
# CS2916 Spring 2024 Project
# In-context Learning with Errors

**Jianhao Huang, Yucheng Wang, Qihao Luo**
Shanghai Jiao Tong University

## Abstract

In-context learning (ICL) is a paradigm for large language models (LLMs) to learn to perform a new task without updating its parameters. This study investigates the capabilities of LLMs in in-context learning with errors, focusing on mathematical reasoning. Using the GSM8K and MATH datasets and a variety of prompt modifications, we explore how LLMs manage errors and modifications in demonstration steps. Our findings suggest that certain types of errors and reordering of demonstration components have minimal impact on the solve rates, thereby providing insights into the adaptability and error tolerance of in-context learning models. This research extends our understanding of the flexibility of transformers in maintaining performance despite modifications, positioning it as a step forward in the practical application of in-context learning methods in AI.

## 1 Introduction

Transformers have emerged as the foundational architectures in many domains, including language processing, computer vision and etc. Recently, LLMs based on transformers have exhibited remarkable in-context learning capabilities, where the model can solve a new task solely through inference based on prompts of the task without further fine-tuning.

Specifically, an in-context trained model over a function class $\mathcal{F}$ can accurately predict the function value $f(x_{\text{query}})$ for most $f \in \mathcal{F}$ by using a prompt sequence consisted of input-label pairs along with the query token $(x_1, f(x_1), \cdots, x_N, f(x_N), x_{\text{query}})$.

Built on this theoretically amenable setting, many follow-up works explored empirical and theoretical properties of in-context learning of transformers from different respectives such as expressive power, generalization, etc. However, the following problem still remains largely open:

*How do transformers learn in-context with error?*

In this paper, we focus on math problems to empirically study how transformers learn in-context with error. We highlight our contributions as follows.

**Our Contributions.**

- We first infer the results of math problems through in-context learning and study how changes in demonstrations affect the results. We find that direct demonstrations give the worst solve rate of $9.63\%$ while the highest solve rate of $46.1\%$ is given by reordering examples.
- We then study to which extent we can modify the demonstrations while the solve rate remains essentially unchanged. To our surprise, removing the first half and reordering the remains without the last step gives a similar solve rate comparing to in-context learning through original demonstrations.

## 2 In-Context Learning with Errors

### 2.1 Basic Study

We first conduct in-context learning with standard demonstrations, documenting the solve rate for mathematical reasoning problems and comparing it with scenarios devoid of examples to demonstrate the effectiveness of ICL. Subsequently, we modify the solutions within the in-context examples and analyzed their impact on the results.

Our selection of modifications is primarily grounded in a basic analysis of the demonstrations. A demonstration can be deconstructed into the following hierarchical layers:

- **Reasoning Steps**: Several reasoning steps forming a chain of thought, guiding the LLM to decompose complex mathematical problems into manageable steps.
- **Prompts**: Restatements regarding the problem, prior steps, and the scenario to enhance the LLM's comprehension of the issue in each reasoning step.
- **Calculations**: Specific mathematical computations.

From these three levels, we explore the impact of various independent modifications on the results. For instance, does altering the sequence of reasoning steps affect the efficacy of ICL? Would the alteration of restatements in prompts or computational errors decrease the model's solve rate?

Furthermore, related researches provide insights into our methods of modifications. Some studies (Tang et al. (2023), Madaan et al. (2023)) focus on whether models are symbolic reasoners, or if they prioritize the symbolic over the semantic aspects of demonstrations. Accordingly, we experiment with replacing numbers in demonstration solutions with Greek letters to compare the effects. Other studies suggest that the sequence and complexity of examples can also impact results (Fu et al. (2023), Lu et al. (2022)), leading us to make relevant modifications.

Ultimately, we make 11 different types of modifications (Prompt Types), including various adjustments to and errors in solutions. These modifications and their corresponding results are listed and analyzed in section 3.

### 2.2 Extension

In this section, we expand the experimental methods from the previous section to new datasets and models. We further investigate the maximum tolerance for errors using the ICL approach. Specifically, we seek to determine how much we could alter an example, given a fixed model and dataset, while still maintaining its performance (solve rate).

It's clear that these modifications cannot be arbitrary. There must be specific constraints on how changes are made; otherwise, it would be possible to manually craft a better answer, which would be meaningless. The core purpose of our modifications is to identify which pieces of information are relatively unnecessary for ICL, thus our starting point in selecting the methods of modifications is to reduce the information provided by demonstrations.

