MetaLadder: Ascending Mathematical Solution Quality via Analogical-Problem Reasoning Transfer

Honglin Lin[1], Zhuoshi Pan[1,2], Yu Li[1], Qizhi Pei[1,3], Xin Gao[1], Mengzhang Cai[1], Conghui He[1], Lijun Wu[1]*

[1]Shanghai AI Laboratory [2]Tsinghua University [3]Renmin University of China

{linhonglin,wulijun}@pjlab.org.cn

Abstract

Large Language Models (LLMs) have demonstrated promising capabilities in solving mathematical reasoning tasks, leveraging Chain-of-Thought (CoT) data as a vital component in guiding answer generation. Current paradigms typically generate CoT and answers directly for a given problem, diverging from human problem-solving strategies to some extent. Humans often solve problems by recalling analogous cases and leveraging their solutions to reason about the current task. Inspired by this cognitive process, we propose **MetaLadder**, a novel framework that explicitly prompts LLMs to recall and reflect on meta-problems, those structurally or semantically analogical problems, alongside their CoT solutions before addressing the target problem. Additionally, we introduce a problem-restating mechanism to enhance the model's comprehension of the target problem by regenerating the original question, which further improves reasoning accuracy. Therefore, the model can achieve reasoning transfer from analogical problems, mimicking human-like "learning from examples" and generalization abilities. Extensive experiments on mathematical benchmarks demonstrate that our MetaLadder significantly boosts LLMs' problem-solving accuracy, largely outperforming standard CoT-based methods (**10.3%** accuracy gain) and other methods.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success in mathematical reasoning tasks by leveraging Chain-of-Thought (CoT) data, which explicitly guides models to decompose problems into intermediate reasoning steps before producing final answers (OpenAI, 2024b; Guo et al., 2025; Team, 2024). Pioneering works such as (Wei et al., 2022) demonstrated that training LLMs on CoT-style solutions significantly improves their ability to solve complex problems, with subsequent studies (Fu et al., 2022; Zhou et al., 2022) further refining this paradigm. For instance, models like Minerva and GPT-4 (Lewkowycz et al., 2022; OpenAI et al., 2023) have showcased near-human performance by distilling high-quality CoT trajectories from expert demonstrations. These methods typically follow a straightforward template: given a problem, the model generates a CoT explanation step-by-step, which then leads to the correct answer. While effective, such approaches align only partially with the nuanced cognitive processes (Daniel, 2011) humans employ during problem-solving.

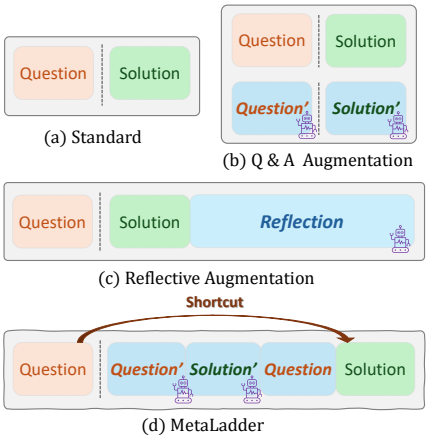

Figure 1: Compare our MetaLadder with other methods (Standard CoT, Question & Answer Augmentation, Reflective Augmentation) on training data construction.

Despite their success, existing CoT-based fine-tuning methods rely on a rigid "Problem → CoT → Answer" framework (Fu et al., 2023; Yu et al., 2024), which diverges from how humans approach challenging mathematical tasks. When solving problems, humans rarely generate solutions in isolation; instead, they actively recall analogous problems and their solutions, especially for difficult or unfamiliar questions (Vosniadou, 1988; Daugherty and Mentzer, 2008). For example, encountering a combinatorics problem, a student might recall similar problems involving permutations or recursive

* Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 4328–4354
November 4-9, 2025

strategies, using their structures to guide the current solution. This ability to leverage prior analogical experiences is critical for generalizing knowledge and tackling novel challenges. However, current LLM training paradigms largely overlook this aspect, treating each problem as an independent instance without encouraging cross-problem reasoning. This limitation constrains models' capacity to transfer learned reasoning patterns, particularly for problems requiring abstract structural or semantic similarities to prior examples.

To bridge this gap, we propose **MetaLadder**, a framework inspired by human-like analogical reasoning and problem comprehension. MetaLadder explicitly guides LLMs to *recall and reflect on meta-problems*—structurally or semantically analogous problems with known CoT solutions—before generating answers for the target problem. These meta-problems and their CoT trajectories serve as scaffolding to derive the current solution, mirroring how humans "stand on the shoulders" of past experiences. Additionally, we introduce a *problem-restating mechanism*: before reasoning, the model regenerates the original question in its own words, enhancing its comprehension of the problem's core components and constraints. This dual mechanism—analogical recall and active restatement—enables the model to decompose complex problems into familiar reasoning patterns, effectively mimicking the human ability to "learn from examples" and generalize solutions across analogous contexts. By integrating these steps, MetaLadder successfully transforms the traditional linear CoT process into a dynamic, context-aware reasoning ladder, where each rung represents a retrieved meta-problem or a refined understanding of the target task.

Extensive experiments validate MetaLadder's effectiveness. On mathematical benchmarks like GSM8K and MATH, models trained with MetaLadder achieve significant improvements over standard CoT fine-tuning baselines, with accuracy gains of 12.4% and 11.3%, respectively, surpassing recent advanced methods. Further analysis reveals that MetaLadder-trained models exhibit stronger generalization to structurally novel problems, solving 9.3% more "out-of-distribution" test cases than vanilla CoT models. Qualitative examples demonstrate that the model not only retrieves relevant meta-problems but also adapts their solutions creatively. These results collectively highlight that emulating human-like analogical reasoning and active comprehension is a powerful yet underexplored direction for advancing LLMs' mathematical reasoning capabilities.

