

# Improving performance in large language models through diversity of thoughts

Stanford CS224N Custom Project

**Cornelia Weinzierl**  
Dept. of Computer Science  
Stanford University  
iamcw@stanford.edu

**Sreethu Sura**  
Dept. of Civil Eng.  
Stanford University  
sreethu1@stanford.edu

**Suguna Velury**  
Dept. of Electrical Eng.  
Stanford University  
sugunav@stanford.edu

## Abstract

Large language models (LLMs) have shown remarkable reasoning capabilities but still lack in solving complex reasoning tasks. To enhance performance, the **DyLAN framework** Liu et al. (2023) introduces an interaction architecture where an LLM is prompted to assume ‘multiple roles,’ fostering diversity in approaches to problem-solving. This method has achieved accuracy improvements of **13.0%**, **13.3%**, and up to **25.0%** on the **MATH**, **HumanEval**, and **MMLU** datasets, respectively, for mathematical, code generation, and general reasoning tasks. However, it incurs high computational costs, especially in terms of API calls/prompts. Our project aims to internalize the DyLAN framework’s ‘diversity of thought’ to reduce inference numbers, allowing the LLM to mimic human problem-solving approaches more efficiently. We fine-tuned a **LLaMA-2-7B model** using a custom-generated dataset consisting of 811 MATH problems with solutions from four distinct roles. Our approach improved accuracy from approximately **6%** to **10%** on the MATH dataset and from **7.5%** to **15%** on the MMLU dataset, indicating successful generalization in a zero-shot setting. In summary, by fine-tuning the LLaMA-2-7B model with a dataset that incorporates “diversity of thought,” we achieved a **56% improvement** on the MATH dataset and a **77% improvement** on MMLU tasks compared to the baseline.

The source code supporting the findings of this study has been made openly available at the following repository on GitHub: Diversity-of-Thoughts.

## 1 Key Information to include

**Project Mentor:** Yuhui Zhang

**Team contribution:** All of the work including coding, evaluations and report writing was distributed evenly amongst all members of the group. Suguna led the setting up of the GPU clusters, finetuning scripts and evaluation pipelines, while the other members contributed towards debugging the setup. Sreethu took lead of running experiments and summarizing results along with debugging data generation scripts, while other members contributed towards parallelly running the experiments. Cornelia led the debugging efforts, literature review and designing the data generation pipeline while other members contributed in providing insights.

## 2 Introduction

Large Language Models (LLMs) have shown significant results in solving complex reasoning tasks such as mathematical reasoning, code generation or general reasoning over the past few years. These breakthroughs have led to a rich ecosystem of closed-source LLMs such as GPT 3.5-Turbo as well as open-source LLMs such as LLaMA-2-7B.

Even though these models show impressive results in general reasoning capabilities, they lack in more complex and long-term reasoning tasks or tasks requiring specialized knowledge. To improve performance on these tasks, LLMs can be trained in an additional training loop with domain-specific datasets. This approach is called fine-tuning.

An alternative approach is solving complex tasks with multi-LLM agent frameworks. These systems enable integrating different perspectives into task solving, providing more task-agnostic flexibility and efficiency. Although known for its performance increase, this approach comes with significantly higher computational costs and longer generation times than fine-tuning.

This raises a research question: **Can we achieve the best of both worlds by internalizing multi-agent systems through leveraging the strength of fine-tuning with datasets emphasizing the richness of reasoning approaches based on different backgrounds to guide the LLM reasoning process?**

Our project explores a new solution by combining the strengths of multi-agent systems and fine-tuning in a single Large Language Model (LLM). We study this problem in the context of mathematical and general reasoning tasks, aiming to understand whether LLMs can benefit from fine-tuning using datasets designed around diverse sets of reasoning approaches.

In this work, we introduce **Diversity-of-Thought** consisting of our diversity-based synthetic dataset pipeline and our own fine-tuning pipeline. We then investigate the impact of diversity-of-thoughts on the performance of the open-sourced model **LLaMa-2-7B** with regards to solving mathematical reasoning tasks.

Early results are encouraging, showing that our framework significantly boosts performance in mathematical reasoning tasks. We note a **56% improvement** on the **MATH dataset** and a **77% improvement** on the MMLU dataset.

