

What Makes In-context Learning Effective for Mathematical Reasoning

Jiayu Liu¹ Zhenya Huang^{1,2} Chaokun Wang¹ Xunpeng Huang³ ChengXiang Zhai⁴ Enhong Chen¹

Abstract

Owing to the capability of in-context learning, large language models (LLMs) have shown impressive performance across diverse mathematical reasoning benchmarks. However, we find that few-shot demonstrations can sometimes bring negative performance and their effectiveness on LLMs’ reasoning abilities remains unreliable. To this end, in this paper, we aim to theoretically analyze the impact of in-context demonstrations on LLMs’ reasoning performance. We prove that the reasoning efficacy (measured by empirical prediction loss) can be bounded by an *LLM-oriented semantic similarity* and an *inference stability of demonstrations*, which is general for both one-shot and few-shot scenarios. Based on this finding, we propose a straightforward, generalizable, and low-complexity demonstration selection method named LMS3. It facilitates to select the most pertinent samples for different LLMs and includes a novel demonstration rejection mechanism to automatically filter out samples that are unsuitable for few-shot learning. Through experiments on three representative benchmarks, two LLM backbones, and multiple few-shot settings, we verify that our LMS3 has superiority and achieves consistent improvements on all datasets, which existing methods have been unable to accomplish. Our code is available at <https://github.com/Ljyustc/LMS3>.

1. Introduction

Mathematical reasoning serves as a milestone in assessing the progress of natural language processing (Zhang et al.,

¹State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China ²Institute of Artificial Intelligence, Hefei Comprehensive National Science Center ³Hong Kong University of Science and Technology ⁴Department of Computer Science, University of Illinois at Urbana-Champaign. Correspondence to: Zhenya Huang <huangzhy@ustc.edu.cn>.

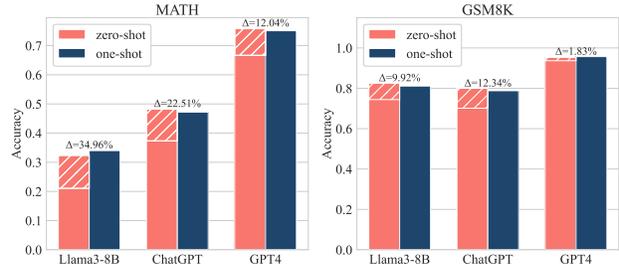


Figure 1. Problem-solving Accuracy of zero-shot and one-shot settings. The hatched areas represent that in the one-shot setting, the model answers incorrectly Δ proportion of problems that are answered correctly in the zero-shot setting.

2020; Liu et al., 2023). Currently, many large language models (LLMs) have exhibited strong performance across various mathematical benchmarks (Hendrycks et al., 2021; Cobbe et al., 2021). A key capability of these LLMs is in-context learning (ICL) (Dong et al., 2022), which enables them to learn from a few examples to implement specific logical structures (Wei et al., 2022) or utilize codes (Chen et al., 2023) to improve reasoning accuracy. Based on this ability, they have adeptly addressed problems across different types and difficulty, ranging from elementary word problems to college-level algebra (Brown et al., 2020; Achiam et al., 2023; Xue et al., 2024; Liu et al., 2025).

However, it remains an unresolved issue whether in-context learning truly enhances LLMs’ mathematical reasoning abilities. To show this phenomenon, in Figure 1, we present the zero-shot and one-shot results of Llama3-8B (Meta, 2024), ChatGPT, and GPT-4 (Achiam et al., 2023) on two representative benchmarks MATH (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021). Surprisingly, we find that 1) When given an example, their problem-solving accuracy does not consistently improve, and sometimes even declines (e.g., ChatGPT on MATH dataset). 2) Further analysis reveals that in the one-shot setting, these LLMs even fail in 1.83%-34.96% of problems (marked with white hatching) that they have correctly solved in the zero-shot scenario. This raises an important research question: *Is in-context learning always effective for mathematical reasoning, and under what conditions does it work?*

To address this issue, existing literature primarily analyzes the impact of demonstrations from an empirical perspective.

For instance, researchers have revealed several important factors, including the similarity to test samples (Liu et al., 2022), the diversity (Gao & Das, 2024), complexity (An et al., 2023) and perplexity (Sorensen et al., 2022) of demonstrations, as well as the Inference LLM used (Peng et al., 2024; Ye et al., 2023). Regarding the theoretical foundations of ICL, existing work has tried to explain the introduction of demonstrations as a form of meta-gradient optimization (Dai et al., 2023), kernel regression (Han et al., 2023), and token reinforcement (Yan et al., 2024). However, these studies 1) have not provided precise quantification of the impact of demonstrations on LLMs’ reasoning performance, nor 2) have they offered theoretical conclusions on when demonstrations are beneficial.

In this paper, we first theoretically analyze the impact of a demonstration on the reasoning performance in one-shot scenario. We find that a sufficient condition for one-shot to outperform zero-shot is that 1) *the demonstration and test sample have similar semantics encoded by the inference LLM* and that 2) *the inference LLM exhibits sufficient stability in reasoning the answer of the demonstration itself*. The former goes beyond traditional methods that rely solely on the semantic similarity between demonstrations and test samples, pointing out the critical role of the inference LLM’s encoding parameters, while also being generalizable to these methods. The latter introduces the concept and measurement of *inference stability of demonstrations* for the first time. It should be emphasized that our theory is general and we further extend it to the k -shot scenario.

Based on our theory, we propose a simple yet effective demonstration selection method, named **LMS3**, to balance the *LLM-oriented Semantic Similarity* and *inference Stability of demonstrations*, allowing for the automatic selection of approximately optimal samples tailored to different LLMs. Additionally, to ensure that the sufficient condition of our theories is essentially satisfied, we introduce an innovative demonstration rejection mechanism that can adaptively identify when few-shot learning should *not* be used, which is the first attempt in the field. Our method has strong theoretical advantages, generalization ability, and low complexity. Experiments on three benchmarks demonstrate its consistent improvements in both one-shot and few-shot scenarios. The contributions of this paper are as follows:

- We theoretically quantify the effect of demonstrations on ICL reasoning performance under one/few-shot settings. We prove that it can be bounded by *LLM-oriented semantic similarity* and *inference stability of demonstrations*.
- We propose a novel demonstration selection method, LMS3, which can generalize to various existing methods and offer better scalability and complexity.

- We validate our method on three mathematical benchmarks using multiple LLMs as backbones, demonstrating improvements in problem-solving accuracy, generalization ability, and interpretability.
- We further test our method on commonsense reasoning task, validating its applicability to more general reasoning tasks.

2. Related Work

Mathematical Reasoning. Mathematical reasoning is a critical benchmark for assessing the level of artificial intelligence (Zhang et al., 2020; Liu et al., 2023). Early research in this area mainly focused on rule-based, template-based, and statistical machine learning methods for simple math word problems (Feigenbaum et al., 1963; Fletcher, 1985). With the development of large language models (LLMs), recent efforts have primarily explored two directions. On one hand, some approaches equip LLMs with chain-like (Kojima et al., 2022), tree-like (Yao et al., 2024), or graph-like (Besta et al., 2024) reasoning processes, or require LLMs to use codes (Chen et al., 2023; Gao et al., 2023) and tools (Ma et al., 2025) to address potential numerical computation errors. On the other hand, there are efforts that provide the model with certain examples in the prompts through retrieval-augmented generation (Wei et al., 2022; Asai et al., 2024), allowing the model to solve problems based on similar approaches using its contextual learning abilities.

In-context Learning. In-context Learning (ICL) focuses on making LLMs learn and reason based on existing examples (Dong et al., 2022). Its advantage lies in the adaptability and flexibility for different tasks and scenarios (Meade et al., 2023). However, the selection of examples remains a central challenge, where current researches have developed supervised methods and unsupervised methods. This paper focuses on unsupervised methods, which can be grouped into three main categories. The first and currently most prominent method is called Similar-ICL (Liu et al., 2022; Luo et al., 2023; Zhang et al., 2023; Fu et al., 2022), which aims to find examples with closest semantic representations to the test sample. The semantic representation approaches include TF-IDF, BM25 (Robertson et al., 2009), T5 encoding (Raffel et al., 2020), BGE-M3 (Chen et al., 2024b), OpenAI embedding, etc. The second line of methods calculates the impact of each demonstration on the test sample (Peng et al., 2024). Impact calculation approaches include influence function (Van et al., 2024; Chang & Jia, 2023), mutual information (Sorensen et al., 2022), perplexity (Gonen et al., 2023), code-length (Wu et al., 2023), etc. The third category uses the feedback from LLMs to dynamically select demonstrations (Nguyen & Wong, 2023; Qin et al., 2023).

Explanation of ICL. Regarding the underlying mechanisms

of ICL, most existing research explored the impact of empirical factors such as the number of examples, gold labels, diversity, and types of LLMs from an experimental perspective (Pan et al., 2023; Peng et al., 2024; Min et al., 2022). Some theoretical explorations explain ICL from perspectives including meta-gradient updates (Dai et al., 2023), kernel regression (Han et al., 2023), and token reinforcement (Yan et al., 2024). In comparison, to the best of our knowledge, we are the first to theoretically quantify the impact of demonstrations on reasoning performance and identify when they are effective.