2 Related Work

2.1 Data Synthesis for Math Reasoning

Data synthesis has significantly contributed to the development of LLMs' mathematical reasoning abilities (Yang et al., 2024; Shao et al., 2024). Some studies focus on expanding the dataset and its diversity by rewriting questions or answers (Yu et al., 2024; Yuan et al., 2023; Liu et al., 2024; Tang et al., 2024; Luo et al., 2023). For example, MetaMath (Yu et al., 2024) diversifies the data through various enhancement methods, including question rephrasing, answer augmentation, and the generation of inverse problems. Another line of research focuses on improving the quality and difficulty of the data (Wang et al., 2024a; Luo et al., 2023; Tong et al., 2024; Zhang et al., 2024b; Fan et al., 2024). For instance, WizardMath (Luo et al., 2023) generates more challenging data through RLEIF, while RefAug (Zhang et al., 2024b) adds reflective information after the original CoT process to encourage enhancing the reasoning process. Our method differs by augmenting data to activate the model's analogical reasoning capabilities, enabling the model to generate and apply solutions based on analogous problems rather than relying solely on paraphrased data, even enabling self-evolution by generating analogous data. More discussion see AppendixC

2.2 RAG for Problem Solving

Retrieval-Augmented Generation (RAG) systems enhance the performance of LLMs by integrating an external search engine for knowledge retrieval (Khandelwal et al., 2019; Lewis et al., 2020; Gao et al., 2023). When a user poses a question, the RAG system first retrieves relevant knowledge fragments through the search engine and then uses these answers along with the original query to generate an answer. To address more complex problems, such as mathematical reasoning, some works incorporate the reasoning capabilities of LLMs into the RAG framework, achieving retrieval-augmented reasoning (RAR) (Melz, 2023). For example, IRCoT (Trivedi et al., 2022) combines RAG with multi-step CoT by using the question and previous reasoning steps as queries, retrieving relevant documents to generate the next reasoning step. RAT (Wang et al., 2024b) generates

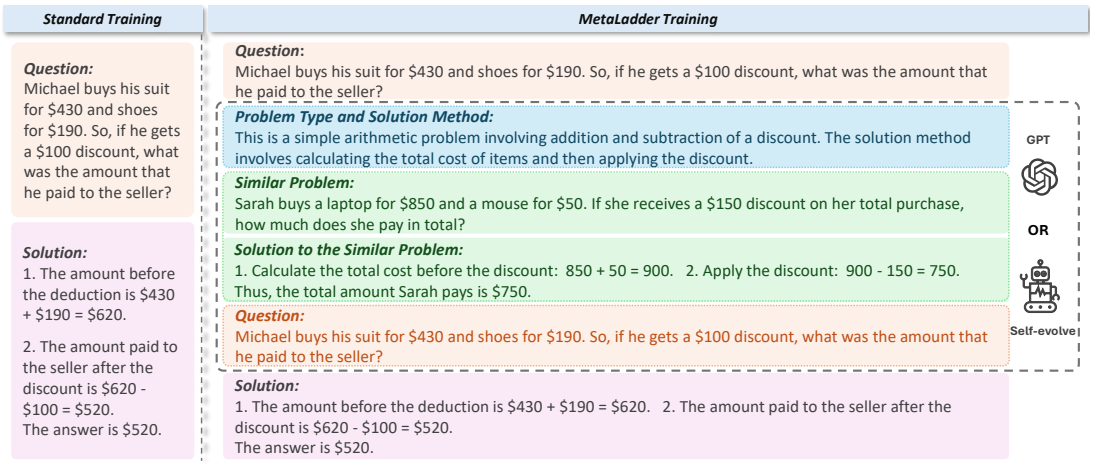

Figure 2: Overview of the MetaLadder framework for generating reflective data. The process starts with the original problem Q, followed by the problem type and solution method S, and the generation of analogous problems Q' and solutions C'. Afterward, the original problem Q is reintroduced to prompt the model to restate the problem. These components are then combined with the solution C of the original problem Q to form the training data.

a complete CoT first and then refines each reasoning step iteratively using RAG. Recent work, such as Search-o1 (Li et al., 2025), extends the RAG paradigm by applying it to o1-like models, further enhancing the model's reasoning capabilities. While RAG emphasizes enhancing the model's performance by retrieving external knowledge without updating the model's parameters, our approach differs in that it internalizes analogical reasoning through model fine-tuning, allowing the model to generate reflective information during reasoning without relying on external data.

3 MetaLadder

We first introduce the overall MetaLadder framework for enhancing mathematical problem-solving (Section 3.1). Then we present each our method in detail by explaining the generation of reflective data to guide the model's reasoning process (Section 3.2), describing the composition of training data to activate the model's analogical reasoning (Section 3.3). Besides, we also add a self-evolve process in our framework to enable the model's ability to autonomously generate data for self-improvement (Section 3.4). Finally, we also incorporate a shortcut inference mechanism for fast and effective generation (Section 3.5).

3.1 Overview

The overview of MetaLadder framework is illustrated in Fig. 2. Given the original data consisting of a target problem Q and its solution C. We first generate additional reflective data by synthesizing an problem analysis and solution strategy S for the target problem, then along with analogous problem Q' and its corresponding solution C', which are structurally or semantically similar to the original data. Moreover, we introduce a problem-restating mechanism, where the target problem Q is inserted before the final solution C to enhance the model's understanding of the target problem. After training on the generated data sequence $QSQ'C'QC$, the model is able to recall analogous problems, restate the target problem and apply analogical reasoning to find the final solution. Notably, because of MetaLadder's analogical reasoning capability, the model can autonomously generate similar problems for training itself, which enables the self-evolution ability of the model. We now introduce the details in the following sections.

3.2 Reflective Data Generation

MetaLadder improves the model's mathematical problem-solving by incorporating reflective data during training. This approach simulates human learning, encouraging the model to recall and reflect on analogous problems, using their solutions to inform reasoning on the target problem. To achieve this, the model requires structured guidance to first understand the problem-solving strategy, then recall analogous problems, and finally apply the solutions from those analogous problems to the current task. Therefore, the reflective data consists of three key components:

1) Problem Type and Solution Method S. Each problem is categorized into a mathematical domain, with an explanation of the relevant concepts and methods (e.g., *"This is a simple arithmetic problem involving addition and subtraction of a discount. The solution method involves calculating the total*

cost of items and then applying the discount."). This helps the model grasp the problem-solving framework for future similar problems.

2) Analogous Problem Q'. The model generates an analogous problem by modifying the context, numbers, or variables, while keeping the core structure intact, offering a new learning context. For example, as shown in Fig 2: *"Original Problem: Michael buys his suit for $430 and shoes for $190. So, if he gets a $100 discount, what was the amount that he paid to the seller?"* The generated analogy problem is then: *"Similar Problem: Sarah buys a laptop for $850 and a mouse for $50. If she receives a $150 discount on her total purchase, how much does she pay in total?"*

3) Solution to Analogous Problem C'. The model provides a solution for the analogous problem, reinforcing the application of similar strategies across different problems. For instance, the solution to the above analogy problem is: *"1. Calculate the total cost before the discount: 850 + 50 = 900. 2. Apply the discount: 900 - 150 = 750. Thus, the total amount Sarah pays is $750."*

To generate the above annotation data, we carefully design prompts for data generation. The detailed prompts are provided in Table 11.

