

Relevant or Random: Can LLMs Truly Perform Analogical Reasoning?

Anonymous ACL submission

Abstract

Analogical reasoning is a unique ability of humans to address unfamiliar challenges by transferring strategies from relevant past experiences. One key finding in psychology is that compared with irrelevant past experiences, recalling *relevant* ones can help humans *better* handle new tasks (Gentner and Smith, 2012). Coincidentally, the NLP community has also recently found that self-generating relevant examples in the context can help large language models (LLMs) better solve a given problem than hand-crafted prompts (Yasunaga et al., 2024). However, it is yet not clear whether relevance is the key factor eliciting such capability, *i.e.*, can LLMs benefit more from self-generated relevant examples than irrelevant ones? In this work, we systematically explore whether LLMs can truly perform analogical reasoning on a diverse set of reasoning tasks. With extensive experiments and analysis, we show that self-generated random examples can surprisingly achieve comparable or even better performance, *e.g.*, 4% performance boost on GSM8K with random biological examples. We find that the accuracy of self-generated examples is the key factor and subsequently design two improved methods with significantly reduced inference costs. Overall, we aim to advance a deeper understanding of LLM analogical reasoning and hope this work stimulates further research in the design of self-generated contexts.

1 Introduction

A hallmark of human intelligence is that they can solve novel problems by drawing analogy from relevant past experiences, a concept known as *analogical reasoning* in cognitive science (Vosniadou and Ortony, 1989). As indicated by the name, recalling previously acquired *relevant* experiences can facilitate humans to *better* tackle new tasks, whereas irrelevant ones are rarely beneficial and can even be distracting (Gentner and Smith, 2012).

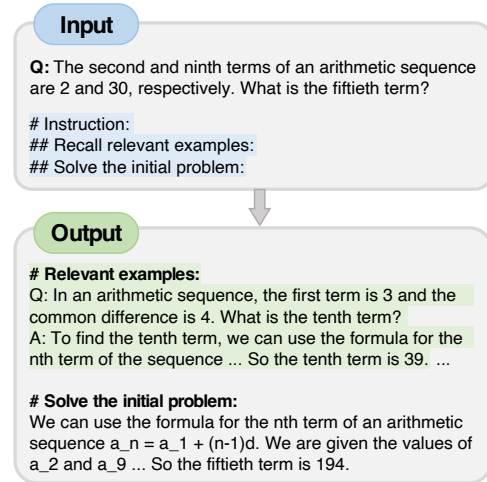


Figure 1: Illustration of LLM analogical reasoning in Yasunaga et al. (2024). LLMs are prompted to self-generate relevant examples as context before solving the new problem.

For instance, when faced with a novel math problem about determinants (*e.g.*, calculating the value of a given fourth-order determinant), humans can resolve this by reflecting upon the methodology employed to ascertain the value of a third-order determinant, whereas biological knowledge (*e.g.*, how the human body regulates its temperature) can generally be considered irrelevant.

With the recent advancements in scaling up model size and data, LLMs have demonstrated impressive zero-shot and few-shot performance across various reasoning tasks, especially, through advanced prompting methods like chain-of-thought (CoT) (Wei et al., 2022). Compared to common approaches such as zero or few-shot CoT (Zhou et al., 2022; Kojima et al., 2022; Wang et al., 2022b), Yasunaga et al. (2024) introduce LLM analogical reasoning, *i.e.*, LLMs self-generate examples relevant to the query as context to better solve new problems; see Fig. 1 for an example. However, it remains unclear whether relevance is the key to eliciting such capability in LLMs. While several

064 studies explore the influence of the relevance of
065 demonstrations in in-context learning and CoT (Liu
066 et al., 2022; Kim et al., 2022; Lyu et al., 2023; Chen
067 et al., 2023; Yang et al., 2023; Wang et al., 2023;
068 Yasunaga et al., 2024), none of them investigate
069 whether self-generated relevant examples consistently
070 outperform irrelevant ones in LLM analogical
071 reasoning.

072 In this paper, to systematically assess the capa-
073 bility of LLMs to perform analogical reasoning,
074 we conduct a series of ablation experiments on a
075 variety of reasoning tasks including problems from
076 GSM8K (Cobbe et al., 2021), MATH (Hendrycks
077 et al., 2021), and BIG-Bench Hard (BBH) (Suzgun
078 et al., 2022). With extensive experiments, we aim
079 to address the following two research questions:

- 080 • **Q1.** Are self-generated *relevant* examples more
081 beneficial to LLMs than *random* ones?
- 082 • **Q2.** If not, what is the pivotal factor for LLMs’
083 performance in analogical reasoning?

084 To answer these questions, we empirically ana-
085 lyze the analogical reasoning abilities of GPT-3.5
086 (turbo) and Llama series models (Touvron et al.,
087 2023). Surprisingly, experimental results show that
088 prompting LLMs to self-generate random exam-
089 ples can achieve comparable or even better perfor-
090 mance on *certain* tasks which is not in line with
091 the key claim of analogical reasoning in Gentner
092 and Smith (2012), indicating that LLMs *cannot*
093 *always* perform analogical reasoning. As for Q2,
094 we point out through controlled experiments that
095 the key factor is *the accuracy of self-generated ex-*
096 *amples*. Informed by these findings, we design two
097 approaches that can outperform existing methods
098 with significantly reduced inference costs. Specif-
099 ically, we ask LLMs to randomly generate a few
100 problems and manually verify their correctness,
101 then use this fixed set of problems as in-context
102 learning demonstrations for all test samples. Con-
103 sistent observations across different model types
104 consolidate the conclusions. We summarize the
105 major contributions of our work below:

- 106 • To the best of our knowledge, we, for the first
107 time, extensively assess the ability of LLMs to
108 perform analogical reasoning and explore their
109 counterintuitive behavior on certain tasks.
- 110 • With extensive experiments and analysis, we
111 demonstrate the effectiveness and limitations of
112 different types of self-generated contexts. Our
113 code base is available at <redacted>.