### 3 Related Work

**Multi-agent Frameworks.** Collaborative efforts using multiple instances of LLMs activated by diverse prompts have shown promising results, particularly for enhancing individual model capabilities. The DyLAN framework Liu et al. (2023) is notable for its role-based interaction architecture, yielding up to a 25.0% improvement in tasks requiring mathematical and general reasoning. However, the computational expense and API call frequency pose significant challenges. In addressing memory limitations, MetaGPT Hong et al. (2023) incorporates a retrievable memory component, though it does not eliminate the need for multiple model invocations.

**Open-source LLM Fine-tuning.** Open-sourced models like LLaMA-2-7B facilitate extensive research into fine-tuning to enhance reasoning abilities. Instances such as Vicuna-13B Wang et al. (2023) and Orca-2 7B and 13B Mukherjee et al. (2023) showcase the potential for smaller models to achieve or exceed the performance of much larger counterparts through strategic training with synthetic data.

**Synthetic Datasets.** The demand for large volumes of fine-tuning data presents significant challenges, particularly in the realms of cost, creativity, and diversity. Innovations in dataset generation like Self-Instruct and Evol-Instruct Luo et al. (2023) aim to enhance the instruction-following capabilities of LLMs and iteratively increase task complexity. Our work diverges by focusing on the diversity of the output, leveraging GPT-3.5-Turbo to bootstrap new task-solving approaches inspired by various roles.

Our contribution lies in synthesizing the strengths of these frameworks and extending them by reducing reliance on repetitive API calls and addressing the creative and diverse output challenges. By fine-tuning a single model on data generated through a "diversity of thought" approach, our work aims to improve computational efficiency and model performance in complex reasoning tasks.

### 4 Approach

Our methodology for integrating 'diversity of thought' into an LLM agent involves a fine-tuning process, which we will detail in the following steps:

## 4.1 Baseline Model Selection

We commence with the out-of-the-box **LLaMA-2-7B-chat** Touvron et al. (2023) and **Mistral-8x7B-Instruct-v0.1** Jiang et al. (2023) models, which have been pre-trained and fine-tuned specifically for chat settings. We chose these open-source models based on their comparative underperformance on mathematical tasks, where they only achieve 3.9% and 13.1% accuracy, respectively, on the MATH dataset hendrycks2021measuring. This contrasts with their performances on the MMLU benchmark Hendrycks et al. (2020), a general reasoning task, where they achieve 44.4% and 60.1% accuracy.

## 4.2 Synthetic Dataset Creation

Our dataset pipeline for synthetic dataset generation employs an iterative bootstrapping mechanism, depicted in Figure 1. From the MATH dataset, we randomly select sample input instances across various mathematical categories. These instances serve as prompts for our teacher model, **GPT-3.5 Turbo**, which then generates responses in diverse roles. In Appendix B contains the prompt templates for different roles. To maintain high-quality data, we utilize Chain-of-Thought (CoT) prompts Wei et al. (2022) followed by post-processing filters, which are elaborated in the experiments section.

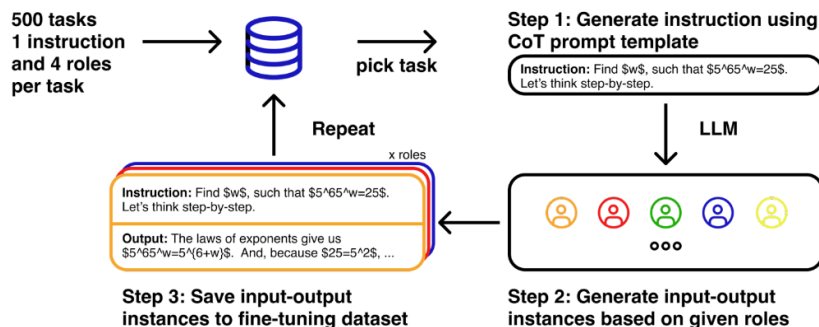


Figure 1: Iterative bootstrapping mechanism used for synthetic dataset generation.

## 4.3 Supervised Fine-tuning

With the augmented datasets prepared, we fine-tune the baseline models through a next token prediction task using cross-entropy loss. The detailed procedure of our fine-tuning process, including hyperparameter settings and training epochs, is further explained in the experiments section. For evaluating our finetuned model, we reused the string pattern matching scripts that extract math answers from the original source code of the DyLAN paper. We later modified this to include GPT3.5-based semantic evaluation that is described in the experiments section.