3. Theoretical Analysis

Notations. In in-context learning (ICL) setup, we have a demonstration pool \mathcal{D} and a test set \mathcal{D}_{test} , which contain \mathcal{M} and \mathcal{N} mathematical problems respectively. The k -shot ICL is formulated as appending k demonstrations $\{(X_1, y_1), \dots, (X_k, y_k)\} \subseteq \mathcal{D}$ with the test data $X_{test} \in \mathcal{D}_{test}$ in prompt to reason the solution

$$\hat{y}_{test} \stackrel{def}{=} LLM((X_1, y_1), \dots, (X_k, y_k), X_{test}), \quad (1)$$

where X_i, X_{test} represent the problem context and y_i represents the labeled solution. The prediction loss on X_{test} is denoted as $L(X_{test}, y_{test})$. In the following, we omit symbol y and use X to express each demonstration for brevity.

To evaluate the influence of a demonstration X on inferring the answer of X_{test} , we use $\mathbf{h}, \mathbf{h}_{test} \in \mathbb{R}^d$ to denote the representation of problem X and X_{test} . Then, the Transformer attention mechanism in ICL setting is denoted as:

$$\begin{aligned} \mathcal{F}_{ICL}(\mathbf{h}_{test}) &= \text{Attn}(V, K, Q, \mathbf{h}_{test}) \\ &= W_V[\mathbf{h}, \mathbf{h}_{test}] \cdot \text{softmax} \left(\frac{(W_K[\mathbf{h}, \mathbf{h}_{test}])^T \cdot W_Q \mathbf{h}_{test}}{\sqrt{d}} \right), \end{aligned} \quad (2)$$

where W_Q, W_K, W_V are the projection matrices for computing the attention queries, keys, and values, respectively. Without loss of generality, we omit W_Q in $\mathcal{F}_{ICL}(\mathbf{h}_{test})$ because we can redefine $W_K = W_K^T \cdot W_Q$. As a result, we only keep $W_K \in \mathbb{R}^{d \times d}, W_V \in \mathbb{R}^{d' \times d}$ in our setting, where d' is the output dimension of layer \mathcal{F}_{ICL} . Following a common approach (Dai et al., 2023), we approximate the attention to a simplified linear attention by:

$$\begin{aligned} \mathcal{F}_{ICL}(\mathbf{h}_{test}) &\approx W_V[\mathbf{h}, \mathbf{h}_{test}] \cdot \left(\frac{(W_K[\mathbf{h}, \mathbf{h}_{test}])^T \cdot \mathbf{h}_{test}}{\sqrt{d}} \right) \\ &= \frac{W_V}{\sqrt{d}} \mathbf{h}_{test} \cdot (W_K \mathbf{h}_{test})^T \cdot \mathbf{h}_{test} + \frac{W_V}{\sqrt{d}} \mathbf{h} \cdot (W_K \mathbf{h})^T \cdot \mathbf{h}_{test} \end{aligned} \quad (3)$$

Analogy to Linear Optimization. We start our analysis of Eq. (3) by considering a linear function $\mathcal{F}(z) \stackrel{def}{=} W \cdot z, W \in \mathbb{R}^{d' \times d}, z \in \mathbb{R}^d$. Specifically, given $\mathcal{F}(z)$ with an initialized parameters W_0 , assume we have a training data $z_0 \in \mathbb{R}^d$, then the gradient of loss $L(\mathcal{F})$ can be written as $\Delta W = \nabla_{\mathcal{F}} L(z_0, W_0) \cdot z_0^T$. Applying the gradient to

parameter optimization, the prediction of a test sample \mathbf{h}_{test} is $\mathcal{F}(\mathbf{h}_{test}) = W_0 \cdot \mathbf{h}_{test} + \nabla_{\mathcal{F}} L(z_0, W_0) \cdot z_0^T \cdot \mathbf{h}_{test}$.

Based on this idea, Eq. (3) can be interpreted as: 1) We have a linear function $\mathcal{F}(z)$ with initialized parameters

$$W_0 = \frac{W_V}{\sqrt{d}} \mathbf{h}_{test} \cdot (W_K \mathbf{h}_{test})^T. \quad (4)$$

2) We introduce a training data $z_0 = W_K \mathbf{h}$ to optimize the parameters, with the gradient at (z_0, W_0) satisfying:

$$\nabla_{\mathcal{F}} L(z_0, W_0) = \frac{W_V}{\sqrt{d}} \mathbf{h}. \quad (5)$$

3) We finally apply the optimized parameters to calculate the result of test data $\mathbf{h}_{test} \in \mathcal{D}_{test}$.

Under this setting, we aim to estimate the influence of the data $z_0 = W_K \mathbf{h}$ (corresponds to demonstration $X \in \mathcal{D}$) on loss $L(\mathcal{F}(\mathbf{h}_{test}))$. Before detailed derivation, we first give three mathematical annotations:

$$\begin{aligned} \hat{W} &\stackrel{def}{=} \underset{W}{\operatorname{argmin}} \frac{1}{|\mathcal{D}_{pre}|} \sum_{z \in \mathcal{D}_{pre}} L(\mathcal{F}(z)) \\ \hat{W}_{\epsilon, z_0} &\stackrel{def}{=} \underset{W}{\operatorname{argmin}} \frac{1}{|\mathcal{D}_{pre}|} \sum_{z \in \mathcal{D}_{pre}} L(\mathcal{F}(z)) + \epsilon \cdot L(\mathcal{F}(z_0))^1 \\ H_{\hat{W}} &\stackrel{def}{=} \frac{1}{|\mathcal{D}_{pre}|} \sum_{z \in \mathcal{D}_{pre}} \nabla_W^2 L(z, \hat{W}), \end{aligned} \quad (6)$$

where \mathcal{D}_{pre} is the data for pre-training an LLM, and $H_{\hat{W}}$ is the Hessian matrix which is positive definite by assumption (Van et al., 2024). It is worth noting that the pre-trained parameters \hat{W} are actually the initialized parameters in our above setting, i.e., $\hat{W} = W_0$. Taking $\epsilon = \frac{1}{|\mathcal{D}_{pre}|}$, the testing loss on \mathbf{h}_{test} is represented as $L(\mathbf{h}_{test}, \hat{W}_{\frac{1}{|\mathcal{D}_{pre}|}, z_0})$. On this basis, we derive the following theorem:

Theorem 1. Assume $\nabla_{\mathcal{F}} L$ is Lipschitz continuous w.r.t \mathcal{F} with constant μ . If inequality (7) holds true, then $L(\mathbf{h}_{test}, \hat{W}_{\frac{1}{|\mathcal{D}_{pre}|}, z_0}) < L(\mathbf{h}_{test}, \hat{W}_{0, z_0})$, i.e., introducing the training sample z_0 (i.e., demonstration X) can reduce the testing loss on \mathbf{h}_{test} . $\frac{1}{\lambda_{dd'}}$, $\frac{1}{\lambda_1}$ are the largest and smallest eigenvalues of $H_{\hat{W}}$, respectively.

$$\begin{aligned} \frac{\lambda_{dd'}}{\lambda_1} \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| &> \|\mathbf{h}_{test} - z_0\| \cdot (\|\frac{W_V}{\sqrt{d}} \mathbf{h}\| + \mu C_1) \\ C_1 &= \|\frac{W_V}{\sqrt{d}} \mathbf{h}_{test}\| \cdot \|W_K \mathbf{h}_{test}\| \cdot \|\mathbf{h}_{test}\| \end{aligned} \quad (7)$$

We refer the readers to Appendix A for the detailed proof, and present the sketch here.

Proof. With $\hat{W}, \hat{W}_{\epsilon, z_0}$, the influence of upweighting z_0 on

¹Please note that \hat{W}_{ϵ, z_0} is a conceptual and intermediate tool for theoretical analysis, rather than actually training LLMs on both \mathcal{D}_{pre} and the demonstration z_0 .

the empirical loss is (Ling, 1984; Koh & Liang, 2017):

$$\begin{aligned} \mathcal{L}_{loss}(z) &= \left. \frac{dL(\mathbf{h}_{test}, \hat{W}_{\epsilon, z_0})}{d\epsilon} \right|_{\epsilon=0} \\ &= -\nabla_W L(\mathbf{h}_{test}, \hat{W})^T \cdot H_{\hat{W}}^{-1} \nabla_W L(z_0, \hat{W}) \end{aligned} \quad (8)$$

Then, the testing loss $L(\mathbf{h}_{test}, \hat{W}_{\frac{1}{|\mathcal{D}_{pre}|}, z_0})$ can be evaluated by Taylor approximation since $\frac{1}{|\mathcal{D}_{pre}|}$ is sufficiently small, i.e., $L(\mathbf{h}_{test}, \hat{W}_{\frac{1}{|\mathcal{D}_{pre}|}, z_0}) \approx$

$$\begin{aligned} &L(\mathbf{h}_{test}, \hat{W}_{0, z_0}) + \frac{1}{|\mathcal{D}_{pre}|} \left. \frac{dL(\mathbf{h}_{test}, \hat{W}_{\epsilon, z_0})}{d\epsilon} \right|_{\epsilon=0} \\ &= L(\mathbf{h}_{test}, \hat{W}_{0, z_0}) - \frac{1}{|\mathcal{D}_{pre}|} \nabla_W L(\mathbf{h}_{test}, \hat{W})^T \\ &\quad \cdot H_{\hat{W}}^{-1} \nabla_W L(z_0, \hat{W}) \end{aligned} \quad (9)$$

Therefore, now the question turns to evaluate

$$\begin{aligned} L_1 &\stackrel{def}{=} \nabla_W L(\mathbf{h}_{test}, \hat{W})^T \cdot H_{\hat{W}}^{-1} \nabla_W L(z_0, \hat{W}) = \\ &\underbrace{(\nabla_W L(z_0, \hat{W}) - \nabla_W L(\mathbf{h}_{test}, \hat{W}))^T \cdot H_{\hat{W}}^{-1} \nabla_W L(\mathbf{h}_{test}, \hat{W})}_{L_{11}} \\ &+ \underbrace{\nabla_W L(\mathbf{h}_{test}, \hat{W})^T \cdot H_{\hat{W}}^{-1} \nabla_W L(\mathbf{h}_{test}, \hat{W})}_{L_{12}} \end{aligned} \quad (10)$$