2 Related Work 114

This work mainly explores whether LLMs can truly
perform analogical reasoning. In light of this, we
review two lines of research that form the basis of
this work: chain-of-thought prompting and LLM
analogical reasoning. 115
116
117
118
119

2.1 Chain-of-Thought Prompting 120

Chain-of-thought (CoT) prompting induces LLMs
to generate intermediate reasoning steps before gen-
erating the final answer (Wei et al., 2022), greatly
improving the reasoning capabilities of LLMs. Typ-
ical CoT prompting approaches include few-shot
CoT (Wei et al., 2022; Zhou et al., 2022; Wang
et al., 2022b; Li et al., 2022; Wang et al., 2022a),
taking several labeled demonstrations of the reason-
ing process, and zero-shot CoT, comprising only
instructions like “Let’s think step by step” (Ko-
jima et al., 2022; Zelikman et al., 2022; Zhang
et al., 2023). Other ongoing research on CoT
has also explored (i) optimizing the demonstra-
tion selection (Fu et al., 2022; Lu et al., 2022; Qin
et al., 2023), (ii) optimizing the quality of reason-
ing chains (Khot et al., 2022; Chen et al., 2022;
Zhao et al., 2023; Shinn et al., 2023), and (iii) CoT
in smaller language models (Magister et al., 2022;
Ho et al., 2022; Fu et al., 2023). 121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139

2.2 LLM Analogical Reasoning 140

While few-shot CoT can provide more detailed
reasoning guidance, it requires labeled examples
which can be unavailable for a new task. To tackle
this problem, Yasunaga et al. (2024) propose ana-
logical prompting to guide LLMs to self-generate
relevant exemplars as few-shot demonstrations,
which is similar to analogical reasoning, *i.e.*, hu-
mans can address new problems by drawing anal-
ogy from relevant past experience (Vosniadou and
Ortony, 1989; Holyoak, 2012). In this work, we
step forward to explore the intrinsic principle of
LLM analogical reasoning. Specifically, we aim
to investigate whether LLMs can authentically ex-
hibit such reasoning capabilities and determine the
extent to which the relevance of self-generated ex-
amples contributes to enhancing this process. 141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156

3 Methodology 157

We mainly follow the analogical prompting ap-
proach outlined in Yasunaga et al. (2024). Specif-
ically, for a given target problem x , analogical
prompting introduces instructions like: 158
159
160
161

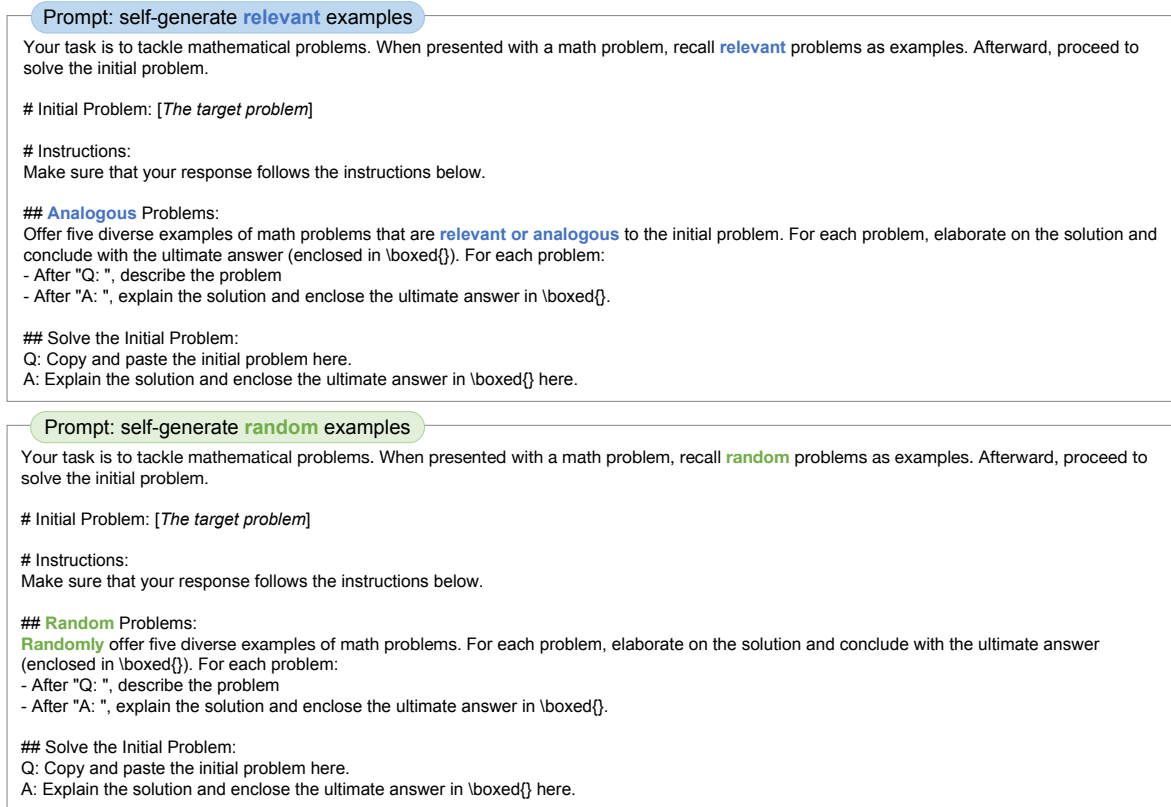


Figure 2: Example prompts for GSM8K (mathematical reasoning). **Top**: The original prompt used in Yasunaga et al. (2024) for self-generating *relevant* math problems. **Bottom**: The prompt designed for self-generating *random* math problems. We mark the differences between these two prompts in blue and green respectively.

162	# Problem: [x]	same problem type (e.g., math).	183
163	# Relevant problems: Recall five <u>relevant</u> and diverse problems. For each problem, describe it and explain the solution.	• <i>Random_{diff}</i> : randomly generate examples of different problem types (e.g., any type except math).	184
164	# Solve the initial problem:	• <i>Random_{bio}</i> : randomly generate biological problems.	185
165	The goal is to induce LLMs to self-generate <i>relevant</i> examples, aiding them to solve the target problem via in-context learning. To ensure better performance and efficiency, several key technical decisions are made in Yasunaga et al. (2024):	Yasunaga et al. (2024) demonstrate that self-generating relevant examples can consistently outperform zero-shot CoT and few-shot CoT (handcrafted examples or retrieved top- k most similar training samples) on different tasks. Therefore, we do not include these two methods in our work. Interested readers can refer to the corresponding results and analysis in Yasunaga et al. (2024). In addition, we show prompts for different methods on all datasets in Appendix A.1.	186
166	• The self-generated examples should be relevant and diverse, achieved through a specially designed instruction.		187
167	• Generate relevant problems and the solution to the initial problem in one pass.		188
168	• 3 to 5 self-generated examples perform the best.		189
169	In this work, we make minimal changes to the original prompt to guide LLMs to generate different types of <i>irrelevant</i> examples as context; see Fig. 2 for example prompts:		190
170	• <i>N/A</i> : generate problems that are N/A (not applicable) to the initial problem.		191
171	• <i>Random_{same}</i> : randomly generate examples of the		192
172			193
173			194
174			195
175			196
176			197
177			198
178			199
179			200
180			201
181			202
182			203
			204
			205

Method	Temporal sequences	Logical deduction five objects	Reasoning about colored objects	Formal fallacies	Word sorting	Average
Relevant	60.0	51.2	76.7	51.2	76.9	63.2
N/A	57.5	45.3	75.5	53.3	77.7	61.9
Random _{same}	53.1	48.8	73.5	52.4	74.1	60.4
Random _{diff}	44.3	44.8	72.4	51.2	69.2	56.4
Random _{bio}	57.1	49.5	76.1	50.8	74.9	61.7

Table 1: Accuracy (%) of different methods on five reasoning tasks in BBH. **Bold** indicates the best results. Self-generated *relevant* examples achieve the best average performance. Detailed results for different seeds are reported in Appendix A.2.