Therefore, we identified three independent directions for modification:

- **Deletion**: Direct removal of certain reasoning steps.
- **Shuffle**: Shuffling the sequence of some reasoning steps.
- **Random Numbers**: Adjusting the numbers used in calculations, such as introducing computational errors.

We refrain from making more detailed changes (such as semantic or stylistic alterations) because we want our modification method to be simpler, not requiring manual intervention, and thus more universally applicable. This approach clarifies which pieces of information are redundant for the ICL method. Moreover, deletion has practical significance as it can speed up inference processes.

We conduct these modifications sequentially. First, we examine different methods of deletion. Then we test various shuffle strategies on the most effective deletion methods, and finally, we adjust the numbers randomly in the best-performing demonstration from the previous step. Ultimately, we develop a combined modification scheme that maintained accuracy relatively well (within a $10\%$ margin).

We conduct experiments with the following eight types of prompts:

- **Gold (G)**: The original example.
- **No (N)**: No use of ICL method.
- **First (F)**: Removed the first half of the reasoning steps in each example.
- **Second (S)**: Removed the second half of the reasoning steps in each example.
- **Middle (M)**: Removed the middle half of the reasoning steps in each example.
- **F-re-all**: On the basis of **First**, shuffled the order of all reasoning steps.
- **F-re-ex**: On the basis of **First**, shuffled the order of reasoning steps except the final step.
- **F-re-ex-err**: Introduced computational errors to **F-re-ex**.

From the analysis of these results, we conclude that different reasoning steps have varying levels of tolerance for deletion and reordering. Some steps allow for deletion or reordering while maintaining the solve rate, whereas others do not. We also observe that certain computational errors do not significantly impact the solve rate. This insight deepens our understanding of the role of demonstrations in ICL, highlighting their variable sensitivity to modifications and the robustness of certain information within the context provided.

## 3 Experiments

### 3.1 Experiment Setup

In the first part of our study, we select the GSM8K dataset (Cobbe et al. (2021)). Our model of choice is Mistral-7B. We choose four problems from the GSM8K training set as standard demonstrations and assess the model's performance using the GSM8K test set, which contains 1319 problems, to evaluate the effects of in-context learning. We conduct a single evaluation for each modification method, with solve rate as the performance metric.

In the second part of our experiments, we expand to the new MATH dataset (Hendrycks et al. (2021)) and the LLAMA-3-8B model. The MATH dataset is specifically designed to evaluate the mathematical abilities of automated systems across various topics such as algebra, calculus, and statistics. We focus our experiments exclusively on the algebra section of the MATH dataset, as we observe that in-context learning do not significantly help with other sections, such as geometry. We select four problems from the MATH training set as standard demonstrations and use the MATH test set, consisting of 1187 problems, to assess the model's performance under ICL conditions. To ensure the reliability of our experimental methods, we repeat these steps three times and calculated the average solve rate as our main metric. Additionally, we employ the RoughL F-measure to evaluate the similarity between the final modified demonstrations and the original examples.

### 3.2 Results

Table 1 shows some demonstrations and results for the Mistral-7B on the GSM8K dataset with different prompt types. We analyze these results in detail.

Overall, it is evident that in-context learning significantly aids mathematical reasoning (**Original** compared to **Direct**). Next, we discuss the impact of modifications on results from the perspectives of reasoning steps, prompts, and calculations.

**Reasoning Steps**: The impact of reasoning steps on in-context learning is significant. Demonstrations devoid of reasoning steps (**Direct**) achieved the lowest solve rate at $9.63\%$. Disordering the sequence of reasoning steps (**Shuffle**) also had a noticeable effect, reducing solve rates from $44.3\%$ to $36.4\%$.

**Prompts**: The results from **CalculationOnly** indicate that the absence of prompts considerably reduces the effectiveness of in-context learning ($44.3\%$ to $27.5\%$). However, adjustments to non-mathematical parts of the prompts (**WrongName**) do not significantly affect outcomes.

**Calculations**: The results of **NoProcess** and **NoResults** highlight that the completeness of the calculation process profoundly impacts in-context learning, which is an intriguing finding. Nonetheless, while ensuring complete calculations, introducing some computational errors (**Miscalculation**) has a minimal effect ($44.3\%$ to $41.9\%$).