Since $H_{\hat{W}}$ is positive definite, we denote $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{dd'} > 0$ are the eigenvalues of $H_{\hat{W}}^{-1}$ and can prove that

$$\begin{aligned} L_{11} &\geq -\lambda_1 \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| \cdot \left(\|\nabla_{\mathcal{F}} L(\mathbf{h}_{test}, \hat{W})\| \right. \\ &\quad \left. - \|\nabla_{\mathcal{F}} L(z_0, \hat{W})\| \cdot \|\mathbf{h}_{test}\| + \|\nabla_{\mathcal{F}} L(z_0, \hat{W})\| \cdot \|\mathbf{h}_{test} - z_0\| \right) \end{aligned} \quad (11)$$

Since $\nabla_{\mathcal{F}} L$ is Lipschitz continuous, we get $L_{11} \geq$

$$\begin{aligned} &-\lambda_1 \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| \cdot (\mu \|\hat{W}(\mathbf{h}_{test} - z_0)\| \cdot \|\mathbf{h}_{test}\| + \\ &\quad \|\nabla_{\mathcal{F}} L(z_0, \hat{W})\| \cdot \|\mathbf{h}_{test} - z_0\|) \end{aligned} \quad (12)$$

Applying Eqs. (4) (5) to Eq. (12), we have:

$$\|\hat{W}(\mathbf{h}_{test} - z_0)\| \leq \left\| \frac{W_V}{\sqrt{d}} \mathbf{h}_{test} \right\| \cdot \|W_K \mathbf{h}_{test}\| \cdot \|\mathbf{h}_{test} - z_0\| \quad (13)$$

$$\|\nabla_{\mathcal{F}} L(z_0, \hat{W})\| \cdot \|\mathbf{h}_{test} - z_0\| = \left\| \frac{W_V}{\sqrt{d}} \mathbf{h} \right\| \cdot \|\mathbf{h}_{test} - z_0\| \quad (14)$$

For L_{12} , we similarly prove that:

$$L_{12} = \sum_{i=1}^{dd'} \lambda_i b_i^2 \geq \lambda_{dd'} \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\|^2 \quad (15)$$

Combining Eqs. (12)-(15), we finally get:

$$\begin{aligned} L_1 &\geq \lambda_{dd'} \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\|^2 - \lambda_1 \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| \cdot (\mu \cdot C_1 \\ &\quad \cdot \|\mathbf{h}_{test} - z_0\| \cdot \|\mathbf{h}_{test}\| + \left\| \frac{W_V}{\sqrt{d}} \mathbf{h} \right\| \cdot \|\mathbf{h}_{test} - z_0\|). \end{aligned} \quad (16)$$

According to Eq. (7), the right-hand side of Eq. (16) is greater than 0, which leads to the conclusion. \square

Extension to k -shot setting. In Theorem 1, we only consider one demonstration X (i.e., the one-shot scenario). For the k -shot scenario, Eq (3) can be written as

$$\begin{aligned} \mathcal{F}_{ICL}^k(\mathbf{h}_{test}) &\approx \frac{W_V}{\sqrt{d}} \mathbf{h}_{test} \cdot (W_K \mathbf{h}_{test})^T \cdot \mathbf{h}_{test} \\ &\quad + \sum_{i=1}^k \frac{W_V}{\sqrt{d}} \mathbf{h}_i \cdot (W_K \mathbf{h}_i)^T \cdot \mathbf{h}_{test}, \end{aligned} \quad (17)$$

where $\mathbf{h}_1, \dots, \mathbf{h}_k$ are the representations of demonstrations X_1, \dots, X_k . This formalization can be interpreted as introducing k training samples $z_1 = W_K \mathbf{h}_1, \dots, z_k = W_K \mathbf{h}_k$ to optimize the linear function $\mathcal{F}(z)$ simultaneously, where the gradient at each training sample z_i satisfies

$$\nabla_{\mathcal{F}} L(z_i, W_0) = \frac{W_V}{\sqrt{d}} \mathbf{h}_i. \quad (18)$$

Similar to the proof of Theorem 1, we derive the following Theorem 2 to illustrate the condition of these samples to ensure a reduction in the loss of testing data X_{test} , where

$$\begin{aligned} \hat{W}_{\epsilon, \bar{z}_k} &\stackrel{def}{=} \underset{W}{\operatorname{argmin}} \frac{1}{|\mathcal{D}_{pre}|} \sum_{z \in \mathcal{D}_{pre}} L(\mathcal{F}(z)) \\ &\quad + \epsilon \cdot \sum_{i=1}^k L(\mathcal{F}(z_i)) \end{aligned} \quad (19)$$

Theorem 2. Assume $\nabla_{\mathcal{F}} L$ is Lipschitz continuous w.r.t \mathcal{F} with constant μ . If inequality (20) holds true, then $L(\mathbf{h}_{test}, \hat{W}_{\frac{1}{|\mathcal{D}_{pre}|}, \bar{z}_k}) < L(\mathbf{h}_{test}, \hat{W}_{0, \bar{z}_k})$, i.e., introducing training samples $\{z_1, \dots, z_k\}$ (i.e., demonstrations X_1, \dots, X_k) can reduce the testing loss on \mathbf{h}_{test} .

$$\begin{aligned} \frac{k \lambda_{dd'}}{\lambda_1} \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| &> \sum_{i=1}^k \|\mathbf{h}_{test} - z_i\| \cdot \\ &\quad \left(\left\| \frac{W_V}{\sqrt{d}} \mathbf{h}_i \right\| + \mu C_1 \right) \end{aligned} \quad (20)$$

Theorem 2 further indicates that, under our setup, the joint effect of different demonstrations follows an additive relationship. This implies that the selection of k different demonstrations can be approximately considered independently. We leave the exploration of more complex interactions among demonstrations to future work.

4. LMS3: Method Design

Based on Section 3, an ideal demonstration X needs to maximize the value of L_1 (i.e., minimize the empirical testing loss $L(\mathbf{h}_{test}, \hat{W}_{\frac{1}{|\mathcal{D}_{pre}|}, z_0})$ in Eq. (9)). This is equivalent to minimize the right-hand side of Eq. (7) according to Eq. (16) and can be further divided into: 1) minimize

$$\operatorname{Sim}(X) \stackrel{def}{=} \|\mathbf{h}_{test} - W_K^T \cdot W_Q \mathbf{h}\|, \quad (21)$$

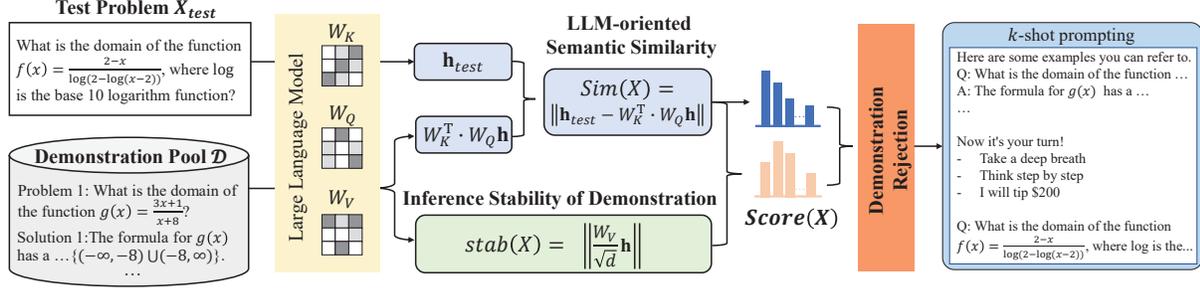


Figure 2. Illustration of our proposed LMS3 method.

(recall $z_0 = W_K \mathbf{h}$ and W_K is indeed $W_K^T \cdot W_Q$ in the aforementioned section), and 2) minimize

$$Stab(X) \stackrel{def}{=} \left\| \frac{W_V}{\sqrt{d}} \mathbf{h} \right\|. \quad (22)$$

Specifically, $Sim(X)$ reflects an **LLM-oriented Semantic Similarity** between the demonstration X and the test data X_{test} . It illustrates the need to find a demonstration similar to the test sample. However, it goes beyond traditional methods by taking into account 1) the whole reasoning path of demonstrations (recall X includes both the problem context and the solution) and 2) the characteristics of the inference LLM itself, which is more consistent with intuition. The value of $Stab(X)$ is an evaluation of the **Inference Stability of Demonstration** X . Based on Eq (5), $Stab(X)$ is indeed the length of gradient of the loss function on X . If $Stab(X)$ is low, it indicates that the LLM has reached a stable prediction with minimal loss on X , and the parameters will not be excessively altered due to the introduction of this sample. In other words, the LLM has been able to stably deduce the correct answer for demonstration X .

Since it is hard to simultaneously achieve the minimum of $Sim(X)$ and $Stab(X)$, two naive approximations are to minimize a demonstration scoring function that calculates their sum or product:

$$Score(X) = Sim(X) + \alpha_1 \cdot Stab(X), \quad (23)$$

$$Score(X) = Sim(X) \cdot Stab(X), \quad (24)$$

However, Eq. (23) requires considering the scale differences between the two objectives and adjusting the hyperparameter α_1 based on different LLMs and datasets, which is challenging to apply in practice. Therefore, we prefer Eq. (24) as the basic form of our scoring function. To implement k -shot in-context learning, we can select the top k samples with the highest $Score(X)$ as demonstrations according to Theorem 2, which can ensure that the most relevant and stable samples are used to enhance the LLM’s performance.