3.3 Analogical Reasoning Activation

To activate the analogical reasoning capabilities of the model, we compose the training data with the generated reflective data in a format as described in Figure 1 (d). In traditional approaches, as shown in Fig. 1, CoT directly generates the solution C from the problem Q, while question and answer augmentation directly rephrase the problem and solution into $Q'C'$. RefAug (Zhang et al., 2024b) adds additional reflection R after the solution C. In contrast, our MetaLadder introduces an analogical reasoning process $SQ'C'Q$ between Q and C, involving the generation of analogous problem and the transfer of knowledge from similar problem to the target problem. Specifically, our analogical reasoning process $SQ'C'Q$ consists of:

Problem Type and Solution Method S. This is the problem and solution analysis, which is generated in Section 3.2.

Analogical Problem and Corresponding Solution $Q'C'$. This is the analogy problem and the solution of the analogy problem, which is also generated in Section 3.2.

Problem-restating mechanism. After presenting $Q'C'$, we reintroduce the original problem Q. This step ensures that the model revisits the target problem and apply the knowledge gained from solving the analogous problem, engaging in analogical reasoning and transferring the learned solution to solve the original problem.

Overall, the enhanced training data format is $QSQ'C'QC$: Original Problem $\rightarrow$ Problem Type and Solution Strategy $\rightarrow$ Analogous Problem and its Solution $\rightarrow$ Original Problem and its Solution. By training on the above formatted data, we aim to improve the model's mathematical reasoning by activating its analogical reasoning from similar problem and solution with a deeper understanding of the target problem.

3.4 Analogical Self-evolution

Similar to other works (Zelikman et al., 2022; Luong et al., 2024; Guan et al., 2025), after training, our model gains the ability to autonomously generate analogous problems that are related to the target problem. This capability facilitates a self-evolving data augmentation process, where the model can iteratively bootstrap its own dataset. Specifically, after making predictions for a given problem, the model is used to generate new problem instances based on its own outputs. These generated problems, being structurally or conceptually similar to the original ones, are then fed back into the training loop for further refinement. The self-evolution process significantly enhances the model's ability to expand its knowledge base. As the model generates new problem instances, it creates novel variations of existing problems by modifying key components—such as numbers, variables, or contexts—while preserving the underlying structure. This process not only reinforces the model's understanding of the problem-solving strategies but also improves its generalization ability across new, previously unseen problem types.

3.5 Shortcut Inference

During training, the model learns to implicitly encode analogy-problem reasoning schemas via the $QSQ'C'QC$ paradigm, where explicit generation of analogical problems Q' and their solutions C' establishes robust neural pathways for structural pattern transfer. At inference time, we try to propose a shortcut inference method that enables a streamlined $Q\hat{S}Q\hat{C}$ process that bypasses analogy-problem generation for fast inference. Specifically, after the model generates $\hat{S}$, we force it to directly restate the original problem Q and output the an-

swer $\hat{C}$ by inserting Q after $\hat{S}$. Surprisingly, skipping the $\hat{Q}'\hat{C}'$ generation not only reduces inference cost but also boost the performance significantly (see results in Section 4.2). This clearly demonstrates MetaLadder can successfully learn analogy knowledge transfer through analogy problem solving.

4 Experiments

4.1 Experimental Setup

Datasets/Benchmarks. We use the training sets from GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) for our experiments. The augmented parts in each problem (problem type and solution method, the analogy problem, the solution to the analogy problem) are generated by GPT-4o-mini (OpenAI, 2024a). The details of the data generation process can be found in the Appendix A.1. For evaluation, besides the test sets from GSM8K and MATH as in-distribution evaluation, we also include out-of-distribution test sets from ASDiv (Miao et al., 2020), College Math (Tang et al., 2024), GaoKao (Chinese College Entrance Exam) En 2023 (Liao et al., 2024) and DM (Saxton et al., 2019) for verification.

Training and Evaluation. We primarily use two popular LLMs for our experiments, covering the general focused LLM and the math-focused LLM: the widely used Llama3-8B (Zhang et al., 2024a) and DeepSeekMath-7B (Shao et al., 2024). For a fair comparison, all the models are trained for 1 epoch. During inference, greedy decoding is applied to get the outputs. As for evaluation metrics, we report Pass@1 accuracy for all the models and baselines. More experimental details can be found in the Appendix A.2 and A.3.

Baselines. We first introduce the following baseline methods that we adopt for comparison, which are also shown in Figure 1: **(i) CoT:** The original CoT data from GSM8K and MATH, which is the standard setting (Figure 1(a)). **(ii) AnalogyAug:** This combines our augmented analogy problem/-solution (as used in MetaLadder) with the original CoT data in a *batch-level* training, making the training data twice as large as the original data (also known as question&answer augmentation, shown in Figure 1(b)). **(iii) RefAug:** A state-of-the-art (SOTA) data augmentation method that enhances model reasoning by *appending* reflective data to the end of the CoT chain (Figure 1(c)).

For our MetaLadder-based settings, we enhance the original CoT data as described in Section 3.3 to train **1) MetaLadder**, the basic setting, and derive the following variant **2) MetaLadder + Self-evolve**: We use the MetaLadder model after one round of training to greedily sample one data point from each problem, and then filter out the correct answers to add back to the training data used in the first round. Since we have generated the analogy problem Q' for helping solve the target problem Q, we add a reverse training setting in this section. Specifically, we train **3) MetaLadder + Reverse**, which simply swaps the target problem Q and the analogous problem Q' to be a new training sample, expanding the training data to twice the size of the original data. Besides, we also experiment on a variant, **4) MetaLadder + Reverse + Self-evolve**, which further incorporates self-evolve data. Besides, for our MetaLadder-related experiments, we also include the shortcut inference mechanism as a comparison. The "-cut" suffix after the method name indicates the use of the shortcut inference.

4.2 Main Results

Our experimental results, as shown in Table 1, reveal the following key findings:

MetaLadder Outperforms Strong Methods. The main experimental results demonstrate the effectiveness of the MetaLadder framework across multiple mathematical benchmarks. MetaLadder consistently outperforms baseline methods on both in-domain and out-of-domain datasets. On the LLaMA3-8B and DeepSeekMath-7B models, MetaLadder surpasses CoT by an average accuracy improvement of 6.7 (36.1 vs. 42.8) and 10.3 (47.3 vs. 57.6) points, respectively, and outperforms RefAug by 5.4 and 9.5 points in accuracy. These results highlight MetaLadder's significant advantage in enhancing the model's reasoning ability, particularly when tackling challenging mathematical problems.