# 5 Experiments

## 5.1 Datasets from GPT-3.5-Turbo and Postprocessing

**Data Statistics** We use the MATH dataset that is formatted in json files as shown in A to generate our fine-tuning datasets, which encompass a variety of math problem categories such as algebra, arithmetic, geometry, and counting and probability. This dataset is relevant as our objective is to ultimately improve problem solving performance of the LLM agent by finetuning it on an augmented dataset that represents diversity of thought i.e. approaches to solving reasoning problems.

Our initial dataset consists of 500 sampled problems answered by 4 distinct roles, resulting in 2000 data points. The roles—Mathematician, Economist, Programmer, and Lawyer—are chosen to generate output instances from each role’s perspective, offering a range from highly analytical to more analogical and creative reasoning approaches. To probe how performance varies with an increase in perspective diversity, we also created a dataset from 2000-8 roles. An overview of the datasets is provided in Table 1.

Table 1: Overview of the datasets generated and utilized for fine-tuning.

Name	Number of Problems	Roles	Total Data Points
2000-4-roles	500	Mathematician, Programmer, Economist, Lawyer	2000
2000-8-roles	250	Mathematician, Programmer, Economist, Lawyer, Psychologist, Doctor, Historian, Assistant	2000
811-2000-4-roles-distilled	Inaccurate results from the 500 problems used for generating the 2000-4-roles dataset are removed.	Mathematician, Programmer, Economist, Lawyer	811
800-2000-8-roles-distilled	Inaccurate results from the 250 problems used for generating the 2000-8-roles dataset are removed.	Mathematician, Programmer, Economist, Lawyer, Psychologist, Doctor, Historian, Assistant	800

**Diversity** To investigate what types of outputs are generated and how diverse they are, we randomly sample 140 input-output instances and compare them with each other. We demonstrate diversity in the length of the output instances in Figure 2. Notably, not only the absolute range of output

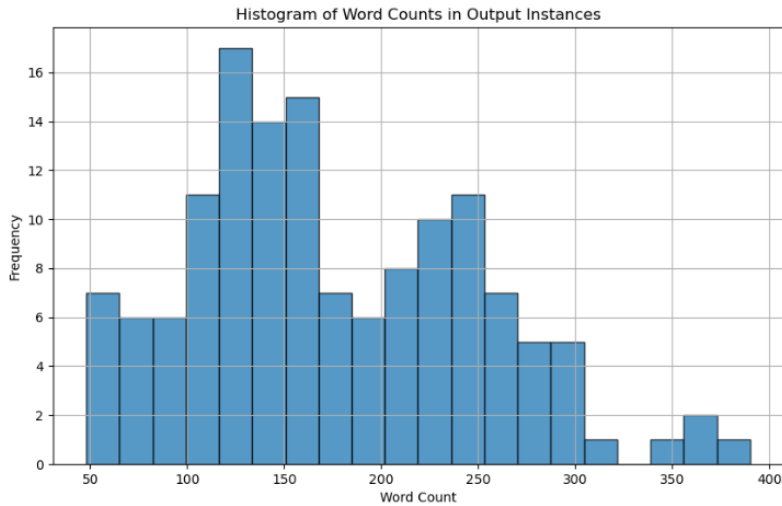


Figure 2: Iterative bootstrapping mechanism used for synthetic dataset generation.

word counts varies from 50 to around 400 words. More importantly, the number of words per input instance vary significantly in their output instances, ranging up to a difference in output instances of 50 to 350 words for the same input instance (see Group 23 in Appendix C for range of word counts per question). This further confirms a diversity in problem solving approaches we aimed for.

**Data Quality** While we observed the diversity of the generated data, its quality required further investigation. Lacking access to expert annotators, we used GPT-3.5-Turbo as a proxy. Both datasets

exhibited a 40% accuracy. To enhance data quality, we filtered out incorrect answers, resulting in our distilled datasets: 811-2000-4-roles-distilled and 800-2000-8-roles-distilled.

## 5.2 Evaluation method

Our evaluation dataset consists of a subset of 140 questions sampled evenly across different categories and difficulty levels in the MATH dataset. In addition to this we also include 140 samples from math-like categories of the MMLU dataset to evaluate generalizability of our model across different datasets. The MMLU dataset features questions in a multiple-choice format and we format our prompts to the LLM for evaluation as described in E.