Furthermore, we design a demonstration rejection mechanism. For instance, it is possible that the examples with the

Method	Theoretical Dependent		Generality	Complexity
	Guarantee	on LLM		
Similar-ICL	✗	✗	✓	$\mathcal{O}(\mathcal{M} + \mathcal{N})$
Influence	✗	✓	✓	$\mathcal{O}(\frac{\mathcal{M}\mathcal{V}}{\mathcal{S}} + \mathcal{N})$
InfICL	✗	✓	✗	$\mathcal{O}(\mathcal{D}\mathcal{M} + \mathcal{D}\mathcal{V} + \mathcal{N})$
TopK+MDL	✗	✓	✗	$\mathcal{O}(\mathcal{M} + \mathcal{K}\mathcal{N})$
TopK+ConE	✗	✓	✗	$\mathcal{O}(\mathcal{M} + \mathcal{K}\mathcal{N})$
IDS	✗	✓	✓	$\mathcal{O}(\mathcal{M} + \mathcal{Q}\mathcal{N})$
MI	✗	✓	✗	$\mathcal{O}(\mathcal{M}\mathcal{N})$
SPELL	✗	✓	✓	$\mathcal{O}(\mathcal{M}\mathcal{R} + \mathcal{N})$
LMS3 (ours)	✓	✓	✓	$\mathcal{O}(\mathcal{M} + \mathcal{N})$

Table 1. Comparison of methods, including Similar-ICL (Liu et al., 2022; Zhang et al., 2023; Fu et al., 2022; Chen et al., 2024b), Influence (Nguyen & Wong, 2023) (\mathcal{S} is the size of subset used to estimate influences, \mathcal{V} is the size of validation set), InfICL (Van et al., 2024) (\mathcal{D} is the number of parameters of external LLMs), TopK+MDL (Wu et al., 2023), TopK+ConE (Peng et al., 2024) (\mathcal{K} is the number of candidate demonstrations), IDS (Qin et al., 2023) (\mathcal{Q} is the number of iterations), MI (Sorensen et al., 2022), SPELL (Gonen et al., 2023) (\mathcal{R} is the number of samples for estimating perplexity). The generality setting to ✗ indicates that these works are more suitable for classification tasks and hard to implement for mathematical reasoning task.

highest $Score(X)$ still do not satisfy Eq. (20). In such cases, unlike existing methods that always select top k examples, we tend to refuse to provide any demonstration and instead use a zero-shot approach, because our theorems suggest that providing examples in this case will have a negative effect. We control $Sim(X)$ to achieve this rejection mechanism, because if an example’s $Sim(X)$ is already too large, $Sim(X) \cdot \mu C_1$ might have exceeded the left-hand side of Eq. (7). However, setting an absolute threshold for $Sim(X)$ is challenging since μ, C_1 is unknown, and calculating the gradient norm $\|\nabla_{\mathcal{W}} L(\mathbf{h}_{test}, \hat{\mathcal{W}})\|$ is costly. Therefore, we adopt a simplified relative threshold. We expect that the $Sim(X)$ of an ideal demonstration should be as small as possible relative to all examples. Consequently, we rank $Sim(X)$ of all candidate examples. If a demonstration X ranked top- k in $Score(X)$ does not have a $Sim(X)$ value within the top λ smallest, we reject to select it.

Theoretically, to compute $Score(X)$, we need to input the

	Llama2-13B			Llama3-8B		
	MAWPS	GSM8K	MATH	MAWPS	GSM8K	MATH
zero-shot	0.835±0.009	0.414±0.004	0.096±0.005	0.951±0.004	0.820±0.016	0.324±0.022
Random	0.816±0.004	0.405±0.007	0.090±0.010	0.951±0.005	0.813±0.003	0.330±0.009
Best-validate	0.826±0.001	0.410±0.005	0.096±0.007	0.932±0.000	0.817±0.008	0.332±0.008
TF-IDF	0.826±0.021	0.424±0.007	0.099±0.006	0.945±0.009	0.803±0.007	0.344±0.005
BM25	0.815±0.008	0.416±0.014	0.098±0.007	0.932±0.003	0.805±0.002	0.334±0.004
T5	0.810±0.004	<u>0.426</u> ±0.013	0.093±0.006	0.948±0.021	0.817±0.002	0.330±0.009
BGEM3	0.818±0.013	0.407±0.004	0.100±0.011	0.938±0.017	0.802±0.000	0.340±0.005
OpenAI	0.805±0.014	0.416±0.005	0.101±0.002	<u>0.965</u> ±0.011	0.809±0.008	<u>0.346</u> ±0.002
SPELL	0.797±0.009	0.394±0.006	0.085±0.003	0.945±0.005	<u>0.821</u> ±0.008	0.343±0.004
Influence	0.836±0.010	0.405±0.009	<u>0.102</u> ±0.000	0.929±0.009	0.800±0.015	0.333±0.006
IDS	<u>0.839</u> ±0.005	0.424±0.012	0.088±0.001	0.920±0.003	0.808±0.001	0.330±0.001
LMS3 (ours)	0.854* ±0.008	0.447* ±0.014	0.124* ±0.003	0.966 ±0.010	0.837* ±0.011	0.353* ±0.002

Table 2. One-shot Answer Accuracy, with the best/runner-up methods highlighted in bold/underlined.

concatenation of each “(demonstration, testing data)” pair (X, X_{test}) into the LLM to obtain their semantic representations $\mathbf{h}, \mathbf{h}_{test}$. However, in practice, this process requires $\mathcal{O}(MN)$ complexity (measured by the number of LLM API calls) for testing, which significantly limits the efficiency. Therefore, we approximate by inputting each data individually into the LLM to obtain its semantic representation, reducing the testing complexity to $\mathcal{O}(M + N)$ (the representations of all demonstrations can be pre-computed).

We illustrate the workflow of our method, named LMS3, in Figure 2 and present the pseudo-code in Appendix B. LMS3 has several advantages as summarized in Table 1. 1)

Theoretical Guarantee: To the best of our knowledge, we are the first to theoretically quantify the impact of demonstrations on ICL reasoning performance and explain why and when they work. 2) **Rational Dependency:** Our analysis verifies that the optimal demonstration depends on the inference LLM (i.e., how the representations $\mathbf{h}, \mathbf{h}_{test}$ are encoded). This is reasonable because an LLM’s understanding of similar problems sets the upper limit on its ability to leverage these problems (Peng et al., 2024). Consequently, the optimal demonstration should be selected adaptively for different LLMs. However, existing methods like Similar-ICL estimate semantic similarity independently of the inference LLM and the demonstration is the same for all LLMs. 3) **Generalization Ability:** If we set $W_K^T \cdot W_Q = I$ as an identity matrix and omit $Stab(X)$, our method degenerates into finding the demonstration with the closest semantic representation to the test data. This perspective unifies the current approaches, summarizing their main differences in the setting of $W_K^T \cdot W_Q$ to obtain semantic representations. Besides, our analysis is based on the impact of demonstrations on the test loss, which is not dependent on the task type. In addition to mathematical reasoning, it is also applicable to generation tasks or classification tasks. 4) **Low Complexity:** Compared to methods based on impact estimation or LLMs’ feedback (Van et al., 2024; Nguyen & Wong, 2023;

Chang & Jia, 2023), our method does not require additional external LLMs, repeated testing of demonstration effects on validation set, or the computation of Hessian matrix, which brings much lower complexity.

5. Experiments

5.1. Experimental Setup

Datasets. We use three datasets that cover a variety of types and difficulty levels. **MAWPS** (Koncel-Kedziorski et al., 2016) consists of 2,373 elementary-level math word problems. **GSM8K** (Cobbe et al., 2021) is composed of 8,792 more challenging elementary problems with a higher number of steps. **MATH** (Hendrycks et al., 2021) is collected from high school math competition, containing 12,500 problems across seven categories, and is currently one of the most widely used benchmarks. Dataset partition and statistics are presented in Appendix C.

Baselines. We use Llama2-13B (Touvron et al., 2023) and Llama3-8B (Meta, 2024) as the backbones and take 10 representative and SOTA baselines including:

- **Random** randomly samples a data from \mathcal{D} .
- **Best-validate** selects the data with the highest accuracy on a validation set,

and some typical Similar-ICL methods:

- **TF-IDF** represents each problem as a TF-IDF vector, and selects the nearest sample.
- **BM25** (Robertson et al., 2009) selects demonstrations by retrieval method BM25.
- **T5** (Raffel et al., 2020) encodes problems with T5-large and selects the most similar one.

- **BGEM3** (Chen et al., 2024b) integrate multiple information retrieval functionalities in a unified embedding.
- **OpenAI** (Neelakantan et al., 2022) represents problems with OpenAI Text-Embedding-3-Small model,

and methods that do not rely on problem similarity:

- **SPELL** (Gonen et al., 2023) selects demonstrations by calculating their perplexity.
- **Influence** (Nguyen & Wong, 2023) divides \mathcal{D} into multiple subsets. The preference of a demonstration is calculated by the difference in validation accuracy between subsets that include and exclude it.
- **IDS** (Qin et al., 2023) iteratively selects demonstrations based on reasoning path similarity.

Implementation Details. When implementing our LMS3, we set λ to 10% for Llama2-13B and 1% for Llama3-8B. The temperature for both LLMs is set to 0.8. For the baseline Influence, the size \mathcal{S} of the subset is set to 20. For the baseline IDS, the number \mathcal{Q} of iterations is set to 3. All experiments are conducted on a server with six NVIDIA RTX 3090 GPUs.