MetaLadder Enhances the Model Beyond Batch-Level Augmentation. Compared to AnalogyAug, which performs question and answer augmentation at the batch level, MetaLadder achieves higher scores on GSM8K and comparable performances on MATH. On both models, MetaLadder improves by 4.7 and 6.5 points, respectively. This suggests that MetaLadder effectively enhances the model's reasoning ability by activating its analogical reasoning capabilities, rather than simply adding more augmented data. Particularly, MetaLadder + Reverse, which swaps the target problem with analogous problems to double the dataset, outperforms

Method	# Sample	In-Domain		Out-of-Domain				Average
		GSM8K	MATH	ASDiv	CM	GE	DM	
LLaMA3-8B								
CoT (Wei et al., 2022)	15K	61.5	19.4	73.2	16.6	19.5	26.3	36.1
RefAug (Zhang et al., 2024b)	15K	59.7	20.3	74.3	17.6	18.4	23.2	35.6
RefAug+CoT (Zhang et al., 2024b)	30K	64.9	21.8	74.4	16.5	21.0	24.8	37.4
AnalogyAug	30K	63.8	22.6	76.1	18.2	20.8	27.2	38.1
MetaLadder	15K	66.2	22.4	76.7	17.2	23.9	29.7	39.4
MetaLadder-cut	15K	69.4	24.0	78.8	18.2	26.0	29.4	41.0
MetaLadder+Self-evolve	22K	66.5	24.8	76.6	19.1	26.8	31.7	40.9
MetaLadder+Self-evolve-cut	22K	**73.5**	26.0	**81.6**	**20.3**	24.4	30.7	**42.8**
MetaLadder+Reverse	30K	71.5	25.6	77.2	19.0	24.7	29.0	41.2
MetaLadder+Reverse+Self-evolve	44K	71.1	26.7	77.2	19.2	27.3	31.4	42.2
MetaLadder+Reverse+Self-evolve-cut	44K	70.5	**27.7**	77.2	19.4	**28.3**	**31.9**	42.5
DeepSeekMath-7B								
CoT (Wei et al., 2022)	15K	64.2	34.3	81.5	31.4	29.4	43.0	47.3
RefAug (Zhang et al., 2024b)	15K	67.4	35.1	83.0	30.3	35.6	43.5	49.2
RefAug+CoT (Zhang et al., 2024b)	30K	67.2	34.9	80.4	31.8	29.4	45.1	48.1
AnalogyAug	30K	67.7	38.9	83.2	30.5	36.9	49.6	51.1
MetaLadder	15K	69.4	38.6	85.9	32.6	37.4	48.4	52.1
MetaLadder-cut	15K	71.5	40.0	87.1	35.2	40.5	49.1	53.9
MetaLadder+Self-evolve	23K	70.5	39.3	86.3	33.3	35.1	50.8	52.6
MetaLadder+Self-evolve-cut	23K	74.1	41.3	87.3	**35.7**	40.5	50.5	54.9
MetaLadder+Reverse	30K	72.3	40.5	85.2	32.1	37.4	51.3	53.1
MetaLadder+Reverse+Self-evolve	46K	72.6	40.7	85.2	33.7	38.2	51.9	53.7
MetaLadder+Reverse+Self-evolve-cut	46K	**76.6**	**45.6**	**89.3**	35.1	**43.1**	**54.8**	**57.6**

Table 1: Accuracy on in-domain and out-of-domain mathematical benchmarks. The **bold** and underlined values represent the first and second best performances, respectively. CM, GE, DM denotes College Math, Gaokao En 2023, DeepMind-Mathematics, respectively. '# Sample' denotes the data size.

AnalogyAug by 3.1 and 2.0 points on the two models, respectively. This further validates the effectiveness of MetaLadder's data generation strategy.

MetaLadder's Self-Evolution Boosts Model Performance. Self-evolution provides further improvements in model performance, with significant gains observed across all datasets. After one round of self-evolution, MetaLadder+Self-evolve improves by 1.5 points on LLaMA3-8B and 0.5 points on DeepSeekMath-7B, demonstrating that a single round of self-training effectively enhances the model's reasoning ability. Additionally, MetaLadder+Reverse+Self-evolve improves by 1.0 points and 0.6 points on LLaMA3-8B and DeepSeekMath-7B in accuracy across all datasets, respectively, confirming the benefits of data augmentation through problem swapping. Ultimately, MetaLadder + Reverse + Self-evolve exceeds CoT by 6.1 points and RefAug by 4.8 points on LLaMA3-8B and achieves the best score of 53.7 on DeepSeekMath-7B.

Shortcut Inference Reduces Inference Cost and Improves Performance. Surprisingly, shortcut inference not only reduces the inference cost by skipping the analogy problem reasoning during inference (e.g., as shown in Table 2, DeepSeek-MetaLadder-cut on MATH achieves 1343.74 seconds, faster than DeepSeek-MetaLadder's 2181.13 seconds and close to CoT's 1253.26 seconds), but also boosts the model performance by a clear margin, e.g., 1.6 accuracy points on LLaMA3-8B and 1.8 points on DeepSeekMath-7B. The results demonstrate MetaLadder has transferred analogy problem-solving knowledge.

These results underscore MetaLadder's outstanding performance in solving both in-domain and out-of-domain problems, further validating the framework's effectiveness in enhancing mathematical problem-solving abilities and cross-domain generalization. Through its data augmentation and self-evolution strategies, MetaLadder not only excels in reasoning tasks within known domains but also demonstrates strong adaptability when facing unfamiliar data.

4.3 Ablation on Components

To thoroughly evaluate the contribution of each component in the MetaLadder framework, we conducted an ablation study that systematically examined the impact of its core elements, where "w/o Strategy", "w/o Analogy", and "w/o Restate" refer to the absence of the problem type and solution method, analogy meta-problem, and problem restating mechanism, respectively.

As shown in Table 2, we observed that removing any of these components resulted in a significant performance drop across both datasets. Specifically, removing the strategy component caused

Method	GSM8K	MATH	Average
w/o Strategy	64.9	22.2	43.6
w/o Analogy	64.7	21.0	42.9
w/o Restate	61.6	22.0	41.8
MetaLadder	**66.2**	**22.4**	**44.3**

Table 2: Ablation study on GSM8K and MATH, where w/o Strategy, w/o Meta-problem, and w/o Restate refer to the absence of the problem type and solution method, analogy meta-problem, and problem restating mechanism, respectively.

a 0.7% average decrease in performance on both datasets, indicating that the strategy is important in guiding the model toward more accurate and efficient solutions. Furthermore, excluding the analogy meta-problem or the problem restating mechanism led to even greater performance degradation, with decreases of 1.4 and 2.5 points, respectively. This highlights the crucial role of these components in enhancing the model's reasoning ability.

4.4 Experiment on Qwen2.5-Math-7B

To evaluate the scalability of MetaLadder, we conducted experiments on a strong math-oriented model: **Qwen2.5-Math-7B**. We compared the performance of standard CoT and our MetaLadder method, using the same 15K training samples. Table 14 summarizes the results.