Method	Task		
	GSM8K	MATH	Average
Relevant	71.5	33.3	52.4
N/A	75.5	36.1	55.8
Random _{same}	75.1	36.3	55.7
Random _{diff}	76.3	34.1	55.2
Random _{bio}	75.3	34.6	54.9

Table 2: Accuracy (%) of different methods on two mathematical reasoning tasks. Self-generated *irrelevant* examples are consistently better than *relevant* ones. Table 14 in Appendix A.2 reports detailed results for different seeds.

206 2021) and MATH (Hendrycks et al., 2021). For
 207 each dataset, we randomly sample 500 examples
 208 from the original test set and run experiments
 209 three times with different random seeds (resulting
 210 in different test samples).

- **Other reasoning.** Following Yasunaga et al. (2024), we evaluate five reasoning tasks in BIG-Bench Hard (BBH) (Suzgun et al., 2022): temporal sequences (temporal reasoning), logical deduction five objects and reasoning about colored objects (logical reasoning), formal fallacies (deductive reasoning) and word sorting (symbolic reasoning). For each task, we use all test samples for evaluation and run experiments three times with different random seeds.

221 We mainly use GPT-3.5 (gpt-3.5-turbo) as the
 222 LLM and obtain all outputs from it with the temper-
 223 ature set to 0. We ask the LLM to self-generate 5
 224 examples for GSM8K, 3 examples for MATH and
 225 BBH following Yasunaga et al. (2024).

226 4.2 Main Results

227 We now address the research questions asked in §1
 228 with empirical results.



Q1. Are self-generated relevant examples more beneficial to LLMs than random ones?

The results averaged over all random seeds are reported in Table 1 and Table 2; more detailed results for every seed are shown in Appendix A.2.

- **Self-generated relevant examples achieve the best average performance on BBH.** From the results in Table 1, we can observe that the superiority of self-generated relevant examples is empirically substantiated on BBH. Specifically, using relevant examples, denoted by ‘relevant’, outperforms other approaches on temporal and logical reasoning tasks. While it performs worse than ‘N/A’ on deductive and symbolical reasoning, it can still improve the accuracy by **1.3%** on average compared to ‘N/A’.

However, the results on mathematical reasoning tasks are quite counterintuitive as described below:

- **Relevant examples do not guarantee better performance.** Different from BBH, all types of self-generated irrelevant examples consistently outperform relevant ones on both mathematical reasoning datasets, showing that LLMs cannot yet perform analogical reasoning on these tasks. Interestingly, when we use randomly generated biological examples (e.g., how the process of photosynthesis occurs in plants), they can yield about **2.5%** better results on average compared to generating relevant math problems. Besides, ‘N/A’ achieves the best average result as it is the second-best on both datasets.

Problems in MATH span various subjects and difficulty levels. To investigate whether the inferior performance of relevant examples on MATH is accidentally caused by certain categories, we further report the accuracy across different subjects

Method	Precalculus	Intermediate Algebra	Algebra	Prealgebra	Counting & Probability	Geometry	Number Theory
Relevant	10.4	9.8	51.8	56.8	22.1	24.2	37.0
N/A	9.1	15.7	55.5	61.0	28.7	25.8	34.2
Random _{same}	12.3	17.6	54.4	60.6	25.4	25.8	34.9
Random _{diff}	13.0	14.1	52.7	56.8	26.2	24.2	33.6
Random _{bio}	13.0	12.2	53.0	59.2	28.7	25.8	32.2

Table 3: Accuracy (%) across different subjects in the MATH dataset. Self-generated irrelevant examples outperform relevant ones on 6 out of 7 subjects.

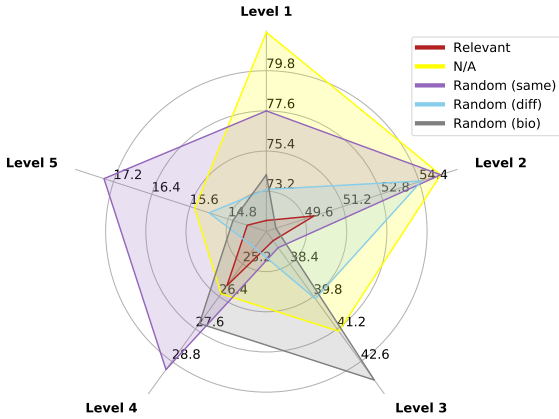


Figure 3: Comparison of all methods at different difficulty levels on the MATH dataset. Level 1 represents the easiest and level 5 is the hardest. ‘relevant’ clearly performs worse than other approaches at all difficulty levels.

and difficulty levels in Table 3 and Fig. 3. The consistent performance gap between ‘relevant’ and other methods across different problem categories demonstrates the inherent flaws of relevant examples, indicating that *mathematical reasoning tasks exhibit different analogical reasoning paradigms from other reasoning tasks*.

It might present challenges to prompt LLMs to accurately generate specific types of demonstrations. Therefore, given the unexpected results on mathematical reasoning tasks, one may wonder:

Q1-1. Are self-generated examples really relevant or irrelevant to the query?

To quantitatively measure the relevance between the generated examples and the query, we compute the average cosine similarity between them. Following Zhang et al. (2023), we use the sentence transformer (Reimers and Gurevych, 2019) to encode all samples. For each method, the reported result is averaged across three seeds.

As observed from Table 4, relevant examples

Method	GSM8K	MATH	Average
Relevant	0.54	0.41	0.48
N/A	0.19	0.28	0.24
Random _{same}	0.30	0.20	0.25
Random _{diff}	0.15	0.10	0.13
Random _{bio}	0.06	0.11	0.09
Oracle	0.65	0.63	0.64

Table 4: Average relevance score (semantic similarity) between self-generated examples and the query. ‘Oracle’ stands for the average similarity score between the query and k most similar training samples (k is the number of self-generated examples).

	Relevant	N/A	Random _{same}
Accuracy	62.0	72.0	86.0

Table 5: Accuracy (%) of self-generated examples on the MATH dataset. The examples generated by ‘relevant’ are less accurate.

are much more semantically similar to the query than irrelevant ones and the relevance score of ‘relevant’ is more biased towards ‘oracle’ rather than ‘random’ or ‘N/A’, demonstrating that *LLMs indeed follow instructions to generate specific types of demonstrations*. Furthermore, we calculate the average similarity score between self-generated relevant examples and queries for BBH (0.46), which is slightly lower than the score of mathematical reasoning tasks (0.48). This result demonstrates that the difference in analogical reasoning performance between BBH (Table 1) and mathematical reasoning (Table 2) is *not* because LLMs can generate more relevant examples for BBH.