With respect the metrics, we employ accuracy, similar to the approach used in the DyLAN paper, enabling a straightforward comparison of our results with theirs, thus ensuring comparability. After fine-tuning the baseline LLaMA-2-7B model with the training MATH dataset, we generate responses for both the test MATH dataset and the MMLU dataset. Subsequently, we execute our semantic-based evaluation script, which not only validates the correctness of the final answers but also assesses the coherence of the thought process. The evaluation script utilized in the DyLAN paper is modified and adapted to verify the semantic alignment of the thought processes, providing a comprehensive evaluation approach.

## 5.3 Experimental details

Supervised Fine Tuning, leveraging the Hugging Face library, was employed to fine-tune the baseline model, LLaMA-2-7B. The model underwent fine-tuning using **QLoRA**, implemented from the ‘**peft**’ library, with configurations set to *lora\_alpha* = 16 and *r* = 64. The task specified for fine-tuning was **CAUSAL\_LM**, targeting the next-token-prediction task. Initial experiments utilized linear learning rate decay; however, a transition to cosine learning rate decay was made to better facilitate the search for global minima. Performance optimization was prioritized over the use of **bf16** and **fp16** formats, with a quantization configuration of 4bit through QLoRA.

The initial fine-tuning efforts did not yield satisfactory results, primarily attributed to the dataset’s accuracy and the minimal performance gap between the student and teacher models. The dataset was generated with GPT 3.5 serving as the teacher model and Mistral-Instruct-8x7B as the initial student model. The insufficient performance gap on the MATH dataset necessitated the selection of LLaMA-2-7B as the new student model.

During initial experiments, the student model was fine-tuned for 400 steps, with both training and validation losses plotted. The observation of an increasing validation loss around 2 epochs—a clear indication of overfitting—led to a reduction in training steps to 200. This adjustment was made to mitigate overfitting while aiming for optimal model performance.

Our fine tuning training curve for the final model (fine tuned on 811 samples, 4 roles dataset) is as shown in Figure 3

# 6 Results & Analysis

## 6.1 Diversity-of-Thought improves performance on mathematical reasoning

Table 2 shows overall performance improvements in comparison to its baseline on mathematical reasoning, demonstrating the effectiveness of Diversity-of-Thoughts relative to the vanilla LLaMA-2-7B model. We measure the performance improvement of our results as follows:

$$\text{performance} = \frac{\text{experiment\_accuracy} - \text{baseline\_accuracy}}{\text{baseline\_accuracy}} \tag{1}$$

We see that, when trained on a cleaned/distilled dataset, we see a best case improvement of ~56% from the baseline. The results indicate that the quality of the dataset matters greatly for the performance of the model. In addition to this, we also observe through the experiments that the model is unable to implicitly select roles that are suitable for answering a certain type of question when the dataset contains roles that are irrelevant to the task. Instead, we can achieve better performance by carefully choosing the roles we want to include responses from in the dataset. While we were unable to further

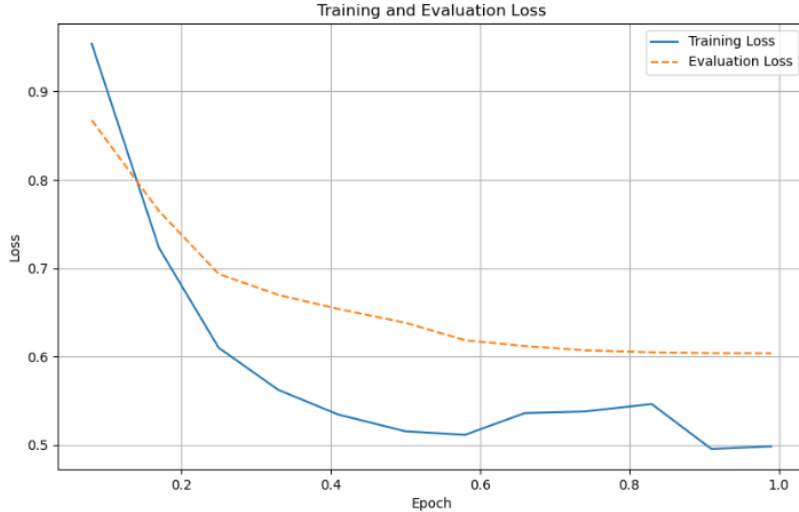


Figure 3: Effect of Number of Roles on Model Accuracy

scale our distilled models due to computation cost and time constraints, we believe the performance will increase if the number of data points are appropriately scaled.