5.2. Performance on One-shot Reasoning

In Table 2, we present the performance of all methods in the one-shot setting. Firstly, it can be seen that our LMS3 outperforms all baselines across all datasets, and this effect is statistically significant with $p \leq 0.05$ (marked *). This directly indicates that the demonstrations chosen by our method better stimulate the LLM’s contextual learning ability. We also present some cases in Appendix D for clearer comparison. Secondly, our LMS3 is the only one that consistently provides improvements over the zero-shot setting, while other methods exhibit certain fluctuations across different datasets. This can be attributed to our method being designed based on a theoretical analysis of when one-shot learning is effective (i.e., Theorem 1). These experimental results validate the rationality, effectiveness, and strong robustness of our theoretical findings. Thirdly, we observe that one-shot learning generally improves the backbone’s performance on the more challenging MATH dataset, but sometimes shows a decrease on other datasets. We believe this is because the problems in MAWPS and GSM8K are relatively simple, and the LLM itself already has the capability to solve them. Introducing additional examples in this case might instead mislead the model’s reasoning thought.

5.3. Performance on Few-shot Reasoning

Now we validate our LMS3 in the k -shot scenario, with Llama3-8B’s performances at $k = \{2, 3, 4\}$ visualized in

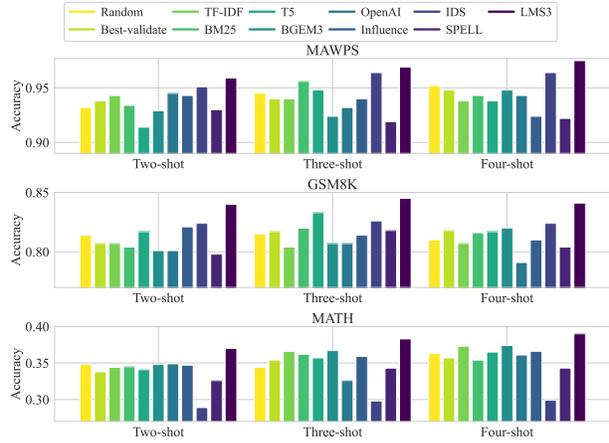


Figure 3. Few-shot Answer Accuracy of Llama3-8B.

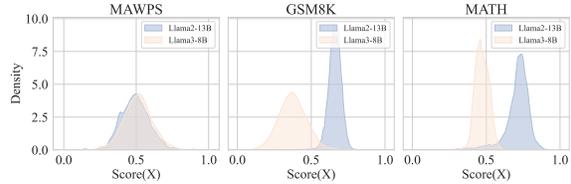


Figure 4. Distribution of $Score(X)$ in Eq. (24).

Figure 3. Firstly, our method remains superior across different settings, which is consistent with our Theorem 2, demonstrating its high applicability to various scenarios. Secondly, as k increases, the trend of reasoning performance varies across different datasets. On MAWPS and MATH, the performances of most methods consistently improve with a higher k . However, on GSM8K, the accuracy for almost all methods declines after $k = 3$. This suggests that an excessive number of demonstrations does not necessarily lead to increased accuracy and we need to balance the number and length of demonstrations. A dataset with longer problem lengths (i.e., GSM8K as indicated in Appendix C) may require fewer examples to achieve optimal performance.

5.4. Analysis of Scoring Function

Figure 4 presents the distribution of $Score(X)$ in Eq. (24) normalized by z-score, which verifies that our $Score(X)$ has good discriminative power for different samples. More importantly, we observe that the variances of the distributions for Llama2 on GSM8K and MATH, Llama3 on MATH, are relatively small. This indicates that the differences between samples in these cases are not significant, which can explain why most other one-shot baselines also perform better than the zero-shot setting in Table 2. In contrast, in other cases (e.g., on MAWPS), the performance gap between different samples is larger, and only our LMS3 can consistently achieve better results than zero-shot setting.

What Makes In-context Learning Effective for Mathematical Reasoning

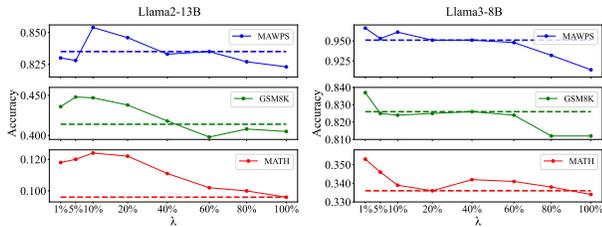


Figure 5. Performance with varying λ . The dashed line corresponds to the result of the zero-shot setting.

Method	MAWPS	GSM8K	MATH
Random	0.951	0.813	0.330
+ours	0.952	0.818	0.349
Best-validate	0.932	0.817	0.332
+ours	0.941	0.829	0.344
TF-IDF	0.945	0.803	0.344
+ours	0.946	0.818	0.351
BM25	0.932	0.805	0.334
+ours	0.934	0.812	0.335
T5	0.948	0.817	0.330
+ours	0.953	0.828	0.333
BGEM3	0.938	0.802	0.340
+ours	0.941	0.822	0.350
OpenAI	0.965	0.809	0.346
+ours	0.973	0.818	0.347
SPELL	0.945	0.821	0.343
+ours	0.946	0.826	0.345
Influence	0.929	0.800	0.333
+ours	0.935	0.810	0.340
IDS	0.920	0.808	0.330
+ours	0.932	0.823	0.346

Table 3. Performance of combining our rejection mechanism with other demonstration selection methods.

5.5. Necessity of Demonstration Rejection

To validate the necessity of our proposed demonstration rejection mechanism, we test the effects of $\lambda = \{1\%, 5\%, 10\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$. It is noteworthy that $\lambda = 100\%$ is equivalent to removing our rejection mechanism. From Figure 5, we can first observe that when $\lambda = 100\%$, the accuracy of LMS3 falls below that of the zero-shot results, which highlights the necessity of our rejection mechanism. Secondly, when λ increases, the performance of Llama2 initially rises and then falls, while the performance of Llama3 consistently declines. On one hand, this indicates that the strength of λ needs to be balanced differently for various LLMs, but this process is not complicated since the optimal λ is basically within 10%. On the other hand, this demonstrates that our $Sim(X)$ can effectively approximate the conditions in Theorems 1 and 2, as using it to implement the rejection mechanism can improve the model’s performance. Thirdly, in Table 3, we apply our rejection mechanism to all baselines (denoted as “+ours”) with Llama3-8B as the backbone. The consistent performance gains reflect the general applicability of our mechanism and underscores the importance of considering

	ChatGPT			GPT-4		
	MAWPS	GSM8K	MATH	MAWPS	GSM8K	MATH
zero-shot	0.906	0.800	0.482	0.941	0.954	0.758
Random	0.858	0.839	0.503	0.976	0.946	0.702
Best-validate	0.831	0.832	0.519	0.979	0.951	0.715
TF-IDF	0.895	0.820	0.514	0.975	0.947	0.724
BM25	0.901	0.828	0.510	0.987	0.953	0.691
T5	0.893	0.840	0.508	0.973	0.950	0.718
BGEM3	0.896	0.838	0.504	0.986	0.955	0.705
OpenAI	0.898	0.829	0.513	0.979	0.945	0.699
Influence	0.878	<u>0.848</u>	0.515	0.974	0.955	0.702
IDS	<u>0.908</u>	<u>0.848</u>	0.505	0.979	<u>0.959</u>	0.742
LMS3 (ours)	0.909	0.862	<u>0.517</u>	0.990	0.961	<u>0.752</u>

Table 4. Generalization performance on ChatGPT/GPT4.

when to include a demonstration in ICL.

5.6. Generalization Ability

One advantage of our method is that it reasonably considers the inference LLM when selecting demonstrations. However, it may raise concerns about its extensibility, as it requires access to the LLM’s internal parameters. To address this issue, we offer the demonstrations selected by LMS3 (Llama3-8B) directly to the most advanced LLMs ChatGPT and GPT-4, compared with other methods that do not rely on the LLM’s parameters. From Table 4, we can see that our LMS3 still achieves nearly optimal results, demonstrating the excellent generalization and representativeness of our selected demonstrations. Besides, we observe that all methods negatively impact the reasoning performance on MATH dataset when applied to GPT-4. We attribute this to that chain-of-thought examples may no longer be significantly beneficial for GPT-4, and future examples for GPT-4 might need to focus more on code (Chen et al., 2023; Gao et al., 2023) or other formats.

5.7. Applicability to More Tasks

To expand the applicability of our work and demonstrate the versatility of LMS3, we include an additional experiment on CommonsenseQA, a large-scale benchmark designed to evaluate commonsense reasoning task and has been widely used in ICL research (Ye et al., 2023; Qin et al., 2023; Min et al., 2022). This extension allows us to showcase LMS3’s effectiveness not only in mathematical reasoning tasks but also in broader reasoning domains that involve more nuanced and diverse knowledge.

The results in Table 5 show that LMS3 outperforms all baseline methods on both Llama2-13B and Llama3-8B, with an accuracy of 0.444 and 0.650, respectively. The improvement is particularly significant on Llama2-13B, where LMS3 surpasses the next best method by a larger margin. These results highlight the effectiveness of LMS3 in enhancing model performance across different tasks, demonstrating its potential for real-world applications.

	Llama2-13B	Llama3-8B
zero-shot	0.428	0.628
Random	0.415	0.625
Best-validate	0.375	0.605
TF-IDF	0.368	0.617
BM25	0.383	0.627
T5	0.419	0.617
BGEM3	0.377	0.589
OpenAI	0.393	0.607
SPELL	0.409	0.637
Influence	0.374	0.605
IDS	0.408	0.640
LMS3 (ours)	0.444	0.650

Table 5. Performance of CommonsenseQA benchmark.