We provide a case study in Table 6 to delve deeper into the demonstrations of different methods. As we can notice, the example generated by ‘relevant’ is more related to the query as they both involve the mathematical concept ‘number bases’.


Query:	For how many ordered pairs (A, B) where A and B are positive integers is $AAA_7 + BBB_7 = 666_7$?
Relevant	In a certain base, the sum of two three-digit numbers is 777. If the digits of one of the numbers are reversed, the sum becomes 888. What is the base of this number system?
N/A	What is the value of x in the equation $2x + 5 = 10$?
Random _{same}	In a bag, there are 5 red marbles, 3 blue marbles, and 2 green marbles. If you randomly pick 2 marbles from the bag without replacement, what is the probability that both marbles are red?
Random _{diff}	How do you bake chocolate chip cookies?
Random _{bio}	How does the process of photosynthesis occur in plants?
Oracle	Find the number of ordered pairs (a, b) of complex numbers such that $a^3b^5 = a^7b^2 = 1$.

Table 6: Demonstration examples of different methods on the MATH dataset. The example generated by ‘relevant’ is more related to the query than other examples generated by ‘N/A’ or ‘random’.

Variant	GSM8K			MATH		
	Relevant	N/A	Random _{same}	Relevant	N/A	Random _{same}
ICL	71.2	73.8	72.0	37.0	39.8	39.2
GPT4-Calibration	75.2	75.6	75.6	44.4	41.2	40.0
Random	70.0	72.0	68.4	36.0	38.0	37.8

Table 7: Accuracy (%) of different variants on GSM8K and MATH. When using GPT4-generated answers (mostly accurate), ‘GPT4-Calibration’ consistently outperforms ‘ICL’ for all methods. In contrast, ‘random’ always performs worse than ‘ICL’.

In contrast, examples such as ‘What is the value of x in the equation $2x + 5 = 10$?’ (N/A) or ‘How do you bake chocolate chip cookies?’ (Random_{diff}) are less relevant to the query. This comparison highlights once again that relevance may not be the key factor for analogical reasoning performance on mathematical reasoning tasks. To understand better the underlying reasons for the counterintuitive results, we then ask the following question:

 **Q2.** If relevance is not the key factor, what is more important for the accuracy of analogical reasoning?

Looking back at Table 6, an interesting observation is that the self-generated relevant example appears to be more difficult to solve than the irrelevant ones, regardless of whether they are math problems or not. Consequently, the accuracy of relevant examples may be lower. To verify this, we conduct a pilot experiment on MATH. Specifically, we randomly select 50 samples for different types of generated math problems, *i.e.*, Relevant, N/A and Random_{same}, and manually evaluate their accuracy. We exclude other methods as it is difficult to define the ‘accuracy’ of the examples they generate. From the results in Table 5, we can observe that while the examples generated by ‘relevant’ are more related to the test query, *they are less accurate*, raising the question whether the performance

of different approaches on mathematical reasoning tasks is strongly correlated with the accuracy of self-generated examples.

Proxy Approaches However, as the accuracy of the examples located at the output cannot be directly controlled, we meticulously design a variant called *ICL*, which extracts the generated examples from the model output as in-context learning (ICL) demonstrations and combines them with the query as input to LLMs, as a proxy for the original method. We also consider the following two variants: (a) *GPT4-Calibration* which replaces the answers of demonstrations in *ICL* with GPT4-generated answers, and (b) *Random* changes the answers of demonstrations in *ICL* to random numbers. Our manual verification confirmed that GPT4-generated answers were mostly accurate. We conduct this experiment on GSM8K and MATH with GPT-3.5 as the LLM reasoner.

From the results of different variants reported in Table 7, we can see that increasing the accuracy of generated examples can indeed improve the performance: *GPT4-Calibration* consistently outperforms *ICL* by incorporating more accurate answers. In contrast, *random* always performs the worst among all variants. Therefore, the key factor influencing the performance on mathematical reasoning is *the accuracy of self-generated examples*

Input

Randomly recall five examples of math problems. Note that your problems should be distinct from each other (e.g., involving different numbers and names). For each problem:

- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Output

Q: If John has 5 apples and he buys 8 more, how many apples does he have in total?
A: To find the total number of apples John has, we add the number of apples he initially has to the number of apples he buys. So, we calculate $5 + 8 = 13$. Therefore, John has `\boxed{13}` apples in total.

...

Q: Find the derivative of the function $f(x) = 3x^2 + 2x - 5$.
A: To find the derivative of $f(x)$, we apply the power rule to each term. The derivative of $3x^2$ is $6x$, the derivative of $2x$ is 2 , and the derivative of a constant -5 is 0 . Therefore, the derivative of $f(x)$ is $f'(x) = 6x + 2$. So, the derivative is `\boxed{6x + 2}`.

Figure 4: Example prompts and outputs for randomly generating math problems. We manually verify the answers to ensure the correctness of the generated examples.

Method	Task		
	GSM8K	MATH	Average
Relevant	71.5	33.3	52.4
N/A	75.5	36.1	55.8
Random _{same}	75.1	36.3	55.7
Random _{diff}	76.3	34.1	55.2
Random _{bio}	75.3	34.6	54.9
ICL _{math}	75.7	36.8	56.3
ICL _{bio}	77.9	34.9	56.4

Table 8: Comparison of different methods on two mathematical reasoning tasks.

Method	Task		
	GSM8K	MATH	Average
ICL _{math}	75.7	36.8	56.3
ICL _{math_repeat}	73.8	36.2	55.0

Table 9: Comparison of two ICL variants on the GSM8K and MATH datasets.

Relevant	N/A	Random _{same}	Random _{diff}	Random _{bio}
71.2	74.9	75.3	75.9	74.3

Table 10: Accuracy (%) of different methods with a new prompt.

rather than their relevance.

It is worthwhile to note that while several papers explore how the correctness of demonstration answers influences in-context learning (Min et al., 2022; Yoo et al., 2022; Wei et al., 2023; Pan et al., 2023; Kossen et al., 2024), our work differs from them in the following aspects: (i) The examples in our work are generated by LLMs rather than real data from NLP benchmarks, *i.e.*, randomly sampled from the training set. In addition, there are rationales (CoT) in self-generated examples, which are different from the input-label format of in-context learning investigated in these papers; and (ii) These studies mainly evaluate in-context learning on different classification or multi-choice datasets, *i.e.*, the output space is a finite set. In contrast, we are evaluating mathematical reasoning tasks, where the output space is infinite.