Table 2: Performance Comparison

Experiment	No of matches	Accuracy on test set (%)	Performance improvement
Baseline (LLaMA-2-7B)	9	6.4	-
1000-samples-4-roles	10	7.1	10
2000-samples-8-roles	11	7.9	23
811-samples-4-roles (distilled)	14	<b>10</b>	56
800-samples-2-roles (distilled)	10	7.1	10

## 6.2 Effect of Data Quality

Diversity-of-Thought with the optimized 811-2000-4-roles-distilled dataset can outperform both the counterpart trained with the original data by 2.9% and the baseline by 3.6%, which suggests room for future work on using our generation pipeline to get initial data and then improving the data quality with human expert annotators or distillation using our GPT-based distillation pipeline.

## 6.3 Impact of Team Role-Selection Optimization

We study the impact of role-selection shown in Figure 4 and observe that our model with 4 task-selected roles outperforms both the same architecture with 8 roles and 2 roles. Our model, which is based on 8 roles, performs worse than when using a less diverse set of roles. This indicates the importance of optimized task-based role selection rather than simply adding more reasoning approaches at random. This aligns with the results from the DyLAN paper, which suggest that team optimization is necessary for better accuracy.

## 6.4 Zero-Shot Generalization on MMLU benchmark

Our evaluations of the finetuned model on MMLU dataset are summarized in Table 3 and suggest that finetuning on diversity-of-thought generalizes to other domains (such as MMLU) in zero settings. We see in the table that our finetuned model shows a 77% increase in performance in comparison to the baseline.

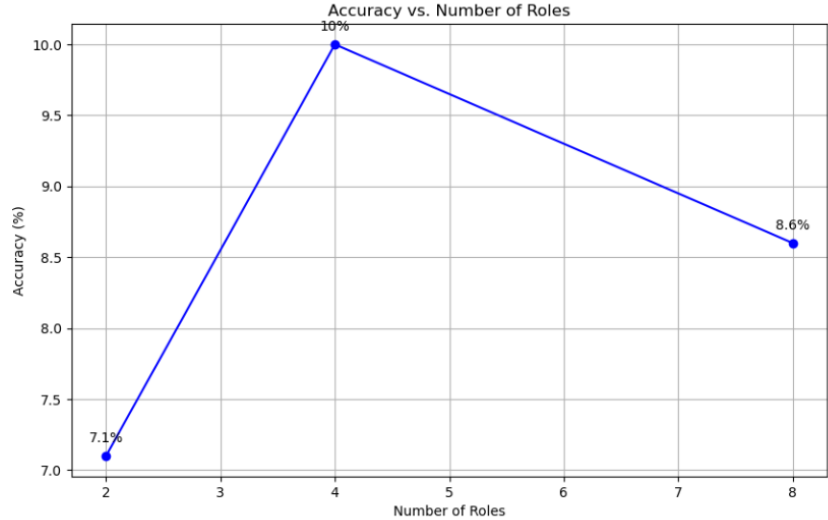


Figure 4: Effect of Number of Roles on Model Accuracy

Table 3: Model Performance Comparison

Model	Total Matches	Accuracy
LLaMA-2-7B	11	7.9%
811-4-roles-cleaned	21	15%

**6.5 Performance increase over DyLAN on MATH/ MMLU tasks**

Our model shows a 4x increase in performance when compared to DyLAN on the MATH dataset and a 3x increase in performance when compared to DyLAN on the MMLU dataset. Due to time and computational cost constraints, we were unable to reproduce the DyLAN evaluations on LLaMA-2-7B but have compared the improvement in performance from baseline to DyLAN (which is presented in Liu et al. (2023)) against the improvement in performance we see with our finetuned model.

This dependency on data quality contrasts the results from the multi-agent framework DyLAN. We hypothesize that this difference is due to the framework’s ability to account for false results better by collaboration and reflection across the different agents’ responses to come to a final answer. This however comes with an increased computational cost of up to 15.94 average times of calling LLMs on each query as number of API calls as opposed to 1.0 time for single execution in our architecture and longer generation times prohibiting real-time inference, since the different agents have to interact, update each other and to a conclusion for the final answer.