6. Conclusion

In this paper, we theoretically analyzed how demonstrations affected LLMs’ mathematical reasoning performance. On this basis, we proposed a LMS3 method that balanced LLM-oriented semantic similarity and inference stability of demonstrations, and introduced a demonstration rejection mechanism to filter out negative situations. Experiments showed that our method was the only one to consistently improve the reasoning accuracy of LLMs, and our demonstrations exhibited strong generalization ability and interpretability. In the future, we will extend our method to more NLP tasks and apply our theory to broader scenarios. Please refer to Appendix E for more discussions and details.

Impact Statement

This work provides valuable insights into the underlying mechanisms of LLMs. By theoretically analyzing how in-context demonstrations affect LLM performance, we offer a deeper understanding of LLM behavior, which paves the way for more effective optimizations in the future. This could lead to reduced costs and improved efficiency in deploying LLMs for reasoning tasks. Additionally, our LMS3 method has practical implications for improving the reliability and relevance of LLM outputs in various applications, especially in decision-making fields that rely on complex reasoning and data-driven insights.

Acknowledgement

This research was partially supported by the National Natural Science Foundation of China (Grants No.62477044), Anhui Provincial Natural Science Foundation (No. 2308085QF229), the Fundamental Research Funds for the Central Universities (No.WK2150110038). Zhenya Huang gratefully acknowledges the support of the Young Elite Scientists Sponsorship Program by CAST (No. 2024QNRC001).

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

An, S., Lin, Z., Fu, Q., Chen, B., Zheng, N., Lou, J.-G., and Zhang, D. How do in-context examples affect compositional generalization? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11027–11052, 2023.

Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024.

Besta, M., Blach, N., Kubicek, A., et al. Graph of thoughts: Solving elaborate problems with large language models. In *AAAI*, volume 38, pp. 17682–17690, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chang, T.-Y. and Jia, R. Data curation alone can stabilize in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8123–8144, 2023.

Chen, J., Lin, H., Han, X., and Sun, L. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17754–17762, 2024a.

Chen, J., Xiao, S., Zhang, P., et al. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024b.

Chen, W., Ma, X., Wang, X., and Cohen, W. W. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dai, D., Sun, Y., Dong, L., Hao, Y., Ma, S., Sui, Z., and Wei, F. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, 2023.

- Ding, Y., Fan, W., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., and Li, Q. A survey on rag meets llms: Towards retrieval-augmented large language models. *arXiv preprint arXiv:2405.06211*, 2024.
- Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., and Sui, Z. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.
- Feigenbaum, E. A., Feldman, J., et al. *Computers and thought*. New York McGraw-Hill, 1963.
- Fletcher, C. R. Understanding and solving arithmetic word problems: A computer simulation. *Behavior Research Methods, Instruments, & Computers*, 17(5):565–571, 1985.
- Fu, Y., Peng, H., Sabharwal, A., Clark, P., and Khot, T. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*, 2022.
- Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J., and Neubig, G. Pal: Program-aided language models. In *International Conference on Machine Learning*, pp. 10764–10799. PMLR, 2023.
- Gao, X. and Das, K. Customizing language model responses with contrastive in-context learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18039–18046, 2024.
- Gonen, H., Iyer, S., Blevins, T., Smith, N. A., and Zettlemoyer, L. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10136–10148, 2023.
- Han, C., Wang, Z., Zhao, H., and Ji, H. Explaining emergent in-context learning as kernel regression. *arXiv preprint arXiv:2305.12766*, 2023.
- He, D., Feng, Z., Jin, D., Wang, X., and Zhang, W. Joint identification of network communities and semantics via integrative modeling of network topologies and node contents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.
- Koncel-Kedziorski, R., Roy, S., Amini, A., et al. Mawps: A math word problem repository. In *NAACL-HLT*, pp. 1152–1157, 2016.
- Ling, R. F. Residuals and influence in regression, 1984.
- Liu, J., Shen, D., Zhang, Y., Dolan, W. B., Carin, L., and Chen, W. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114, 2022.
- Liu, J., Huang, Z., Zhai, C., and Liu, Q. Learning by applying: A general framework for mathematical reasoning via enhancing explicit knowledge learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4497–4506, 2023.
- Liu, J., Huang, Z., Liu, Q., Ma, Z., Zhai, C., and Chen, E. Knowledge-centered dual-process reasoning for math word problems with large language models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.
- Luo, M., Xu, X., Dai, Z., Pasupat, P., Kazemi, M., Baral, C., Imbrasaite, V., and Zhao, V. Y. Dr. icl: Demonstration-retrieved in-context learning. *arXiv preprint arXiv:2305.14128*, 2023.
- Ma, Z., Huang, Z., Liu, J., Wang, M., Zhao, H., and Li, X. Automated creation of reusable and diverse toolsets for enhancing llm reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24821–24830, 2025.
- Meade, N., Gella, S., Hazarika, D., Gupta, P., Jin, D., Reddy, S., Liu, Y., and Hakkani-Tur, D. Using in-context learning to improve dialogue safety. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 11882–11910, 2023.
- Meta, A. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*, 2024.
- Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, 2022.
- Neelakantan, A., Xu, T., Puri, R., Radford, A., Han, J. M., Tworek, J., Yuan, Q., Tezak, N., Kim, J. W., Hallacy,

- C., et al. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*, 2022.
- Nguyen, T. and Wong, E. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*, 2023.
- Pan, J., Gao, T., Chen, H., and Chen, D. What in-context learning” learns” in-context: Disentangling task recognition and task learning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.
- Pei, H., Yang, B., Liu, J., and Chang, K. C.-C. Active surveillance via group sparse bayesian learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1133–1148, 2020.
- Peng, K., Ding, L., Yuan, Y., Liu, X., Zhang, M., Ouyang, Y., and Tao, D. Revisiting demonstration selection strategies in in-context learning. *arXiv preprint arXiv:2401.12087*, 2024.
- Qin, C., Zhang, A., Dagar, A., and Ye, W. In-context learning with iterative demonstration selection. *arXiv preprint arXiv:2310.09881*, 2023.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.
- Robertson, S., Zaragoza, H., et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.
- Sorensen, T., Robinson, J., Rytting, C., Shaw, A., Rogers, K., Delorey, A., Khalil, M., Fulda, N., and Wingate, D. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 819–862, 2022.
- Talmor, A., Herzig, J., Lourie, N., and Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *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)*, pp. 4149–4158, 2019.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- Van, M.-H., Wu, X., et al. In-context learning demonstration selection via influence analysis. *arXiv preprint arXiv:2402.11750*, 2024.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022.
- Wei, J., Wei, J., Tay, Y., et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.
- Wu, Z., Wang, Y., Ye, J., and Kong, L. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1423–1436, 2023.
- Xiao, T., Liu, J., Huang, Z., Wu, J., Sha, J., Wang, S., and Chen, E. Learning to solve geometry problems via simulating human dual-reasoning process. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 6559–6568, 2024.
- Xue, S., Huang, Z., Liu, J., Lin, X., Ning, Y., Jin, B., Li, X., and Liu, Q. Decompose, analyze and rethink: Solving intricate problems with human-like reasoning cycle. *Advances in Neural Information Processing Systems*, 37: 357–385, 2024.
- Yan, J., Xu, J., Song, C., Wu, C., Li, Y., and Zhang, Y. Understanding in-context learning from repetitions. In *The Twelfth International Conference on Learning Representations*, 2024.
- Yao, S., Yu, D., Zhao, J., et al. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.
- Ye, J., Wu, Z., Feng, J., Yu, T., and Kong, L. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pp. 39818–39833. PMLR, 2023.
- Zhang, D., Wang, L., et al. The gap of semantic parsing: A survey on automatic math word problem solvers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(9):2287–2305, 2020.
- Zhang, P., Xiao, S., Liu, Z., Dou, Z., and Nie, J.-Y. Retrieve anything to augment large language models. *arXiv preprint arXiv:2310.07554*, 2023.

A. Proof of Theorem 1

We interpret the attention mechanism in ICL setting as: 1) We have a linear function $\mathcal{F}(z)$ with initialized parameters

$$W_0 = \frac{W_V}{\sqrt{d}} \mathbf{h}_{test} \cdot (W_K \mathbf{h}_{test})^T. \quad (25)$$

2) We introduce a training data $z_0 = W_K \mathbf{h}$ to optimize the parameters, with the gradient at (z_0, W_0) satisfies

$$\nabla_{\mathcal{F}} L(z_0, W_0) = \frac{W_V}{\sqrt{d}} \mathbf{h}. \quad (26)$$

3) We finally apply the optimized parameters to calculate the result of test data \mathbf{h}_{test} .