Given the above findings, a natural question is:

Q2-1. Can we ask the LLM to randomly generate a few math or biological problems and manually verify their correctness, then use this fixed set of problems as ICL demonstrations for all test queries?

We refer to these two methods as ICL_{math} and ICL_{bio}, and conduct experiments with them on GSM8K and MATH (see Fig. 4 for example prompts and outputs for generating math problems). Detailed prompts and outputs for different methods are provided in Appendix A.3. Following the original setting, we ask the LLM to randomly generate 5 examples for GSM8K and 3 examples for MATH. As observed from Table 8, ensuring the accuracy of self-generated examples does lead to better performance regardless of the problem type. ICL_{math} and ICL_{bio} achieve similar average performance, once again demonstrating that relevance does not matter (see Appendix A.4 for more analysis on relevance). Moreover, both ICL variants only need to generate examples once, which significantly reduces the inference cost and further demonstrates their superiority.

4.3 Further Analysis

Repeating Problems While generating a few accurate problems as ICL demonstrations can achieve better performance, a bolder idea might be to generate one problem and repeat it multiple times as few-shot demonstrations for ICL. To investigate this,

Method	Relevant	N/A	Random _{same}	Random _{diff}	Random _{bio}	ICL _{math}	ICL _{bio}
Llama-2-70b-Chat	45.1	51.4	50.9	54.3	47.1	55.5	56.1
Llama-3-8B-Instruct	69.5	72.3	72.6	74.1	73.5	75.8	76.8

Table 11: Accuracy (%) of different methods on GSM8K using Llama-2-70b-Chat and Llama-3-8B-Instruct models. Self-generated relevant examples always perform worse than irrelevant ones and both ICL variants outperform other approaches.

Variant	Method		
	Relevant	N/A	Random _{same}
ICL	56.2	58.2	58.6
GPT4-Calibration	60.8	61.0	60.8
Random	53.2	54.0	59.6

Table 12: Accuracy (%) of different variants on GSM8K using Llama-2-70b-Chat. ‘GPT4-Calibration’ consistently performs better than ‘ICL’ and ‘random’ for different methods.

we randomly select a generated math problem and repeat it to perform ICL, denoted by ICL_{math_repeat}. From the results shown in Table 9, we can see that ICL_{math_repeat} consistently performs worse than ICL_{math} on both datasets, indicating that the diversity of generated problems also matters.

Robustness to Prompt Formats To verify the robustness of different methods to prompt formats, we experiment with a new prompt paraphrased from the original one by GPT-4 and present the results on GSM8K in Table 10. We also observe better performance with irrelevant examples than relevant ones, showing the robustness.

Generalization to Open-source LLMs Our experiments and analysis so far used GPT-3.5 as the LLM, which is closed-source and gets updated over time. To verify whether the observations and conclusions are consistent across different models and additionally for reproducibility, we extend the experiments to Llama-2-Chat (Touvron et al., 2023). Specifically, we use vLLM (Kwon et al., 2023) to serve a Llama-2-70b-Chat model for experiments and report the results of different methods/variants on GSM8K in Table 11 and Table 12. We can draw similar observations: (i) self-generated relevant examples underperform all types of irrelevant ones, (ii) ‘GPT4-Calibration’ consistently outperforms the other two variants, and (iii) ICL_{math} and ICL_{bio} perform better than other approaches, demonstrating that the conclusions can be generalized to different models.

We further conduct experiments with the latest

Number	Relevant	N/A	Random _{same}	Random _{diff}	Random _{bio}
3	73.1	77.3	75.0	75.3	75.5
5	71.5	75.5	75.1	76.3	75.3

Table 13: Accuracy (%) of all methods with different numbers of demonstrations.

Llama-3-8B-Instruct model. The results reported in Table 11 demonstrate the generalizability of the conclusions across different model scales. In addition, since investigating analogical reasoning requires LLMs to self-generate different types of problems, we only experiment with instruction-tuned LLMs to ensure that they can follow the given instructions.

Different Numbers of Demonstrations While we mainly follow the setting in Yasunaga et al. (2024) to ask the LLM to generate $k = 5$ examples for GSM8K, we further investigate the effect of the number of demonstrations. Specifically, we conduct controlled experiments with $k = 3$ and report the results in Table 13. We can observe that irrelevant examples consistently outperform relevant ones across different numbers of demonstrations, emphasizing their effectiveness.

In addition, we show the generalization to different tasks in Appendix A.5.

5 Conclusion

In this work, we have systematically assessed the capability of LLMs to perform analogical reasoning. We have identified key research questions and empirically analyzed a representative set of LLMs on a diverse collection of reasoning tasks. Extensive experimental results and analysis show that LLMs *cannot always* perform analogical reasoning and the key influencing factor is the accuracy of self-generated examples rather than their relevance. Given these findings, we have designed two ICL-based approaches with better performance and significantly reduced inference costs. In the future, we would like to investigate additional analogical prompting methods to generate more accurate examples.

465 Limitations

466 This work has several limitations. First, due to the
467 inference cost of ChatGPT¹, we conduct experi-
468 ments on subsets of the test data for mathematical
469 reasoning tasks. Besides, we include 3 datasets re-
470 quiring different reasoning capabilities in this work.
471 A further improvement could be to explore more
472 diverse types of tasks.

473 References

474 Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and
475 Hsin-Hsi Chen. 2023. [Self-ICL: Zero-shot in-context
476 learning with self-generated demonstrations](#). In *Pro-
477 ceedings of the 2023 Conference on Empirical Meth-
478 ods in Natural Language Processing*, pages 15651–
479 15662, Singapore. Association for Computational
480 Linguistics.

481 Wenhui Chen, Xueguang Ma, Xinyi Wang, and
482 William W Cohen. 2022. [Program of thoughts
483 prompting: Disentangling computation from reason-
484 ing for numerical reasoning tasks](#). *arXiv preprint
485 arXiv:2211.12588*.

486 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian,
487 Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias
488 Plappert, Jerry Tworek, Jacob Hilton, Reiichiro
489 Nakano, et al. 2021. [Training verifiers to solve math
490 word problems](#). *arXiv preprint arXiv:2110.14168*.

491 Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and
492 Tushar Khot. 2023. [Specializing smaller language
493 models towards multi-step reasoning](#). *arXiv preprint
494 arXiv:2301.12726*.

495 Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark,
496 and Tushar Khot. 2022. [Complexity-based prompt-
497 ing for multi-step reasoning](#). *arXiv preprint
498 arXiv:2210.00720*.

499 D. Gentner and L. Smith. 2012. [Analogical reason-
500 ing](#). In V.S. Ramachandran, editor, *Encyclopedia of
501 Human Behavior (Second Edition)*, second edition
502 edition, pages 130–136. Academic Press, San Diego.