Table 1: Results for Mistral-7B on GSM8K with different prompt types

| **Question:** Mr. Sam shared a certain amount of money between his two sons, Ken and Tony. If Ken got $1750, and Tony got twice as much as Ken, how much was the money shared? | | |
|---|---|---|
| **Prompt Type** | **Demonstration** | **Solve Rate** |
| **Original** | Tony got twice 1750 which is $2 \times 1750 = 3500$. The total amount shared was $1750 + 3500 = 5250$. | 44.3% |
| **Direct** | The total amount shared was 5250. | 9.63% |
| **CalculationOnly** | $2 \times 1750 = 3500$. $1750 + 3500 = 5250$. | 27.5% |
| **NoProcess** | Tony got twice 1750 which is 3500. The total amount shared was 5250. | 31.9% |
| **NoResults** | Tony got twice 1750 which is $2 \times 1750 =$. The total amount shared was $1750 + 3500 =$. | 27.6% |
| **Abstract** | Tony got twice $\alpha$ which is $2 \times \alpha = \beta$. The total amount shared was $\alpha + \beta = \delta$. | 27.1% |
| **Miscalculation** | Tony got twice 1750 which is $2 \times 1750 = 3000$. The total amount shared was $1750 + 3000 = 4750$. | 41.9% |
| **WrongName** | John got twice 1750 which is $2 \times 1750 = 3500$. The total amount shared was $1750 + 3500 = 5250$. | 41.9% |
| **Shuffle** | The total amount shared was $1750 + 3500 = 5250$. Tony got twice 1750 which is $2 \times 1750 = 3500$. | 36.4% |
| **Simpler** | *(Replaced with some simpler examples.)* | 35.9% |
| **ReorderExamples** | *(Changed the order of the examples.)* | 46.1% |
| **Mismatch** | *(Each example got the answer to another question.)* | 10.8% |

Furthermore, we observe that adjusting the order and complexity of examples affects the effectiveness of in-context learning, aligning with findings from the original paper (Fu et al. (2023), Lu et al. (2022)). We also find that replacing numbers in examples with Greek letters (**Abstract**) impairs the model's ability to learn effectively from demonstrations, consistent with the conclusion in the paper that Large Language Models are In-Context Semantic Reasoners rather than Symbolic Reasoners (Tang et al. (2023)).

These results prompt us to reflect on how LLMs learn from demonstrations. We believe that demonstrations provide a paradigm for problem-solving to LLMs, where the integrity and consistency of the paradigm are crucial to ICL, whereas correctness has less impact. For instance, incomplete calculation processes (**NoProcess** and **NoResults**) might lead LLMs to produce similarly incomplete calculations, resulting in lower solve rates, while inconsistent restatements (**Mismatch**) might confuse the model's understanding of the problem, leading to poorer outcomes (44.3% to 10.8%).

In further research, we expand to a new dataset, MATH, and a new model, LLAMA-3-8B. Our focus was on the maximum tolerance of the ICL method to errors, that is, how much we could alter examples while still maintaining the performance.

Our first approach involve removing some of the reasoning steps from the examples. We aim to identify which portions of information are relatively superfluous for ICL, thus considering three types of deletions: removing the first half of the reasoning steps from each example (**First**), removing the second half (**Second**), and removing the middle half (**Middle**). The final results are depicted in Figure 1.

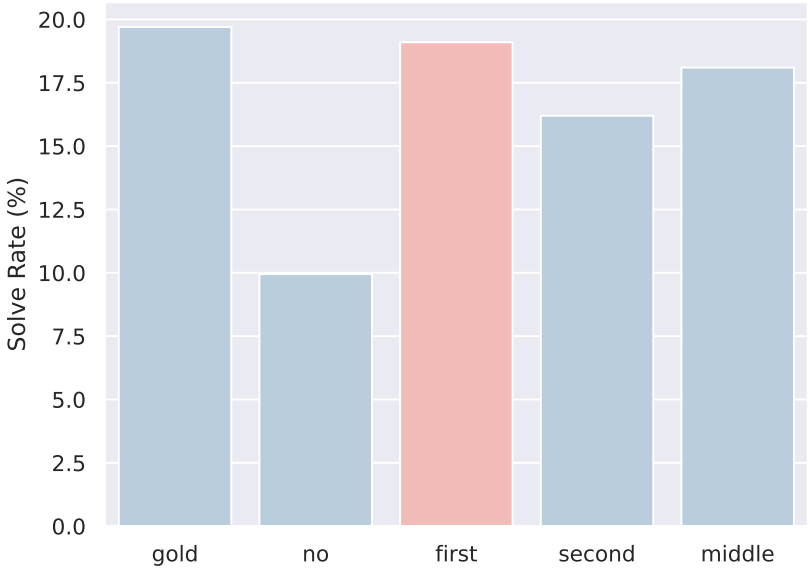


Figure 1: Results for different types of deletions.