Even when applied to a strong backbone like Qwen2.5-Math-7B, MetaLadder achieves consistent improvements across all benchmarks, with an average gain of **+2.9 points** over standard CoT. This demonstrates the generalizability of the analogical reasoning framework and its potential benefits on increasingly capable models.

5 Analysis

5.1 The Amount of MetaLadder Data

We first investigate the impact of different amounts of MetaLadder-enhanced data on model performance across two models. As shown in Figure 3(a), in our experiments, we gradually replace the original CoT data with MetaLadder-enhanced data. As the proportion of MetaLadder-augmented data increased, the model's performance steadily improves, reaching its peak when the original data is completely replaced with augmented data. This demonstrates the effectiveness and scalability of the augmentation method.

5.2 Multi-round Self-evolve

To further explore the impact of multiple rounds of self-evolution, we examine the performance improvements with each additional round of self-training. As shown in Figure 3(b), the performance of MetaLadder steadily improves with the increasing number of self-evolution rounds. On the LLaMA3-8B model, after one round of self-evolution, the average accuracy across all testsets increases from 39.4 to 40.9. After two rounds, the accuracy further improves to 41.7, with an increase of 2.3 points. On the DeepSeekMath-7B model, after two and three rounds of self-evolution, the average accuracy increases to 53.1 and 53.9, respectively, with improvements of 1.0 points and 1.8 points. This demonstrates that multi-round self-evolution significantly enhances the model's reasoning ability. Further experimental results can be found in the Table 7.

5.3 Impact of Reflection on Train and Test

To investigate the impact of reflection on model performance, we compared the effects of performing reflection before and after generating the final answer. We train Pre-RefAug by shifting the reflection component of RefAug to the training stage, and Post-MetaLadder by placing the reflection component of MetaLadder after answer generation.

As shown in Table 3, Pre-RefAug outperformed RefAug by 1.4 and 2.1 points on two models, while MetaLadder achieved scores 1.3 and 0.3 points higher than Post-MetaLadder. Our results demonstrate that allowing reflection before providing the final answer leads to further performance improvement. This suggests that reflection data not only enhances the model's reflective capabilities during training but also guides its reasoning during testing in an in-context learning manner, ultimately boosting performance.

5.4 Combine with Other Augmentation

We further investigate the effectiveness of our method when combined with other augmentation approaches. We compare the performance before and after applying the MetaLadder method on two augmented training sets: 1) In the AugCoT method, we enhance the original solutions using prompts that are almost identical to those used for generating MetaLadder's analogous data, aiming to better align the distribution and complexity of the answers with the meta-problem. 2) We also use data gen-

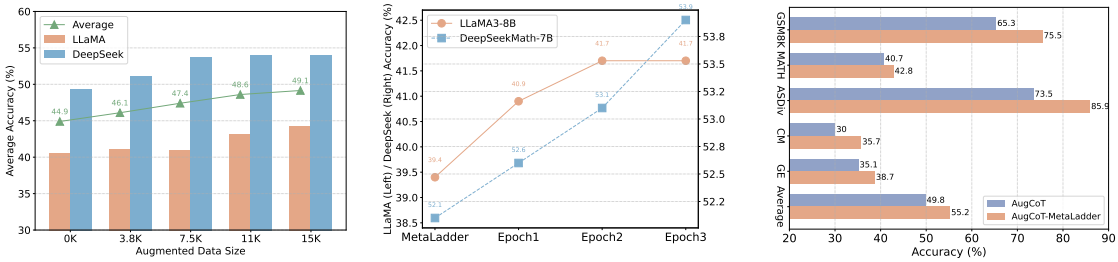

Figure 3: **Left**: Performance of enhancing different amounts of original data. **Middle**: Results of 3 rounds of evolution on LLaMA and DeepSeek. **Right**: Combination of MetaLadder with AugCoT method.

Method	Train → Inference	LLaMA		DeepSeek	
		GSM8K	MATH	GSM8K	MATH
RefAug	$QCR \rightarrow QC$	59.7	20.3	67.4	35.1
Pre-RefAug	$QRC \rightarrow QRC$	**61.0**	**21.7**	**69.1**	**37.5**
Post-Metaladder	$QQCSQ'C' \rightarrow QQC$	64.1	21.9	**70.3**	37.1
MetaLadder	$QSQ'C'QC \rightarrow QSQ'C'QC$	**66.2**	**22.4**	69.4	**38.6**

Table 3: Model performance when reflections placed before and after the answer.

erated by the mainstream MetaMath method (Yu et al., 2024), which employs both question rephrasing and answer augmentation to generate a more diverse set of problems.

The results in Figure 3(c) demonstrate the effectiveness of the MetaLadder framework in improving the performance of DeepSeekMath-7B on mathematical benchmark tasks. Compared to the AugCoT baseline, AugCoT-MetaLadder shows a more significant improvement across all datasets, with an average increase of 5.4 points in accuracy. This result suggests that the performance boost brought by MetaLadder is not mainly because of the enhanced data only, but rather due to the structured, reflective data generation process within the MetaLadder framework.

Additionally, as shown in Table 8, we present the experimental results based on the augmented data from MetaMath20K and MetaMath40K. After being enhanced with MetaLadder, MetaMath20K-MetaLadder outperforms the original MetaMath20K with an average performance improvement of 1.3 points, which highlights the positive impact of MetaLadder on model accuracy.

These results further suggest that combining MetaLadder with other augmentation methods can more effectively boost the model's performance, demonstrating the potential of structured data augmentation in improving mathematical reasoning.

5.5 Case Study

To have a more straightforward understanding of the advantage of our MetaLadder, we show some cases and make discussions in this section.

In case 16 (in Appendix, with more cases in Appendix D.1.), we examine the performance of MetaLadder and the standard CoT approach on a root-finding problem. The problem is: *"Compute $a + b + c$, given that a, b, and c are the roots of $\frac{1}{x} + 5x^2 = 6x - 24$."* CoT solves the problem directly, focusing on converting the equation to $4x^3 - 5x^2 + 5x - 1 = 0$ and immediately applying Vieta's formulas to obtain $a + b + c = \frac{5}{4}$. This approach involves minimal abstraction and no explicit reuse of methods from other problems. It delivers a direct result but provides limited insight into the generalization of the solution method. In contrast, the MetaLadder framework explicitly identifies this task as part of a general class of polynomial-related problems and uses Vieta's formulas as the backbone of the solution. In addition, it builds a reusable methodology, highlights similarities with the original problem ($\frac{1}{x} + 5x^2 = 6x - 24$), and emphasizes systematic computation techniques. The solution still delivers an accurate result ($a + b + c = \frac{5}{4}$), but it also builds a more structured understanding of the type of problem.