503 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul
504 Arora, Steven Basart, Eric Tang, Dawn Song, and Ja-
505 cob Steinhardt. 2021. [Measuring mathematical prob-
506 lem solving with the math dataset](#). *arXiv preprint
507 arXiv:2103.03874*.

508 Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022.
509 [Large language models are reasoning teachers](#). *arXiv
510 preprint arXiv:2212.10071*.

511 Keith J Holyoak. 2012. [Analogy and relational reason-
512 ing](#). *The Oxford handbook of thinking and reasoning*,
513 pages 234–259.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao
Fu, Kyle Richardson, Peter Clark, and Ashish Sab-
harwal. 2022. [Decomposed prompting: A modular
approach for solving complex tasks](#). *arXiv preprint
arXiv:2210.02406*. 514
515
516
517
518

Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taek
Kim, Kang Min Yoo, and Sang-goo Lee. 2022.
[Self-generated in-context learning: Leveraging auto-
regressive language models as a demonstration gen-
erator](#). *arXiv preprint arXiv:2206.08082*. 519
520
521
522
523

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yu-
taka Matsuo, and Yusuke Iwasawa. 2022. [Large lan-
guage models are zero-shot reasoners](#). In *Thirty-sixth
Conference on Neural Information Processing Sys-
tems (NeurIPS 2022)*. 524
525
526
527
528

Jannik Kossen, Yarin Gal, and Tom Rainforth. 2024. [In-
context learning learns label relationships but is not
conventional learning](#). In *The Twelfth International
Conference on Learning Representations*. 529
530
531
532

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying
Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E.
Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Effi-
cient memory management for large language model
serving with pagedattention](#). In *Proceedings of the
ACM SIGOPS 29th Symposium on Operating Systems
Principles*. 533
534
535
536
537
538
539

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen,
Jian-Guang Lou, and Weizhu Chen. 2022. [On the
advance of making language models better reasoners](#).
arXiv preprint arXiv:2206.02336. 540
541
542
543

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan,
Lawrence Carin, and Weizhu Chen. 2022. [What
makes good in-context examples for GPT-3?](#) In
*Proceedings of Deep Learning Inside Out (DeeLIO
2022): The 3rd Workshop on Knowledge Extrac-
tion and Integration for Deep Learning Architectures*,
pages 100–114, Dublin, Ireland and Online. Associa-
tion for Computational Linguistics. 544
545
546
547
548
549
550
551

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu,
Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark,
and Ashwin Kalyan. 2022. [Dynamic prompt learning
via policy gradient for semi-structured mathematical
reasoning](#). *arXiv preprint arXiv:2209.14610*. 552
553
554
555
556

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer,
and Hannaneh Hajishirzi. 2023. [Z-ICL: Zero-shot
in-context learning with pseudo-demonstrations](#). In
*Proceedings of the 61st Annual Meeting of the As-
sociation for Computational Linguistics (Volume 1:
Long Papers)*, pages 2304–2317, Toronto, Canada.
Association for Computational Linguistics. 557
558
559
560
561
562
563

Lucie Charlotte Magister, Jonathan Mallinson, Jakub
Adamek, Eric Malmi, and Aliaksei Severyn. 2022.
[Teaching small language models to reason](#). *ArXiv
preprint, abs/2212.08410*. 564
565
566
567

¹experiments done between 12/2023 and 05/2024

568	Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe,	An empirical study of what matters . In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.	625
569	Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022.		626
570	Rethinking the role of demonstrations: What makes in-context learning work? In <i>Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing</i> , pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.		627
571			628
572			629
573			
574		Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022a. Rationale-augmented ensembles in language models . <i>arXiv preprint arXiv:2207.00747</i> .	630
575			631
576	Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. 2023. What in-context learning “learns” in-context: Disentangling task recognition and task learning . In <i>Findings of the Association for Computational Linguistics: ACL 2023</i> , pages 8298–8319, Toronto, Canada. Association for Computational Linguistics.		632
577			633
578		Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models . <i>arXiv preprint arXiv:2203.11171</i> .	634
579			635
580			636
581			637
582	Chengwei Qin, Aston Zhang, Anirudh Dagar, and Wenming Ye. 2023. In-context learning with iterative demonstration selection . <i>arXiv preprint arXiv:2310.09881</i> .		638
583			639
584		Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models . In <i>Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022)</i> .	640
585			641
586	Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks . In <i>Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)</i> , pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.		642
587			
588		Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently . <i>arXiv preprint arXiv:2303.03846</i> .	643
589			644
590			645
591			646
592			647
593			
594	Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning .		648
595			649
596		Jinghan Yang, Shuming Ma, and Furu Wei. 2023. Auto-icl: In-context learning without human supervision . <i>arXiv preprint arXiv:2311.09263</i> .	650
597			
598	Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them . <i>arXiv preprint arXiv:2210.09261</i> .		651
599			652
600		Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H. Chi, and Denny Zhou. 2024. Large language models as analogical reasoners . In <i>The Twelfth International Conference on Learning Representations</i> .	653
601			654
602			655
603			
604	Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge . In <i>Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)</i> , pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.		656
605			657
606			658
607			659
608			660
609			661
610			662
611			663
612			
613	Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models . <i>arXiv preprint arXiv:2307.09288</i> .		664
614			665
615			666
616			667
617			668
618			669
619			670
619	Stella Vosniadou and Andrew Ortony. 1989. <i>Similarity and analogical reasoning</i> . Cambridge University Press.		671
620			672
621			673
622	Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. Towards understanding chain-of-thought prompting: A knowledge-enhanced chain-of-thought framework . In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.		674
623			675
624			676
			677
			678

679 Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei,
680 Nathan Scales, Xuezhi Wang, Dale Schuurmans,
681 Olivier Bousquet, Quoc Le, and Ed Chi. 2022.
682 Least-to-most prompting enables complex reason-
683 ing in large language models. *arXiv preprint*
684 *arXiv:2205.10625*.

685 **A Appendix**

686 **A.1 Prompts for Different Methods**

687 The prompts for different methods on all datasets
688 are shown in [Fig. 5](#) ~ [Fig. 7](#).

689 **A.2 Detailed Results for Different Random** 690 **Seeds**

691 We report detailed results for different random
692 seeds in [Table 14](#) ~ [Table 15](#).

693 **A.3 Prompts and Outputs for Example** 694 **Generation**

695 We show detailed prompts and outputs for ran-
696 domly generating math and biological problems
697 in [Fig. 8](#) and [Fig. 9](#), respectively.