We find that the results from the **First** setup nearly maintain the performance of the original examples (from $19.7\%$ to $19.1\%$). In contrast, removing the second half of the reasoning steps significantly impact performance (from $19.7\%$ to $16.2\%$).

Building on the best results from the previous step, we further experiment by shuffling the order of all reasoning steps on the **First** basis (**F-re-all**) and preserving only the order of the last step while shuffling the others (**F-re-ex**). Lastly, we introduce computational errors to the better outcome among these (**F-re-ex-err**). The results are shown in Figure 2.

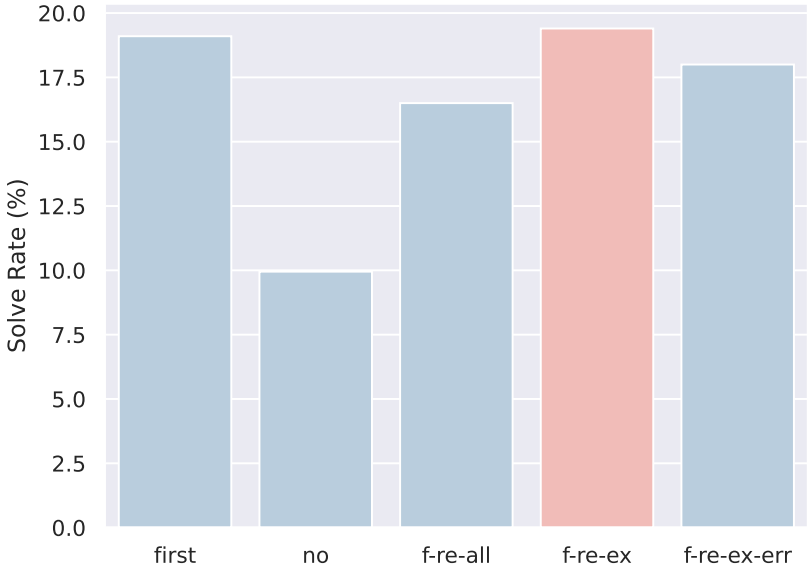


Figure 2: Results for different shuffle strategies and introducing computational errors.

We discover that if the position of the last step is maintained, scrambling the order of the other steps does not significantly affect the solve rate (from $19.7\%$ to $19.4\%$). Additionally, introducing computational errors do not substantially decrease the solve rate.

The final demonstration we obtain is substantially altered — it lacks half of the reasoning steps, has scrambled logic, contains errors in calculations, and is approximately half the length of the original version. The Rouge-L F-measure between the original demonstration and the altered version is only $0.61$. However, the model is still able to extract useful information and maintain accuracy (from $19.7\%$ to $18.0\%$).

Our results highlight the significant role of the latter part of demonstrations, especially the final step, in mathematical reasoning within in-context learning. This may be due to several reasons: (i) The last reasoning step is closely linked to the final ANSWERKEY provided by the model. The consistency from the last step to the ANSWERKEY seems to be more crucial than consistency in restatements. (ii) The final reasoning step often contains summarizing phrases like "In summary," making its position critical.

## 4 Conclusion

In this paper, we studied how well LLMs can handle errors during in-context learning, with a focus on mathematical problem-solving. The experiments conducted with various modifications to the prompt types on the GSM8K and MATH datasets reveal that LLMs display significant resilience to errors and alterations in the problem-solving demonstrations. We observed that the deletion of reasoning steps and the introduction of computational errors often did not drastically reduce performance, suggesting a robustness in the learning process that can adapt to incomplete or incorrect information. Our study also highlighted the importance of the final reasoning steps in maintaining higher solve rates, indicating a potential area for optimizing in-context learning setups.

Since our study is limited to solving mathematical problems, its results may not be applicable to another field. Future research can explore other (more general) aspects of the prompts in in-context learning such as language style and vocabulary, and further refine the in-context learning methodologies in order to generalize them to AI domains other than mathematics.

## References

[1] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[2] Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning, 2023.

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

[4] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556. URL https://aclanthology.org/2022.acl-long.556.

[5] Aman Madaan, Katherine Hermann, and Amir Yazdanbakhsh. What makes chain-of-thought prompting effective? a counterfactual study. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1448–1535, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.101. URL https://aclanthology.org/2023.findings-emnlp.101.

[6] Xiaojuan Tang, Zilong Zheng, Jiaqi Li, Fanxu Meng, Song-Chun Zhu, Yitao Liang, and Muhan Zhang. Large language models are in-context semantic reasoners rather than symbolic reasoners, 2023.