This example highlights how MetaLadder can improve the accuracy and reliability of the model in solving conceptually rich and broadly applicable problems, further underscoring the value of reflective reasoning in enhancing the model's overall problem-solving capabilities.

5.6 Discussion on Analogical Data Quality

To validate the quality of the automatically generated analogical problems and annotations, we conducted both similarity analysis and correctness

verification.

Similarity Evaluation. We computed cosine similarity, Jaccard distance, and Levenshtein distance between each generated analogical problem and its original counterpart (see Table 9). These metrics indicate strong structural and semantic similarity.

To further test the effect of analogical variation, we additionally constructed a set of more divergent analogical problems by modifying the generation prompt with instructions such as "please change the nouns and add extra conditions to the problem." Notably, Table 9 also shows that even when the analogical problems deviate more from the original in surface form, they can still lead to improved performance, suggesting that moderate divergence in analogical examples can be beneficial for generalization.

Correctness Verification. We further assessed the correctness of generated solutions and the reliability of problem type and strategy annotations using `LLaMA3.1-70B-Instruct`. The model was prompted to answer whether each solution or annotation was correct. The results are in Table 4.

Diversity Visualization. To evaluate the diversity of the generated analogical problems, we performed t-SNE visualization on the embeddings of the original and generated problems and their solutions (See Figure 4). The result show that the generated analogical problems broadly cover the semantic space of the original dataset, indicating good diversity.

Model Performance. Furthermore, we find that the trained model itself can generate structurally and semantically similar problems during inference, demonstrating that the analogical patterns are learnable. Representative examples are provided in Appendix D.1 for qualitative validation.

5.7 Discussion on Cost

While MetaLadder relies on LLMs to generate analogical data, we find the cost and latency to be manageable in practice. Additionally, comparable results can be achieved using open-source models, making the method more accessible to the broader research community.

Cost and Time Analysis. Using GPT-4o-mini, which charges \$0.60 per million output tokens and \$0.15 per million input tokens, we generated 15K analogical samples (approx. 2K tokens each for input and output) for under \$20. The entire generation process completed in 1.5 hours. This demonstrates that high-quality data generation is both cost- and time-efficient even on commercial APIs.

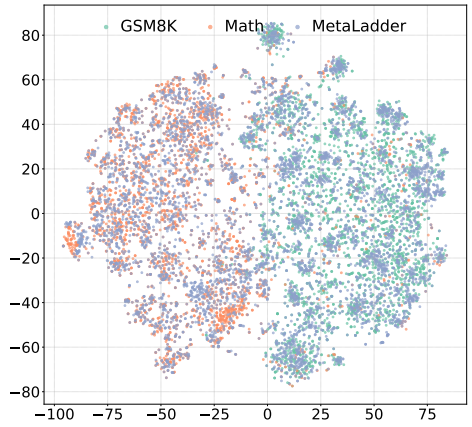

Figure 4: Problem distribution of GSM8K, Math, MetaLadder.

Local Model Alternatives. To reduce dependency on commercial APIs, we also tested local generation using `LLaMA3.1-70B-Instruct` (4×A100 GPUs). We replicated the generation process and trained both CoT and MetaLadder models using data from LLaMA and DeepSeek. Results are shown in Table 15. Across both backbones, MetaLadder outperforms standard CoT. Notably, the average gain is +1.77 points for LLaMA and +2.53 for DeepSeek, indicating that our framework remains effective even when using open-source models for data generation.

6 Conclusion

In this paper, we introduce MetaLadder, a novel framework that enhances the mathematical problem-solving abilities of LLMs. By explicitly prompting the model to reflect on analogical problems and their solutions, MetaLadder enables it to transfer reasoning across similar tasks, mimicking human learning. Additionally, our problem-restating mechanism further enhances the model's reasoning accuracy. Experimental results across multiple mathematical benchmarks demonstrate that MetaLadder significantly improves LLM performance, surpassing both standard Chain-of-Thought (CoT) methods and other state-of-the-art approaches. Our work highlights the importance of integrating analogical reasoning and meta-cognitive strategies into LLMs for complex reasoning tasks.

Limitations

Although the MetaLadder framework has shown promising progress in mathematical problem solving, there are still some limitations worth further exploration and improvement. For instance, the performance of MetaLadder relies on the quality of the analogy problems Q' and their corresponding solutions C'. During the generation of analogy problems, data augmentation biases may be introduced, especially when the analogy problems are generated with a strong reliance on certain problem types or solution methods. The model may overfit to common problem types or solutions present in the training data, potentially impacting its ability to generalize to novel problems. Future work could focus on improving the quality of generated analogy problems, enhancing the model's ability to handle a wider variety of problem types, and further investigating the trade-off between inference efficiency and reasoning depth.

Acknowledgements

This work is supported by Shanghai Artificial Intelligence Laboratory.

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Kahneman Daniel. 2011. Thinking, fast and slow. *Macmillan*.

Jenny Lynn Daugherty and Nathan Mentzer. 2008. Analogical reasoning in the engineering design process and technology education applications.

Run-Ze Fan, Xuefeng Li, Haoyang Zou, Junlong Li, Shwai He, Ethan Chern, Jiewen Hu, and Pengfei Liu. 2024. Reformatted alignment. *Arxiv preprint*, 2402.12219.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *ICML 2023*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. 2024. Common 7b language models already possess strong math capabilities. *arXiv preprint arXiv:2403.04706*.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. *Preprint*, arXiv:2501.05366.

Minpeng Liao, Wei Luo, Chengxi Li, Jing Wu, and Kai Fan. 2024. Mario: Math reasoning with code interpreter output – a reproducible pipeline. *Preprint*, arXiv:2401.08190.

Haoxiong Liu, Yifan Zhang, Yifan Luo, and Andrew Chi-Chih Yao. 2024. Augmenting math word problems via iterative question composing. *Preprint*, arXiv:2401.09003.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. *Preprint*, arXiv:2401.08967.

Eric Melz. 2023. Enhancing llm intelligence with arm-rag: Auxiliary rationale memory for retrieval augmented generation. *Preprint*, arXiv:2311.04177.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984.

OpenAI. 2024a. Gpt-4o mini: advancing cost-efficient intelligence.

OpenAI. 2024b. Learning to reason with llms.

Josh OpenAI, Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. *Preprint*, arXiv:1904.01557.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Y Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024. Mathscale: Scaling instruction tuning for mathematical reasoning. In *ICML*. OpenReview.net.

