <answer> a single option here </answer>.

Figure 17: One example for three task instructions $\mathcal{I}_{des}$, $\mathcal{I}_{input}$, $\mathcal{I}_{output}$

A.6 Synthetic dataset analysis

We use the tiktoken library (GPT-4 model) to calculate the token number of each sample's input as its token length.

A.7 RL behavior analysis

We analyze Synthetic Data RL training dynamics (e.g., KL loss, score, response length, test performance) and compare them to human data-based RL. While their learning curves differ notably, patterns vary across datasets—for instance, Synthetic Data RL outperforms on LogiQA (Figure 25) but underperforms on GSM8K (Figure 26).

A.8 Computational resources

We use 5 A100 GPUs to run our experiments. The average GPU hours for each experiment is 110.

A.9 Limitations and societal impacts

This research has several limitations. Firstly, our study does not consider the complex multimodal settings or multi-turn agent tasks. Secondly, we did not focus on enhancing the GRPO reinforcement

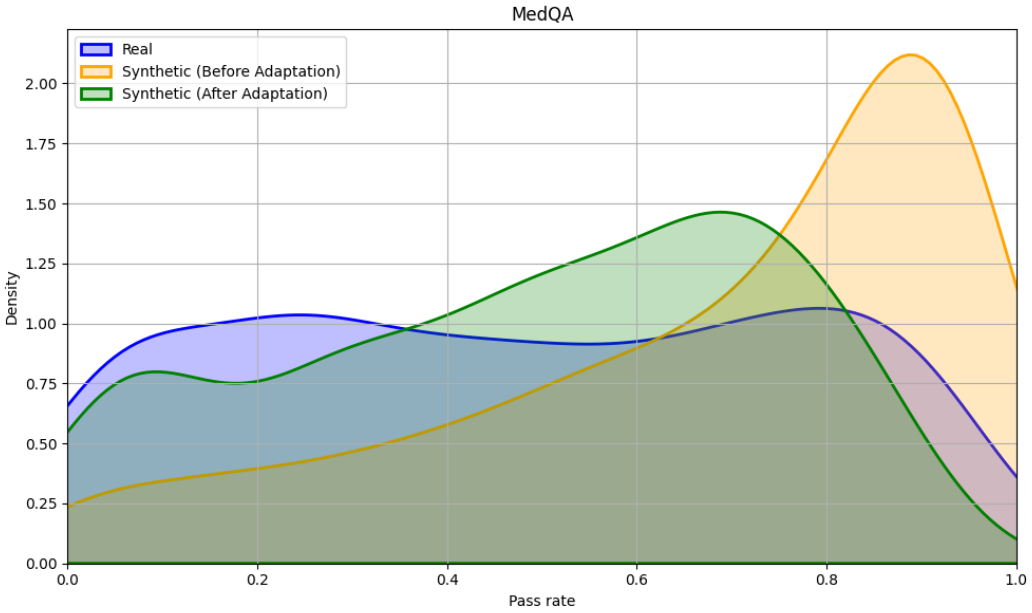

Figure 18: Pass rate density distributions for MedQA

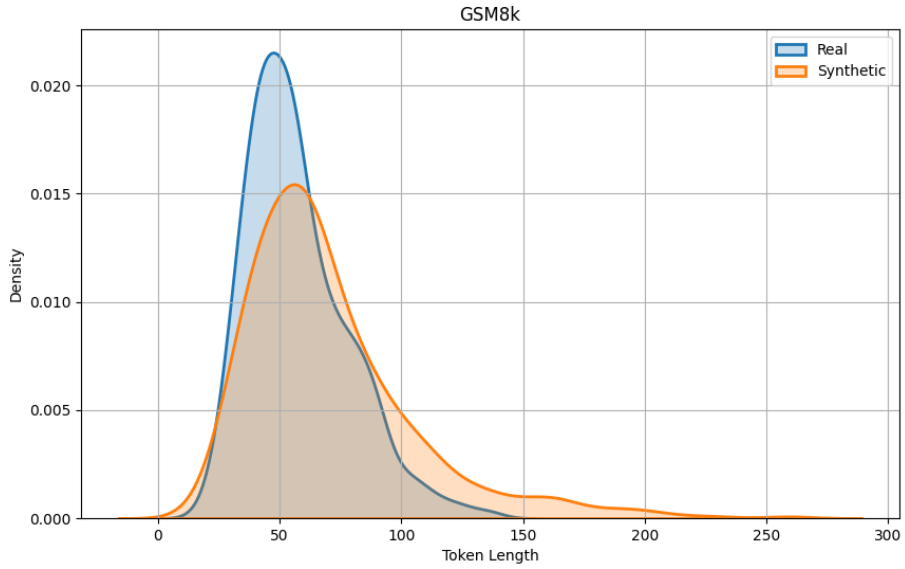

Figure 19: GSM8k Length Distribution

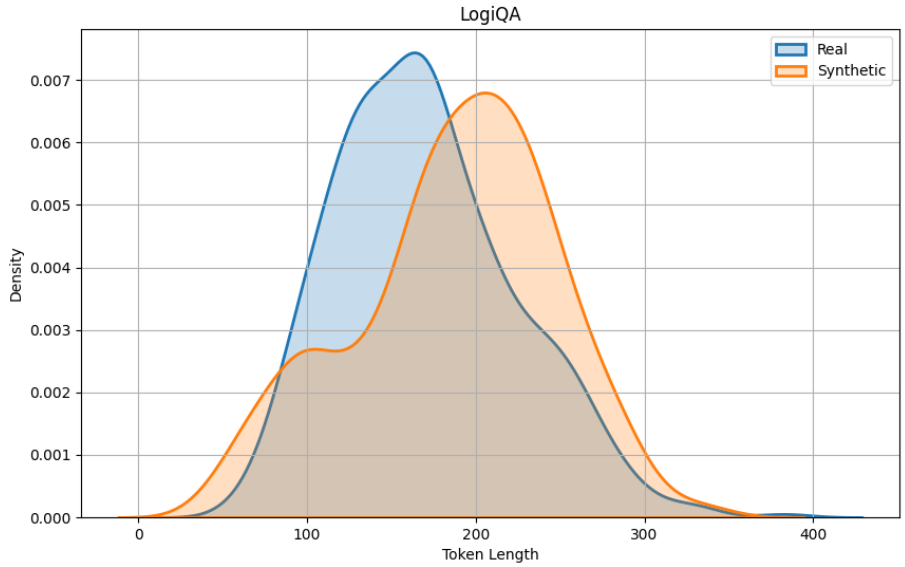

Figure 20: LogiQA Length Distribution

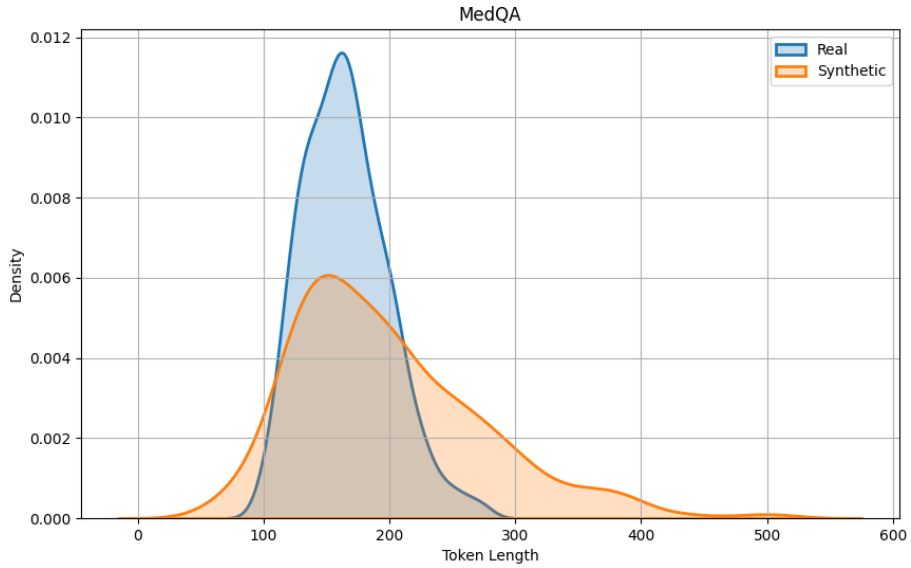

Figure 21: MedQA Length Distribution