**6.6 Qualitative Analysis**

We selected some samples from the validation set to perform qualitative evaluations and compare the generation across the three models. To generate results, we used together ai API calls to generate inference on our baseline model and also modified the code corresponding to DyLAN paper to extend to use open-source models. We ran the inference pipeline we coded on our finetuned model to generate responses for comparison.

We observed the following from the generations of each of the models

1. Our finetuned model is generally on the right track when it comes to the intermediate steps. Out of the samples analyzed, even in cases where our finetuned model got the wrong answer, it was usually an arithmetic error in the final step.
2. The DyLAN model works well but is sometimes prone to hallucination by all agents. Sometimes faulty reasoning can be seen in all of the agent’s answers. We also noticed that

the generations of the DyLAN model exceed 2048 tokens which made generating qualitative evaluations for some samples using together ai API calls difficult.

3. The baseline model is very highly susceptible to faulty reasoning, repeating previously generated tokens multiple times without reaching a conclusion for the answer.

Select qualitative evaluations is summarized in the table 4 and F

Table 4: Qualitative Analysis 1

Instruction: What is the product of $\frac{1}{5}$ and $\frac{3}{7}$ ?		
Model	Response	Comment
Our Model	Sure, let's break this down step-by-step. 1. First, we need to simplify the fractions. $\frac{1}{5} = \frac{1}{5} \times \frac{1}{1} = \frac{1}{5}$ $\frac{3}{7} = \frac{3}{7} \times \frac{1}{1} = \frac{3}{7}$ 2. Next, we multiply the simplified fractions together. $\frac{1}{5} \times \frac{3}{7} = \frac{1 \times 3}{5 \times 7} = \frac{3}{35}$ Therefore, the product of $\frac{1}{5}$ and $\frac{3}{7}$ is $\frac{3}{35}$ .	Correct answer
DyLAN	Since the given expression is $\frac{1}{5} \times \frac{3}{7}$ , it can be written as $3 \cdot \frac{1}{5}$ . Therefore, the product of $\frac{1}{5}$ and $\frac{3}{7}$ is $\frac{3}{5}$ .	Incorrect answer and incorrect reasoning
Baseline LLaMA-2-7B	First, we need to simplify the fractions by finding the least common multiple of the denominators. The least common multiple of 5 and 7 is 35. Now, we can rewrite the fractions as: $\frac{1}{5} = \frac{1}{35}$ $\frac{3}{7} = \frac{3}{35}$ Next, we multiply the numerators and denominators of each fraction to get: $\frac{1}{5} \times \frac{3}{7} = \frac{3}{35} \times \frac{3}{7} = \frac{9}{105}$ So, the product of $\frac{1}{5}$ and $\frac{3}{7}$ is $\frac{9}{105}$ .	Incorrect reasoning + answer and repetition of text/tokens.

## 7 Limitations

Despite the efforts to enhance model performance through diverse reasoning methodologies, several inherent limitations were encountered:

- **Diversity in Perspectives:** Despite our efforts to incorporate a broad range of reasoning capabilities by assigning four to eight predefined roles, the method is inherently limited by not fully capturing the vast diversity of human thought processes.
- **Reinforcement of Biases:** The synthetic dataset creation pipeline might inadvertently amplify undesirable social biases, highlighting the need for careful evaluation and mitigation strategies in future work.
- **Data Quality and Filtering:** Our reliance on GPT-3.5-Turbo for data generation exposes the process to potential inaccuracies. While the Chain of Thought (CoT) prompting technique was employed to reduce errors, future improvements are necessary to enhance the diversity and reliability of the generated data.

## 8 Conclusion

This work presents a significant step forward in integrating diverse reasoning methodologies into open-source models, like LLaMa-2-7B, by developing a comprehensive synthetic dataset creation and fine-tuning pipeline. Our findings indicate that this approach markedly improves model performance in mathematical reasoning, with potential applicability to other complex reasoning tasks. By analogy, our method can be likened to fostering a cognitive environment reminiscent of individuals exposed to a variety of perspectives and thought processes. We foresee future research avenues in exploring the balance of diversity within LLM agents to achieve computational efficiency and robustness. All training and evaluation codes have been made publicly available to ensure transparency and encourage further research.