Qwen Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. In *NeurIPS*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Stella Vosniadou. 1988. Analogical reasoning as a mechanism in knowledge acquisition: A developmental perspective. *Center for the Study of Reading Technical Report; no. 438*.

Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2024a. Mathcoder: Seamless code integration in llms for enhanced mathematical reasoning. In *The Twelfth International Conference on Learning Representations*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. 2024b. Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *Preprint*, arXiv:2403.05313.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. 2024. On memorization of large language models in logical reasoning. *arXiv preprint arXiv:2410.23123*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

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

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488. Curran Associates, Inc.

Di Zhang, Jiatong Li, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. 2024a. Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b. *arXiv preprint arXiv:2406.07394*.

Zhihan Zhang, Tao Ge, Zhenwen Liang, Wenhao Yu, Dian Yu, Mengzhao Jia, Dong Yu, and Meng Jiang. 2024b. Learn beyond the answer: Training language models with reflection for mathematical reasoning. In *EMNLP*, pages 14720–14738. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Experimental Details

A.1 Annotation Details

Data annotation was performed using the GPT-4o-mini-2024-07-18 model API with a sampling temperature of 0.7. The full prompt used for annotation is shown in Block 11.

In Experiment 8, we use prompt shown in Block 12 to rephrase the original solutions in GSM8K and MATH.

A.2 Train Details

Model training was conducted using the LLaMA Factory [1] on 8 NVIDIA A100 GPUs. We train all models for one epoch with a batch size of 128, using the AdamW optimizer (Loshchilov, 2017) with a learning rate of 5e-6 and cosine learning rate decay. The training prompt is shown in Block 13.

A.3 Evaluation Details

GSM8K (Cobbe et al., 2021): GSM8K consists of grade-school arithmetic tasks with relatively low difficulty, primarily used to evaluate basic mathematical reasoning abilities. The testset includes 1319 basic math word problems, covering simple arithmetic operations such as addition, subtraction, multiplication, and division. Compared to other more complex datasets, the problems in GSM8K are straightforward and suitable for testing a model's performance on solving basic mathematical problems.

MATH (Hendrycks et al., 2021): The MATH testset contains 5000 challenging competition-level math problems. These problems are designed to be complex and require the model to possess higher-level mathematical reasoning capabilities, far surpassing the simpler problems found in GSM8K. MATH spans multiple mathematical domains, including algebra, geometry, and number theory, making it an ideal benchmark for evaluating a model's performance on complex mathematical reasoning tasks.

ASDiv (Miao et al., 2020): ASDiv (Academia Sinica Diverse MWP Dataset) is a diverse English math word problem dataset intended to evaluate the capabilities of various MWP solvers. This dataset includes 2,305 math word problems that cover a wide range of language patterns and problem types, offering more diversity than existing MWP datasets. It includes problems commonly found in elementary school and is annotated with problem types and grade levels to help assess the difficulty and complexity of each problem.

Gaokao 2023 EN (Liao et al., 2024): Gaokao 2023 EN contains 385 math problems from the 2023 Chinese National College Entrance Examination (Gaokao), which are primarily high school-level open-ended problems. These problems cover a wide range of mathematical topics and include content taught during high school in China. The Gaokao EN2023 dataset is designed to assess students' ability to apply mathematical reasoning in real-world situations, containing both basic problems and more complex ones. It serves as an important benchmark for evaluating models' performance on Gaokao-style math problems.

CollegeMath (Tang et al., 2024): The CollegeMath dataset includes 2,818 college-level math problems extracted from 9 textbooks, spanning 7 mathematical domains such as linear algebra and differential equations. CollegeMath is designed to test a model's ability to reason across diverse mathematical topics, with a particular focus on generalization to complex mathematical reasoning tasks at the college level. The problems are more difficult, making the dataset well-suited for evaluating a model's ability to solve advanced mathematical problems.

DeepMind-Mathematics (Saxton et al., 2019): The DeepMind-Mathematics test set containing 1000 problems from a variety of problem types, based on a national school mathematics curriculum (up to age 16), designed to assess basic mathematical reasoning across different domains. The dataset generates question and answer pairs of varying types, generally at school-level difficulty, and aims to test the mathematical learning and algebraic reasoning abilities of learning models.

All model evaluation was carried out using the framework provided at https://github.com/ZubinGou/math-evaluation-harness/tree/main with zero-shot evaluation, greedy sampling and a maximum generation length of 2048 tokens.

To validate that the improvement from the shortcut method was not due to avoiding the truncation of MetaLadder's output, we also conducted the main experiment with a maximum length of 4096 tokens, and no significant changes in the metrics were observed.

B More Experiments

[1] https://github.com/hiyouga/LLaMA-Factory

Component	Total	Correct
Solution to Analogical Problem	14,973	14,185 (94.7%)
Problem Type and Strategy	14,973	14,364 (95.9%)

Table 4: Verification of solution correctness and annotation reliability using LLaMA3.1-70B-Instruct.

Method	1st Turn	2nd Turn	3rd Turn
CoT	60.05	31.26	23.44
MetaLadder	**65.55**	**34.45**	**23.75**

Table 5: Performance (%) on multi-turn reasoning using MathChat-FQA.

Method	3ppl	4ppl	5ppl	6ppl
CoT	0.58	0.42	0.18	0.10
MetaLadder	0.62	0.44	0.22	0.12

Table 6: Performance on Logic Puzzles from the KK dataset.

Method	# Sample	In-Domain		Out-of-Domain				Average
		GSM8K	MATH	ASDiv	CM	GE	DM	
LLaMA3-8B								
MetaLadder	15K	66.2	22.4	76.7	17.2	23.9	29.7	39.4
MetaLadder+Self-evolve	22K	66.5	24.8	76.6	19.1	26.8	31.7	40.9
MetaLadder+Self-evolve2	30K	68.9	25.1	79.3	19.1	25.7	32.2	41.7
MetaLadder+Self-evolve3	38K	69.2	25.9	77.9	19.8	26.0	31.6	41.7
DeepSeekMath-7B								
MetaLadder	15K	69.4	38.6	85.9	32.6	37.4	48.4	52.1
MetaLadder+Self-evolve	23K	70.5	39.3	86.3	33.3	35.1	50.8	52.6
MetaLadder+Self-evolve2	31K	71.9	39.8	86.1	33.6	37.9	49.2	53.1
MetaLadder+Self-evolve3	40K	72.2	40.6	86.3	34.0	38.7	51.5	53.9

Table 7: **Accuracy of self-evolution on in-domain and out-of-domain mathematical benchmarks**. The **bold** and underlined values represent the first and second best performances, respectively. CM, GE, DM denotes College Math, Gaokao En 2023, DeepMind-Mathematics, respectively.