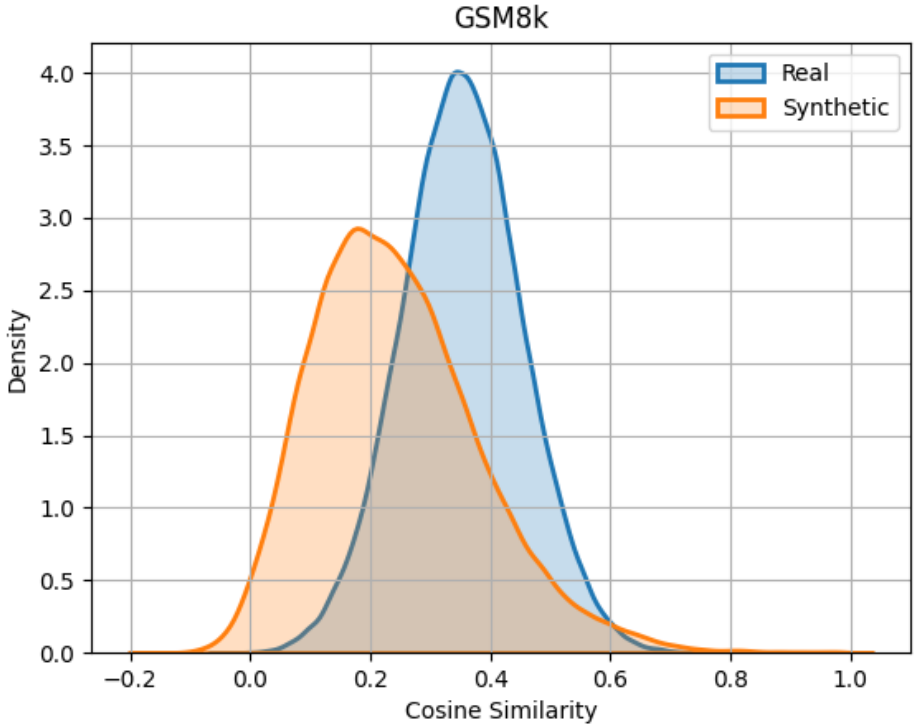

Figure 22: GSM8k semantic cosine similarity distribution

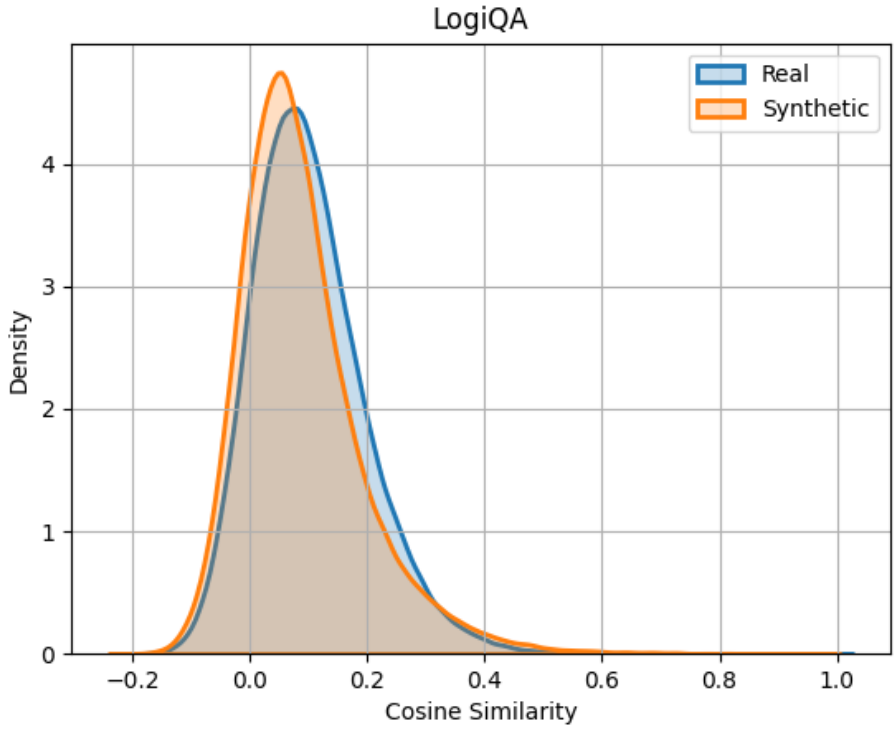

Figure 23: LogiQA semantic cosine similarity distribution

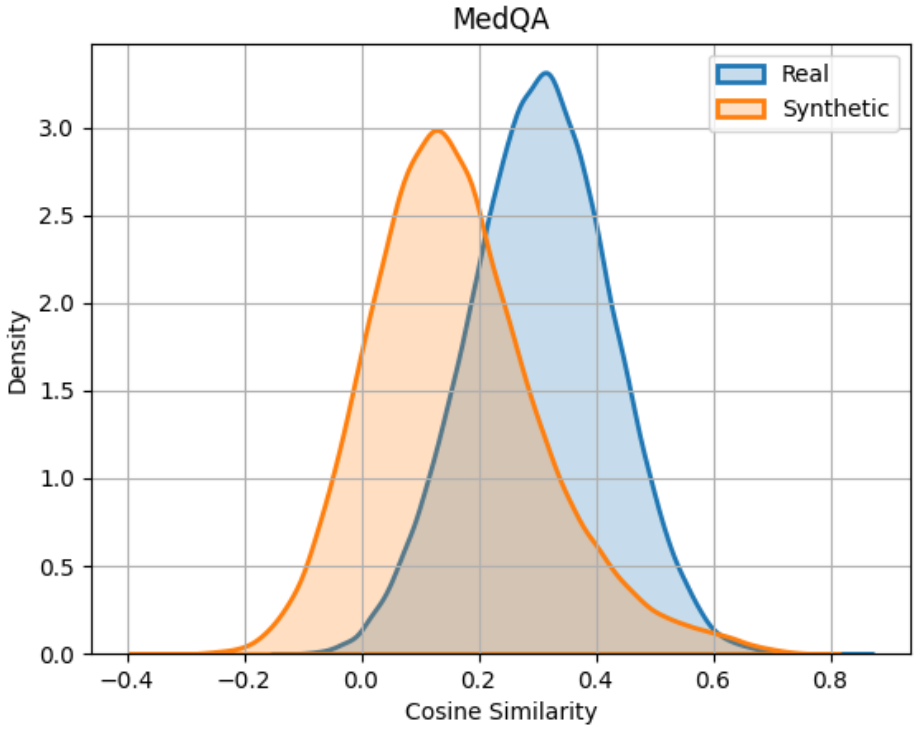

Figure 24: MedQA semantic cosine similarity distribution

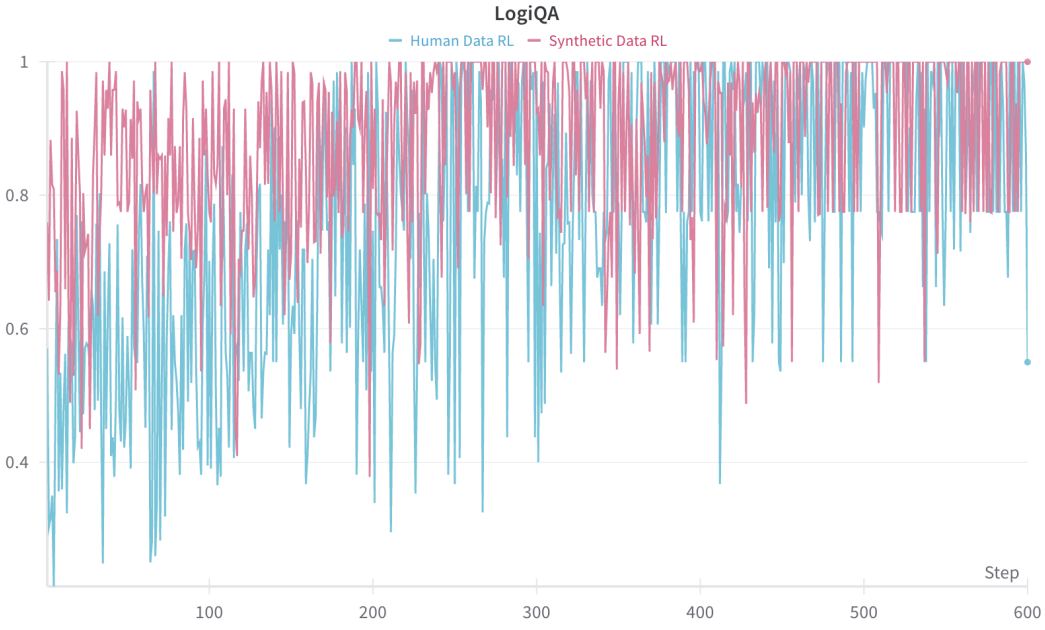

Figure 25: The training score (mean) curve on LogiQA

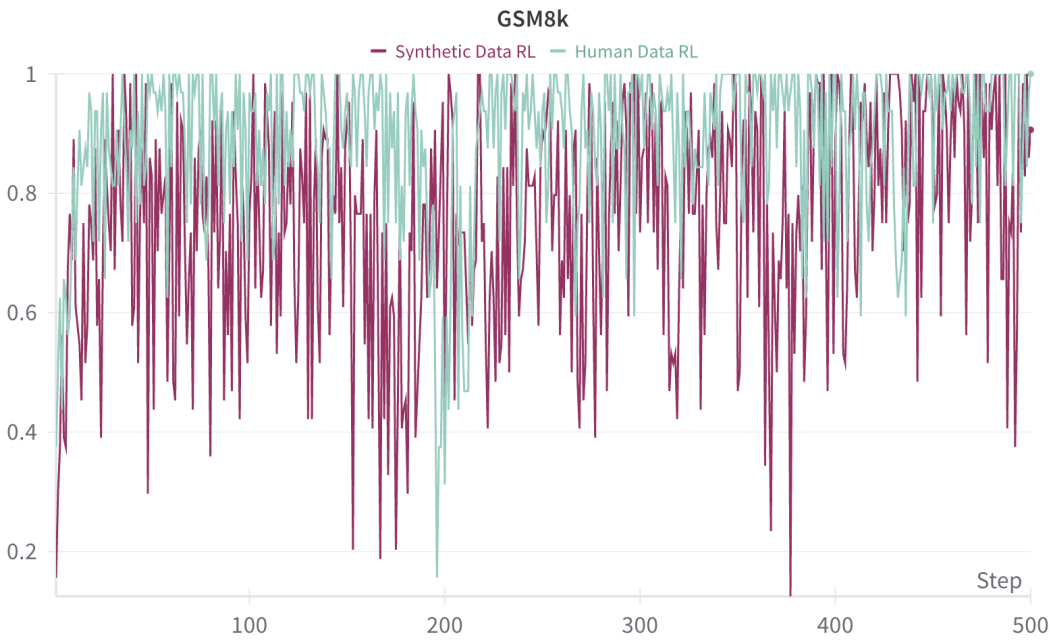

Figure 26: The training score (mean) curve on LogiQA

learning algorithm itself. Lastly, due to computational resource constraints, we were unable to evaluate the performance of larger models, such as a 14B parameter model, or explore the impact of a more extensive data budget, for example, 10,000 instances. These areas present opportunities for future investigation. As a machine learning work, we do not see any negative societal impact.