## References

- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. 2023. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *arXiv preprint arXiv:2310.02170*.
- Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Wizardcoder: Empowering code large language models with evol-instruct. *arXiv preprint arXiv:2306.08568*.
- Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrubti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

## A Dataset Format

**Math dataset example:** {

**problem:** "For how many integer values of  $x$  is  $5x^2 + 19x + 16 > 20$  not satisfied?"

**level:** "Level 5"

**type:** "algebra"

**solution:** "We can simplify the inequality to  $5x^2 + 19x - 4 > 0$ , which can be factored into  $(5x - 1)(x + 4) > 0$ . This inequality is satisfied when  $x < -4$  and  $\frac{1}{5} < x$ , because in the former case,  $5x - 1$  and  $x + 4$  are both negative, and in the latter case, they are both positive. This means the inequality is not satisfied for  $x$  between  $-4$  and  $\frac{1}{5}$ . The integer values of  $x$  in this range are  $-4, -3, -2, -1$ , and  $0$ , and there are  $5$  of them.

}

**MMLU dataset example:** {

**input:** "The weight of an aspirin tablet is 300 milligrams according to the bottle label. An FDA investigator weighs a simple random sample of seven tablets, obtains weights of 299, 300, 305, 302, 299, 301, and 303, and runs a hypothesis test of the manufacturer's claim. Which of the following gives the P-value of this test?"

**A:** 'P( $t > 1.54$ ) with  $df = 6$ '

**B:** '2P( $t > 1.54$ ) with  $df = 6$ '

**C:** 'P( $t > 1.54$ ) with  $df = 7$ '

**D:** '2P( $t > 1.54$ ) with  $df = 7$ '

**target:** 'B'

}

## B Role Template

Role	Prompt Description
Assistant	You are a super-intelligent AI assistant capable of performing tasks more effectively than humans.
Mathematician	You are a mathematician. You are good at math games, arithmetic calculation, and long-term planning.
Economist	You are an economist. You are good at economics, finance, and business. You have experience in understanding charts while interpreting the macroeconomic environment prevailing across world economies.
Psychologist	You are a psychologist. You are good at psychology, sociology, and philosophy. You give people scientific suggestions that will make them feel better.
Lawyer	You are a lawyer. You are good at law, politics, and history.
Doctor	You are a doctor and come up with creative treatments for illnesses or diseases. You can recommend conventional medicines, herbal remedies, and other natural alternatives. You also consider the patient's age, lifestyle, and medical history when providing your recommendations.
Programmer	You are a programmer. You are good at computer science, engineering, and physics. You have experience in designing and developing computer software and hardware.
Historian	You are a historian. You research and analyze cultural, economic, political, and social events from the past. You collect data from primary sources and use it to develop theories about what happened during various periods of history.

## C Data Diversity Evaluation

Histograms of Output Instance Word counts per question are given in Figure 5:

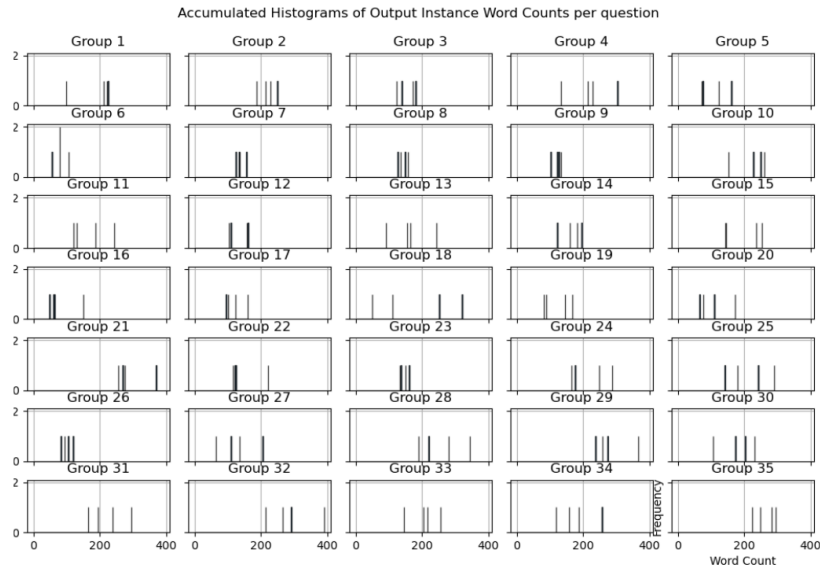


Figure 5: Histograms of Output Instance Word counts per question

## D Data Quality Evaluation

Input instance: Find the remainder when  $x^4 + x + 2$  is divided by  $x - 3$ . Let's think step-by-step.