698 **A.4 Guided Problem Generation**

699 In addition to random problem generation in §4.2-
700 Q2-1, we further investigate guided problem gener-
701 ation. Specifically, we randomly select 5 training
702 samples to guide LLMs to self-generate relevant
703 math problems. We then manually verify their cor-
704 rectness and use this fixed set of problems as ICL
705 demonstrations for experiments. The performance
706 of this approach (56.1) is slightly lower than that of
707 ICL_{math} (56.3), verifying that relevance is not the
708 key influencing factor.

709 **A.5 Generalization to Different Tasks**

710 To test the generalizability of our findings beyond
711 the math domain, we further conduct experiments
712 on CommonsenseQA (commonsense reasoning)
713 (Talmor et al., 2019). The comparison between
714 different methods is shown in [Table 16](#), which
715 demonstrates that our findings can be generalized
716 to different types of tasks.

Seed	GSM8K					MATH				
	Relevant	N/A	Random _{same}	Random _{diff}	Random _{bio}	Relevant	N/A	Random _{same}	Random _{diff}	Random _{bio}
42	71.8	76.6	73.2	74.0	74.0	37.4	42.2	41.6	39.0	39.2
100	71.2	75.2	75.2	75.8	74.8	29.0	30.6	32.6	29.4	31.2
1000	71.4	74.8	77.0	79.2	77.0	33.6	35.6	34.6	34.0	33.4
Average	71.5	75.5	75.1	76.3	75.3	33.3	36.1	36.3	34.1	34.6

Table 14: Accuracy (%) of all methods with different random seeds on two mathematical reasoning tasks.

Seed		Temporal sequences	Logical deduction five objects	Reasoning about colored objects	Formal fallacies	Word sorting	Average
42	Relevant	58.0	52.8	76.0	50.4	77.2	62.9
	N/A	56.4	44.8	77.6	54.0	76.8	61.9
	Random _{same}	52.4	48.8	74.8	51.6	72.8	60.1
	Random _{diff}	43.2	46.8	74.0	52.4	67.6	56.8
	Random _{bio}	56.8	52.0	74.0	52.0	76.4	62.2
100	Relevant	58.4	50.8	78.4	51.2	76.8	63.1
	N/A	55.2	46.0	74.8	52.8	79.2	61.6
	Random _{same}	50.8	48.4	73.6	53.2	75.2	60.2
	Random _{diff}	46.4	46.8	72.8	50.0	70.4	57.3
	Random _{bio}	58.0	48.4	78.4	51.2	73.6	61.9
1000	Relevant	63.6	50.0	75.6	52.0	76.8	63.6
	N/A	60.8	45.2	74.0	53.2	77.2	62.1
	Random _{same}	56.0	49.2	72.0	52.4	74.4	60.8
	Random _{diff}	43.2	40.8	70.4	51.2	69.6	55.0
	Random _{bio}	56.4	48.0	76.0	49.2	74.8	60.9

Table 15: Accuracy (%) of all methods with different random seeds on BBH.

Relevant	N/A	Random _{same}	Random _{diff}	Random _{bio}	ICL _{cs}	ICL _{bio}
70.8	73.4	71.2	72.9	72.6	74.6	74.1

Table 16: Accuracy (%) of different methods on CommonsenseQA. ‘cs’ in ICL_{cs} stands for ‘commonsense’.

Prompt: self-generate **relevant** examples

Your task is to tackle mathematical problems. When presented with a math problem, recall **relevant** problems as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure that your response follows the instructions below.

Analogous Problems:
Offer five diverse examples of math problems that are **relevant or analogous** to the initial problem. For each problem, elaborate on the solution and conclude with the ultimate answer (enclosed in `\boxed{}`). For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate **N/A** examples

Your task is to tackle mathematical problems. When presented with a math problem, recall **n/a** problems as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure that your response follows the instructions below.

N/A Problems:
Offer five diverse examples of math problems that are **n/a** to the initial problem. For each problem, elaborate on the solution and conclude with the ultimate answer (enclosed in `\boxed{}`). For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate **random math** examples

Your task is to tackle mathematical problems. When presented with a math problem, recall **random** problems as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure that your response follows the instructions below.

Random Problems:
Randomly offer five diverse examples of math problems. For each problem, elaborate on the solution and conclude with the ultimate answer (enclosed in `\boxed{}`). For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate **random no-math** examples

Your task is to tackle mathematical problems. When presented with a math problem, recall **random problems (remember not to output math problems)** as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure that your response follows the instructions below.

Random Problems:
Randomly offer five diverse examples of **any type, except math** problems. For each problem, elaborate on the solution and conclude with the ultimate answer (enclosed in `\boxed{}`). For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate **random biological** examples

Your task is to tackle mathematical problems. When presented with a math problem, recall **random biological problems (remember not to output math problems)** as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure that your response follows the instructions below.

Random Problems:
Randomly offer five diverse examples of **biological problems (remember not to output math problems)**. For each problem, elaborate on the solution and conclude with the ultimate answer (enclosed in `\boxed{}`). For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Figure 5: Prompts for different methods on GSM8K.

Prompt: self-generate **relevant** examples

Your task is to tackle mathematical problems. When presented with a math problem, recall **relevant** problems as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure to include all of the following points:

Relevant Problems:
Recall three examples of math problems that are **relevant** to the initial problem. Note that your problems should be distinct from each other and from the initial problem (e.g., involving different numbers and names). For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Say "Let's solve the following math problem." Then formulate your response in the following format:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate **N/A** examples

Your task is to tackle mathematical problems. When presented with a math problem, recall **n/a** problems as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure to include all of the following points:

N/A Problems:
Recall three examples of math problems that are **n/a** to the initial problem. Note that your problems should be distinct from each other and from the initial problem (e.g., involving different numbers and names). For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Say "Let's solve the following math problem." Then formulate your response in the following format:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate **random math** examples

Your task is to tackle mathematical problems. When presented with a math problem, recall **random** problems as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure to include all of the following points:

Random Problems:
Randomly recall three examples of math problems. Note that your problems should be distinct from each other and from the initial problem (e.g., involving different numbers and names). For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Say "Let's solve the following math problem." Then formulate your response in the following format:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate **random no-math** examples

Your task is to tackle mathematical problems. When presented with a math problem, recall **random problems (remember not to output math problems)** as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure to include all of the following points:

Random Problems:
Randomly recall three examples of **any type, except math** problems. Note that your problems should be distinct from each other and from the initial problem. For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Say "Let's solve the following math problem." Then formulate your response in the following format:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate **random biological** examples

Your task is to tackle mathematical problems. When presented with a math problem, recall **random biological problems (remember not to output math problems)** as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure to include all of the following points:

Random Problems:
Randomly recall three examples of **biological problems (remember not to output math problems)**. Note that your problems should be distinct from each other and from the initial problem. For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Say "Let's solve the following math problem." Then formulate your response in the following format:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Figure 6: Prompts for different methods on MATH.