Theorem 1. Assume $\nabla_{\mathcal{F}} L$ is Lipschitz continuous w.r.t \mathcal{F} with constant μ . If inequality (7) holds true, then $L(\mathbf{h}_{test}, \hat{W}_{\frac{1}{|\mathcal{D}_{pre}|}, z_0}) < L(\mathbf{h}_{test}, \hat{W}_{0, z_0})$, i.e., introducing the training sample z_0 (i.e., demonstration X) can reduce the testing loss on \mathbf{h}_{test} . $\frac{1}{\lambda_{dd'}}$, $\frac{1}{\lambda_1}$ are the largest and smallest eigenvalues of $H_{\hat{W}}$, respectively.

$$\begin{aligned} \frac{\lambda_{dd'}}{\lambda_1} \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| &> \|\mathbf{h}_{test} - z_0\| \cdot \left(\|\frac{W_V}{\sqrt{d}} \mathbf{h}\| + \mu C_1 \right) \\ C_1 &= \left\| \frac{W_V}{\sqrt{d}} \mathbf{h}_{test} \right\| \cdot \|W_K \mathbf{h}_{test}\| \cdot \|\mathbf{h}_{test}\| \end{aligned} \quad (27)$$

Proof. With $\hat{W}, \hat{W}_{\epsilon, z_0}$, the influence of upweighting z_0 on the parameters \hat{W} is (Ling, 1984; Koh & Liang, 2017):

$$\begin{aligned} \mathcal{I}_{parameter}(z_0) &= \left. \frac{d\hat{W}_{\epsilon, z_0}}{d\epsilon} \right|_{\epsilon=0} \\ &= -H_{\hat{W}}^{-1} \nabla_W L(z_0, \hat{W}), \end{aligned} \quad (28)$$

and the influence on the loss function is:

$$\begin{aligned} \mathcal{I}_{loss}(z_0) &= \left. \frac{dL(\mathbf{h}_{test}, \hat{W}_{\epsilon, z_0})}{d\epsilon} \right|_{\epsilon=0} \\ &= \nabla_W L(\mathbf{h}_{test}, \hat{W})^T \cdot \left. \frac{d\hat{W}_{\epsilon, z_0}}{d\epsilon} \right|_{\epsilon=0} \\ &= -\nabla_W L(\mathbf{h}_{test}, \hat{W})^T \cdot H_{\hat{W}}^{-1} \nabla_W L(z_0, \hat{W}) \end{aligned} \quad (29)$$

Then, the testing loss $L(\mathbf{h}_{test}, \hat{W}_{\frac{1}{|\mathcal{D}_{pre}|}, z_0})$ can be evaluated by Taylor approximation since $\frac{1}{|\mathcal{D}_{pre}|}$ is sufficiently small, i.e., $L(\mathbf{h}_{test}, \hat{W}_{\frac{1}{|\mathcal{D}_{pre}|}, z_0}) \approx$

$$\begin{aligned} &L(\mathbf{h}_{test}, \hat{W}_{0, z_0}) + \frac{1}{|\mathcal{D}_{pre}|} \left. \frac{dL(\mathbf{h}_{test}, \hat{W}_{\epsilon, z_0})}{d\epsilon} \right|_{\epsilon=0} \\ &= L(\mathbf{h}_{test}, \hat{W}_{0, z_0}) - \frac{1}{|\mathcal{D}_{pre}|} \nabla_W L(\mathbf{h}_{test}, \hat{W})^T \\ &\quad \cdot H_{\hat{W}}^{-1} \nabla_W L(z_0, \hat{W}). \end{aligned} \quad (30)$$

Therefore, now the question turns to evaluate

$$L_1 \stackrel{def}{=} \nabla_W L(\mathbf{h}_{test}, \hat{W})^T \cdot H_{\hat{W}}^{-1} \nabla_W L(z_0, \hat{W}). \quad (31)$$

Specifically, L_1 can be represented as $L_1 =$

$$\begin{aligned} &\underbrace{(\nabla_W L(z_0, \hat{W}) - \nabla_W L(\mathbf{h}_{test}, \hat{W}))^T \cdot H_{\hat{W}}^{-1} \nabla_W L(\mathbf{h}_{test}, \hat{W})}_{L_{11}} \\ &+ \underbrace{\nabla_W L(\mathbf{h}_{test}, \hat{W})^T \cdot H_{\hat{W}}^{-1} \nabla_W L(\mathbf{h}_{test}, \hat{W})}_{L_{12}} \end{aligned} \quad (32)$$

Since $H_{\hat{W}}$ is positive definite, there exists an orthogonal matrix O and a diagonal matrix $\Sigma = \text{diag}(\lambda_1, \dots, \lambda_{dd'})$ satisfying $H_{\hat{W}}^{-1} = O^T \cdot \Sigma \cdot O$, where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{dd'} > 0$ are the eigenvalues of $H_{\hat{W}}^{-1}$. For L_{11} , it equals

$$\begin{aligned} &\underbrace{(O(\nabla_W L(z_0, \hat{W}) - \nabla_W L(\mathbf{h}_{test}, \hat{W})))^T}_{a^T} \text{diag}(\lambda_1, \dots, \lambda_{dd'}) \cdot \\ &\underbrace{O \nabla_W L(\mathbf{h}_{test}, \hat{W})}_{b} = \sum_{i=1}^{dd'} \lambda_i a_i b_i \geq -\sqrt{\sum_{i=1}^{dd'} a_i^2} \sqrt{\sum_{i=1}^{dd'} (\lambda_i b_i)^2} \\ &\geq -\lambda_1 \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| \cdot \|\nabla_W L(\mathbf{h}_{test}, \hat{W}) - \nabla_W L(z_0, \hat{W})\| \\ &\geq -\lambda_1 \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| \cdot \|\text{Flat}(\nabla_{\mathcal{F}} L(\mathbf{h}_{test}, \hat{W}) \cdot \mathbf{h}_{test}^T \\ &\quad - \nabla_{\mathcal{F}} L(z_0, \hat{W}) \cdot z_0^T)\| \\ &\geq -\lambda_1 \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| \cdot \left(\|\text{Flat}((\nabla_{\mathcal{F}} L(\mathbf{h}_{test}, \hat{W}) - \nabla_{\mathcal{F}} L(z_0, \hat{W})) \cdot \mathbf{h}_{test}^T)\| \right. \\ &\quad \left. + \|\text{Flat}(\nabla_{\mathcal{F}} L(z_0, \hat{W}) \cdot (\mathbf{h}_{test} - z_0)^T)\| \right) \\ &= -\lambda_1 \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| \cdot \left(\|\nabla_{\mathcal{F}} L(\mathbf{h}_{test}, \hat{W}) - \nabla_{\mathcal{F}} L(z_0, \hat{W})\| \right. \\ &\quad \left. \cdot \|\mathbf{h}_{test}\| + \|\nabla_{\mathcal{F}} L(z_0, \hat{W})\| \cdot \|\mathbf{h}_{test} - z_0\| \right), \end{aligned} \quad (33)$$

where $\text{Flat}(W)$ is the operation that flatten a matrix $W \in \mathbb{R}^{d' \times d}$ into a vector of length $\mathbb{R}^{d'd}$. Since $\nabla_{\mathcal{F}} L$ is Lipschitz continuous, we get $L_{11} \geq$

$$\begin{aligned} &-\lambda_1 \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| \cdot (\mu \|\hat{W}(\mathbf{h}_{test} - z_0)\| \cdot \|\mathbf{h}_{test}\| + \\ &\|\nabla_{\mathcal{F}} L(z_0, \hat{W})\| \cdot \|\mathbf{h}_{test} - z_0\|) \end{aligned} \quad (34)$$

Applying Eqs. (25)(26) to Eq. (34), we have:

$$\begin{aligned} \|\hat{W}(\mathbf{h}_{test} - z_0)\| &= \sqrt{(\mathbf{h}_{test} - z_0)^T \hat{W}^T \hat{W} (\mathbf{h}_{test} - z_0)} \\ &= \left\| \frac{W_V}{\sqrt{d}} \mathbf{h}_{test} \right\| \cdot |(W_K \mathbf{h}_{test})^T (\mathbf{h}_{test} - z_0)| \\ &\leq \left\| \frac{W_V}{\sqrt{d}} \mathbf{h}_{test} \right\| \cdot \|W_K \mathbf{h}_{test}\| \cdot \|\mathbf{h}_{test} - z_0\| \end{aligned} \quad (35)$$

$$\|\nabla_{\mathcal{F}} L(z_0, \hat{W})\| \cdot \|\mathbf{h}_{test} - z_0\| = \left\| \frac{W_V}{\sqrt{d}} \mathbf{h} \right\| \cdot \|\mathbf{h}_{test} - z_0\| \quad (36)$$

For L_{12} , we similarly have:

$$L_{12} = \sum_{i=1}^{dd'} \lambda_i b_i^2 \geq \lambda_{dd'} \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\|^2 \quad (37)$$

Algorithm 1 Our LMS3

Input: k -shot, X_{test} , \mathcal{D} , λ
Output: Selected demonstration set $\mathcal{D}_k \subseteq \mathcal{D}$

- 1: Calculate $Sim(X)$, $Stab(X)$, $Score(X)$ for $X \in \mathcal{D}$ based on Eqs.(21)(22)(24).
- 2: Define $Score_k \subseteq \mathcal{D}$ as the set of k samples with the smallest $Score(X)$ values.
- 3: Define $Sim_\lambda \subseteq \mathcal{D}$ as the set of λ samples with the smallest $Sim(X)$ values.
- 4: $\mathcal{D}_k = \{\}$.
- 5: **for** $X \in Score_k$ **do**
- 6: **if** $X \in Sim_\lambda$ **do**
- 7: $\mathcal{D}_k = \mathcal{D}_k \cup \{X\}$;

	MAWPS	GSM8K	MATH
Num. Problems	2,373	8,792	12,500
$ \mathcal{D} $	1,898	7,473	7,500
$ \mathcal{D}_{test} $	475	1,319	5,000
Avg. Problem Length	30.08	45.88	30.79
Avg. Solution Length	5.90	48.31	84.50
Difficulty Level	Elementary	Elementary	High School

Table 6. Statistics of datasets.

Combining Eqs. (34)-(37), we finally get:

$$L_1 \geq \lambda_{dd'} \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\|^2 - \lambda_1 \|\nabla_W L(\mathbf{h}_{test}, \hat{W})\| \cdot (\mu \|\frac{W_V}{\sqrt{d}} \mathbf{h}_{test}\| \cdot \|W_K \mathbf{h}_{test}\| \cdot \|\mathbf{h}_{test} - z_0\| \cdot \|\mathbf{h}_{test}\| + \|\frac{W_V}{\sqrt{d}} \mathbf{h}\| \cdot \|\mathbf{h}_{test} - z_0\|).$$
(38)

According to Eq. (27), the right-hand side of Eq. (38) is greater than 0, which leads to the conclusion. \square

B. Pseudo-Code of LMS3

The pseudo-code of our LMS3 is presented in Algorithm 1.