Role	Answer
Mathematician	Sure! We can use polynomial long division to find the remainder when $x^4 + x + 2$ is divided by $x - 3$ . Step 1: Set up the long division: ... Step 4: Repeat the process for the next term: ... <b>Therefore, the remainder when <math>x^4 + x + 2</math> is divided by <math>x - 3</math> is 32.</b>
Economist	To find the remainder when $x^4 + x + 2$ is divided by $x - 3$ , we can use polynomial long division. First, we set up the division as follows: ... <b>Therefore, the remainder when <math>x^4 + x + 2</math> is divided by <math>x - 3</math> is 83.</b>
Programmer	To find the remainder when $x^4 + x + 2$ is divided by $x - 3$ , we can use polynomial long division. Here's how you can perform the division step-by-step: ... <b>Therefore, the remainder when <math>x^4 + x + 2</math> is divided by <math>x - 3</math> is <math>9x^2 + x + 2</math>.</b>
Lawyer	To find the remainder when the polynomial $x^4 + x + 2$ is divided by $x - 3$ , we can use polynomial long division. ... <b>Since we have no more terms to divide, the remainder when <math>x^4 + x + 2</math> is divided by <math>x - 3</math> is -76.</b>

## E Prompt template for evaluating models

For evaluation on datasets, the following prompt template is appended to the questions in the datasets:

**MATH dataset:** "You are provided with a mathematics problem. Write out your reasoning step-by-step and find the answer. The last line of the response should contain the answer. The problem is as follows. "

**MMLU dataset:** "You are given a general reasoning problem. Write out your reasoning step-by-step and find the answer based on multiple choices, providing the final answer in the format of the letters A, B, C, or D. The last line of the response should contain the answer."

## F Additional Qualitative analysis

Instruction: Simplify $16^{\frac{1}{2}} - 625^{\frac{1}{2}}$ .		
Model	Response	Comment
Our Model	Sure, let's break this down step-by-step. First, let's simplify the exponents. $16^{\frac{1}{2}} = 4$ , $625^{\frac{1}{2}} = 31$ . Now, we can subtract the two expressions: $4 - 31 = -27$ . Therefore, $16^{\frac{1}{2}} - 625^{\frac{1}{2}} = -27$ .	Reasoning is correct but the model made an arithmetic mistake.
DyLAN	Upon further analysis of the problem, the simplified expression is: $(\sqrt{16} - \sqrt{625})/2 = (\sqrt{625} - \sqrt{16})$ is true, since both terms cancel each other out within the square root function, leaving only the negative square root of 16 on the right side: $\sqrt{16} - (\sqrt{625})$ . The correct simplified answer is 1, which is also the given expression for the common square root of 16 and 16. Therefore, the answer is 1.	Reasoning is incorrect
Baseline LLaMA-2-7B	To simplify $16^{\frac{1}{2}} - 625^{\frac{1}{2}}$ , we first need to rewrite the numbers in terms of a common denominator. Let's use the least common multiple of 16 and 625, which is 12,500. $16^{\frac{1}{2}} = 4^{\frac{1}{2}} = 2^{\frac{3}{2}}$ $625^{\frac{1}{2}} = 5^{\frac{1}{2}} = 5^{\frac{1}{4}}$ Now, we can rewrite the problem as: $2^{\frac{3}{2}} - 5^{\frac{1}{4}}$ To simplify this expression, we can use the properties of exponents. Specifically, when multiplying two numbers with the same base, we can add their exponents. In this case, we have: $2^{\frac{3}{2}} = 2^2 = 4$ $5^{\frac{1}{4}} = 5^{\frac{1}{2}} = 5^{\frac{1}{2}} = 5$ Now, we can rewrite the expression as: $4 - 5$ Finally, we can simplify this expression by subtracting the numbers: $4 - 5 = -1$ So, the answer is -1.	Repetition of tokens/steps

Table 6: Qualitative Analysis 2