Prompt: self-generate relevant examples

Your task is to tackle reasoning problems. When presented with a problem, recall **relevant** problems as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure to include all of the following points:

Relevant Problems:
Recall three examples of problems that are **relevant** to the initial problem. Note that your problems must be distinct from each other and from the initial problem. For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Say "Let's solve the following reasoning problem." Then formulate your response in the following format:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate N/A examples

Your task is to tackle reasoning problems. When presented with a problem, recall **n/a** problems as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure to include all of the following points:

N/A Problems:
Recall three examples of problems that are **n/a** to the initial problem. Note that your problems must be distinct from each other and from the initial problem. For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Say "Let's solve the following reasoning problem." Then formulate your response in the following format:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate random reasoning examples

Your task is to tackle reasoning problems. When presented with a problem, recall **random** problems as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure to include all of the following points:

Random Problems:
Randomly recall three examples of reasoning problems. Note that your problems must be distinct from each other and from the initial problem. For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Say "Let's solve the following reasoning problem." Then formulate your response in the following format:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate random no-reasoning examples

Your task is to tackle reasoning problems. When presented with a problem, recall **random problems (remember not to output reasoning problems)** as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure to include all of the following points:

Random Problems:
Randomly recall three examples of **any type, except reasoning** problems. Note that your problems must be distinct from each other and from the initial problem. For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Say "Let's solve the following reasoning problem." Then formulate your response in the following format:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Prompt: self-generate random biological examples

Your task is to tackle reasoning problems. When presented with a problem, recall **random biological problems (remember not to output reasoning problems)** as examples. Afterward, proceed to solve the initial problem.

Initial Problem:
[The target problem]

Instructions:
Make sure to include all of the following points:

Random Problems:
Randomly recall three examples of **biological problems (remember not to output reasoning problems)**. Note that your problems must be distinct from each other and from the initial problem. For each problem:
- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Solve the Initial Problem:
Say "Let's solve the following reasoning problem." Then formulate your response in the following format:
Q: Copy and paste the initial problem here.
A: Explain the solution and enclose the ultimate answer in `\boxed{}` here.

Figure 7: Prompts for different methods on BBH.

Input

Randomly recall five examples of math problems. Note that your problems should be distinct from each other (e.g., involving different numbers and names). For each problem:

- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Output

Q: If John has 5 apples and he buys 8 more, how many apples does he have in total?
A: To find the total number of apples John has, we add the number of apples he initially has to the number of apples he buys. So, we calculate $5 + 8 = 13$. Therefore, John has `\boxed{13}` apples in total.

Q: Solve for x in the equation $2x - 4 = 10$.
A: To solve for x , we first add 4 to both sides of the equation to get $2x = 14$. Then, we divide both sides by 2 to isolate x , resulting in $x = 7$. Thus, the solution is `\boxed{x = 7}`.

Q: A rectangle has a length of 10 cm and a width of 4 cm. What is its area?
A: The area of a rectangle is found by multiplying its length by its width. Therefore, the area of this rectangle is $10 * 4 = 40$ square centimeters. So, the area is `\boxed{40}`.

Q: A fair six-sided die is rolled. What is the probability of rolling a number greater than 4?
A: A six-sided die has numbers 1 through 6. The numbers greater than 4 are 5 and 6, which are 2 out of the 6 possible outcomes. Therefore, the probability of rolling a number greater than 4 is $\frac{2}{6} = \frac{1}{3}$. Thus, the probability is `\boxed{\frac{1}{3}}`.

Q: Find the derivative of the function $f(x) = 3x^2 + 2x - 5$.
A: To find the derivative of $f(x)$, we apply the power rule to each term. The derivative of $3x^2$ is $6x$, the derivative of $2x$ is 2, and the derivative of a constant -5 is 0. Therefore, the derivative of $f(x)$ is $f'(x) = 6x + 2$. So, the derivative is `\boxed{6x + 2}`.

Figure 8: Prompts and outputs for generating math problems.

Input

Randomly recall five diverse examples of biological problems (remember not to output math problems). For each problem:

- After "Q: ", describe the problem
- After "A: ", explain the solution and enclose the ultimate answer in `\boxed{}`.

Output

Q: In a certain species of plant, the allele for red flowers (R) is dominant over the allele for white flowers (r). If two heterozygous plants are crossed, what is the probability that an offspring will have white flowers?
A: To solve this problem, we use a Punnett square to determine the genotypes of the offspring. The cross of two heterozygous plants $Rr \times Rr$ produces the following genotypes: $RR, Rr, Rr,$ and rr . The probability of an offspring having white flowers rr is 1 out of 4, since only one genotype out of the four possible genotypes results in white flowers. Therefore, the probability is `\boxed{\frac{1}{4}}`.

Q: A small pond is introduced with 10 fish, and the fish population doubles every month. How many fish will there be in the pond after 6 months?
A: To find the fish population after 6 months, we use the formula for exponential growth: $P = P_0 * 2^n$, where P is the final population, P_0 is the initial population, and n is the number of time periods (months, in this case). Plugging in the values, we get $P = 10 * 2^6$. Calculating this gives $P = 10 * 64 = 640$. Therefore, there will be `\boxed{640}` fish in the pond after 6 months.

Q: What is the primary mechanism by which the human kidney concentrates urine?
A: The primary mechanism by which the human kidney concentrates urine is through the counter-current multiplier system in the loop of Henle. This system allows for the reabsorption of water and solutes from the filtrate back into the bloodstream, which concentrates the urine. The key to this process is the creation of a high osmolarity in the medullary interstitial fluid, which draws water out of the filtrate in the collecting ducts, concentrating the urine. The ultimate answer is the `\boxed{\text{counter-current multiplier system}}`.

Q: What is the principle behind the theory of natural selection as proposed by Charles Darwin?
A: The principle behind the theory of natural selection is that individuals within a species show variation in physical characteristics and behaviors. These variations can affect the individuals' ability to survive and reproduce in their environment. Individuals with traits that are better suited to their environment are more likely to survive and reproduce, passing on those advantageous traits to their offspring. Over time, this can lead to changes in the species as a whole. The ultimate answer is the `\boxed{\text{survival and reproduction of the fittest}}`.

Q: How do antibiotics kill bacteria?
A: Antibiotics can kill bacteria through several mechanisms, including inhibiting cell wall synthesis, disrupting protein synthesis, impairing nucleic acid synthesis, and interfering with metabolic pathways. The specific mechanism depends on the class of antibiotic. For example, penicillin disrupts the synthesis of the bacterial cell wall, leading to cell lysis and death. Therefore, the ultimate answer is by `\boxed{\text{inhibiting vital processes}}` specific to bacteria.

Figure 9: Prompts and outputs for generating biological problems.