C. Dataset Partition and Statistics

For GSM8K and MATH, we follow their publicly available train/test splits as \mathcal{D} and \mathcal{D}_{test} . For MAWPS, we randomly split the dataset into an 8:2 ratio for $\mathcal{D}/\mathcal{D}_{test}$. We summarize the dataset statistics in Table 6. For each dataset, we also randomly select 200 problems from \mathcal{D} as the validation set to support the implementation of some baselines.

D. Case Study

Figure 6 shows three cases to validate LMS3’s interpretability. For brevity, we omit the solutions for demonstrations.

For cases 1 and 2, the baselines OpenAI and Best-validate both made errors. This indicates that considering only the semantic similarity of demonstrations or the effectiveness of

demonstrations on the validation set alone, is insufficient. It is essential to balance similarity and the inference stability of demonstrations, as LMS3 does, to consistently achieve better results compared to zero-shot setting. In case 3, we again observe that the two baselines incorrectly answer a problem that zero-shot got right. In contrast, LMS3’s rejection mechanism determines that the best demonstration still has a relatively large similarity distance $Sim(X)$ from the test sample (ranked in the top 1.19% exceeding $\lambda = 1\%$). By automatically rejecting this demonstration and adopting the zero-shot setting, LMS3 maintains the original performance, which verifies the necessity and effectiveness of our proposed demonstration rejection mechanism.

E. Discussion and Future Work

Here, we discuss several future directions of LMS3.

First, in this paper, motivated by LLMs’ unreliable performance on mathematical reasoning benchmarks, we derive our Theorems 1 and 2 to illustrate when in-context learning is effective for mathematical reasoning and design a corresponding demonstration selection method LMS3. However, our theories and method can be easily extended to other natural language processing tasks, such as commonsense reasoning (Talmor et al., 2019) and sentiment analysis (Socher et al., 2013; He et al., 2017). Besides, it is also worthwhile to explore how our analysis can be further adapted to multi-modal tasks, such as geometric reasoning (Xiao et al., 2024) and graph learning (Pei et al., 2020). Therefore, in the future, we are willing to refine our theories by incorporating the characteristics of different tasks and to test our LMS3 more broadly.

Second, from our theories, we further find that the correctness of the reasoning paths in the demonstrations may have less impact on the test performance compared with its semantics, which aligns with observations reported in previous studies (Wei et al., 2023; Min et al., 2022). This phenomenon does not meet the goal of enabling LLMs to learn and imitate the reasoning paths of demonstrations. Therefore, we will explore ways to amplify the influence of the demonstration reasoning paths based on our theory for improving the interpretability and controllability of ICL.

Moreover, similar to ICL, retrieval-augmented generation (RAG) (Ding et al., 2024; Chen et al., 2024a) also enables LLMs to perform reasoning based on given information (e.g., retrieved knowledge). However, this information is often semantically unrelated to the test problem (e.g., providing external knowledge) or may not involve reasoning problems in itself (e.g., the knowledge is a declarative statement that does not require LLMs to reason). Therefore, we believe that how to extend our theory to analyze when RAG is effective presents a promising research direction.

What Makes In-context Learning Effective for Mathematical Reasoning

Case 1: Find the sum of the slope and y-intercept of the line through the points (7,8) and (9,0).

Zero-shot: The slope of the line through (7,8) and (9,0) is $m = \frac{0-8}{9-7} = -4$. The y-intercept of the line is 8. Therefore, the sum of the slope and y-intercept is $-4 + 8 = 4$. ❌

OpenAI

Demonstration: What is the sum of the coordinates of the midpoint of the line segment connecting points (5,4) and (-9,8)?

Answer: Let's solve the problem step by step. First, we need to find the slope (m) ... = -4. Now, we need to find the y-intercept (b) ... So, the y-intercept is 28. Finally, we need to find the sum of the slope and y-intercept: $-4 + 28 = 24$. Therefore, the sum of the slope and y-intercept is 24. ❌

Best-validate

Demonstration: Let $f(x) = \begin{cases} ax+3, & \text{if } x > 2 \\ x-5, & \text{if } -2 \leq x \leq 2 \\ 2x-b, & \text{if } x < -2 \end{cases}$. Find a+b if the piecewise function is continuous (which means that its graph can be drawn without lifting your pencil from the paper).

Answer: Let's take a deep breath and think step by step. The slope of the line ... -4. The slope-intercept form of a line is $y = mx + b$ So, the y-intercept is 36. The sum of the slope and y-intercept is: $m + b = -4 + 36 = 32$. Therefore, the sum of the slope and y-intercept is 32. ✅

LMS3 (Ours)

Demonstration: Find the slope of the line that is perpendicular to the line $2x + 3y = 6$.

Answer: Let's break it down step by step! First, we need to find the slope of the line ... We can use the formula ... So, the slope of the line is -4. Next, we need to find the y-intercept of the line. We can use the formula ... So, the y-intercept is 36. Finally, we need to find the sum of the slope and y-intercept. ... Therefore, the sum of the slope and y-intercept is 32. ✅

Case 2: Which type of conic section is described by the equation $(x-y)(x+y) = -2y^2 + 1$?

Zero-shot: The given equation can be simplified as follows: $(x-y)(x+y) = -2y^2 + 1$. This is the equation of a hyperbola. ❌

OpenAI

Demonstration: Which type of conic section is described by the equation $x^2 = 2x + 2y - 3$? Enter "C" for circle, "P" for parabola, "E" for ellipse, "H" for hyperbola, and "N" for none of the above.

Answer: Let's break it down step by step! First, we can expand the left-hand side of the equation: $(x-y)(x+y) = x^2 - y^2$. So, the equation becomes: $x^2 - y^2 = -2y^2 + 1$. Now, let's move all the terms to one side: $x^2 + y^2 = 1$. This is the equation of a circle! Therefore, the correct answer is: C. ✅

Best-validate

Demonstration: Shown below are rows 1, 2, and 3 of Pascal's triangle.

Answer: I'm ready to take on the challenge! Let's start by factoring the equation: $(x-y)(x+y) = -2y^2 + 1$. This is a quadratic equation in x^2 and y^2 , and it's in the form: $ax^2 + by^2 + c = 0$ where a, b, and c are constants. The equation represents a conic section. Since the coefficient of x^2 (a) is positive, and the coefficient of y^2 (b) is positive, the equation represents an ellipse. So, the correct answer is: E. ❌

LMS3 (Ours)

Demonstration: Find the focus of the parabola $Sy = x^2$.

Answer: To determine the type of conic section described by the given equation, we can start by simplifying the equation: $(x-y)(x+y) = -2y^2 + 1$. Expanding the left-hand side, we get: $x^2 - y^2 = -2y^2 + 1$. Rearranging the equation, we get: $x^2 + y^2 = 1$. This is the equation of a circle, so the correct answer is: C. ✅

Case 3: There are four birds at the Caboose. Sally Two is three years older than Granny Red. Granny Red is two times as old as Sally Four. If Sally Four is the same age as Sally Thirtytwo, and Sally Thirtytwo is 8 years old, what's the total age of the four birds?

Zero-shot: Let's break it down step by step! ... so Sally Four is also 8 years old. 2... so Granny Red is $2 \times 8 = 16$ years old. 3... so Sally Two is $16 + 3 = 19$ years old. Now, let's add up the ages of the four birds... The total age of the four birds is: $8 + 19 + 16 + 8 = 51$ years old. ✅

OpenAI

Demonstration: Jerry counts six birds nesting in the bushes, 2/3rd of that number of birds flying overhead, and 3 groups of eight birds each feeding. How many birds does he count in total?

Answer: Let's break it down step by step! ... so Sally Four is also 8 years old. 2... so Granny Red is $2 \times 8 = 16$ years old. 3... so Sally Two is $16 + 3 = 19$ years old. 4. Finally, we add up the ages of the four birds: 8 (Sally Four) + 16 (Granny Red) + 19 (Sally Two) + 4 (Caboose) = 47 . ❌

Best-validate

Demonstration: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Answer: Let's break it down step by step! ... so Sally Four is also 8 years old. 2... so Granny Red is $2 \times 8 = 16$ years old. 3... so Sally Two is $16 + 3 = 19$ years old. 4. Now, let's add up the ages of the four birds: Sally Four (8), Sally Two (19), Granny Red (16), and... wait, there's one more bird, but we're not given its age! Let's assume it's 0 years old (a baby bird?). Then, the total age of the four birds is: $8 + 19 + 16 + 0 = 43$ years. So, the total age of the four birds is 43 years. ❌

LMS3 (Ours)

Demonstration: Sebastian plays drums for percussion in an orchestra. He sits next to the brass section where four people play trombone, two play trumpet, and one plays a French horn. In front of them are the strings section with three violinists, a cellist, and a contrabassist. Next to the strings are a trio of clarinets and a quartet of flute players in the woodwinds section. The maestro conducts them all. How many people are in the orchestra? (Rejected, ranked 1.19% by Sim(X))

Answer: Let's break it down step by step! ... so Sally Four is also 8 years old. 2... so Granny Red is $2 \times 8 = 16$ years old. 3... so Sally Two is $16 + 3 = 19$ years old. Now, let's add up the ages of the four birds... The total age of the four birds is: $8 + 19 + 16 + 8 = 51$ years old. ✅

Figure 6. Case Study.