	GSM8K	MATH	ASDiv	CM	GE	DM	Average
MetaMath20K	71.1	38.4	84.5	32.0	32.5	46.9	50.9
MetaMath20K-MetaLadder	72.9	39.8	86.5	31.6	34.3	44.9	51.7
MetaMath40K	73.9	38.5	84.6	32.0	34.3	48.2	51.9
MetaMath40K-MetaLadder	**75.7**	40.0	**87.0**	30.5	34.8	45.8	52.3
AugCoT	65.3	40.7	73.5	30.0	35.1	**54.0**	49.8
AugCoT-MetaLadder	75.5	**42.8**	85.9	**35.7**	**38.7**	52.8	**55.2**

Table 8: **DeepSeekMath-7B performance on two augmented datasets**. MetaMath20K constructed by uniformly sampling half of the data from MetaMath40K, and AugCoT representing original solutions rephrased by 4o-mini to match the style of analogous data used in MetaLadder.

Method	Cos	JD	LD	GSM8K	MATH
Original Problem	1.00	0.00	0.00	64.0	21.1
Analogous Problem	0.93	0.48	101.00	66.2	22.4
Enhanced Analogous Problem	0.91	0.61	131.30	65.3	23.3

Table 9: The impact of repeating the original problem and using analogous problems with greater divergence on model performance. Cos refers to Cosine Similarity, JD refers to Jaccard Distance, and LD refers to Levenshtein Distance.

Method	GSM8K	MATH
LLaMA-CoT	61.86	749.26
LLaMA-MetaLadder	172.62	1479.85
LLaMA-MetaLadder-cut	111.26	1081.58
DeepSeek-CoT	65.9	1253.26
DeepSeek-MetaLadder	210.15	2181.13
DeepSeek-MetaLadder-cut	113.96	1343.74

Table 10: Total time cost of inference on the whole GSM8K and MATH testsets, in seconds. Our MetaLadder, combined with shortcut reasoning, significantly reduces inference time, achieving speeds close to CoT.

B.1 Experiment on MathChat

Although MetaLadder is primarily designed for single-turn tasks, we investigate its potential extension to multi-turn settings using the MathChat dataset. Multi-turn reasoning requires the model to integrate prior reasoning steps, track evolving context, and maintain coherence across turns. To adapt MetaLadder for this setting, we retain the analogy and restatement components at each turn, with the previous output provided as part of the prompt context.

Experimental Setup. We evaluate on the Follow-up Question Answering (FQA) task from MathChat. For each turn, the previous answer is prepended to the new question. We compare CoT and MetaLadder under the same few-shot setting.

Observations. MetaLadder outperforms CoT in the first and second turns, indicating that analogical reasoning and restatement mechanisms provide early-turn benefits. Performance converges in later turns, possibly due to increased context complexity.

Future Work. We plan to extend training to explicitly support multi-turn reasoning by:

- Incorporating analogies from prior turns to support temporal or causal reasoning.
- Restating prior conclusions alongside current questions to reinforce contextual memory and prevent drift.

These improvements aim to better capture the dynamics of sustained interactions in multi-turn tasks.

B.2 Experiment on KK

To investigate generalizability, we extended MetaLadder to Logic Puzzles using the KK (Xie et al., 2024) dataset, which contains 2,000 samples for 3ppl and 4ppl. We applied Qwen2.5-7B-Math for this experiment. The results are summarized in the Table 6.

The preliminary results suggest that MetaLadder also improves performance in the logical puzzle reasoning, helping the model to "clarify conditions" and "transfer structures," much like it does for mathematical reasoning tasks.

C More Discussion on Related Work

Several recent methods have explored ways to improve chain-of-thought (CoT) reasoning through prompt engineering, data augmentation, or reflective learning. While we acknowledge these valuable contributions, we clarify here how MetaLadder differs fundamentally from these approaches.

Beyond the *"Problem → CoT → Answer"* Paradigm. Works such as *RefAug* and *Critique Fine-Tuning* introduce reflective strategies and post-hoc rationales that deviate from the standard CoT pipeline. However, MetaLadder takes a different route by incorporating explicit analogical reasoning and problem restatement *directly into the training data*, enabling the model to generalize reasoning patterns through structured learning, not just reflective inference.

Comparison to Plan-and-Solve Prompting. The Plan-and-Solve method focuses on optimizing zero-shot prompts by asking the model to plan before solving(Wang et al., 2023). This process is applied only at test time and does not alter training data. In contrast, MetaLadder uses a teacher model to extract problem-solving strategies and encodes these strategies into the training samples themselves, providing the model with richer structured supervision.

Comparison to Question Augmentation. (Li et al., 2024) augment training data by generating additional similar questions, thereby improving robustness through data diversity. However, this process does not involve explicit reasoning over the analogous questions. In MetaLadder, the model is not only exposed to similar problems, but also guided to reflect on their structures and solutions before answering the target problem—enabling analogical reasoning transfer rather than mere data expansion.

Key Novelty. The core novelty of MetaLadder lies in its integration of three synergistic components into training:

- **Strategy learning:** Explicit guidance on problem type and solution method.
- **Analogical reasoning:** Use of structurally/semantically similar problems with CoT solutions as scaffolds.
- **Problem restatement:** Prompted rephrasing of the original question to deepen understanding before solution.

Together, these elements simulate a human-like learning pattern and lead to consistently improved

performance across multiple datasets and model scales. We believe this structured training formulation goes beyond prior reflective or augmentation-based techniques, and offers a new direction for reasoning-aware data design.

D More cases

D.1 Inference Cases

We present more cases in this section to show the generated predictions of our MetaLadder trained model.

D.2 Error Analysis

Below, we highlight a typical failure case, analyze common errors, and discuss potential directions for improvement.

Failure Example. Consider the problem shown in Table 17. In this case, the model correctly generated a similar problem, but made calculation errors in both solutions. The solution to the similar problem should have been 26 pints ($3.25 \times 4 \times 2$), and the solution to the original problem should have been 20 pints ($2.5 \times 4 \times 2$).

Other Common Error Types. In addition to errors in calculation, we observe the following common issues in the model's reasoning process:

- When the solution to the analogous problem is incorrect, the same mistake often propagates into the solution for the original problem.
- In some cases, the model fails to correctly identify or apply the problem type and solution method (referred to as "strategy"), leading to incorrect reasoning from the outset.
- We also observe enumeration errors or missing steps during multi-step reasoning, especially in more complex tasks.

Performance Across Problem Types. MetaLadder improves over standard CoT in algebra problems but shows more limited gains in calculus and number theory. This suggests that analogical transfer is more effective for concrete reasoning tasks and less so for abstract domains requiring deeper understanding.

Directions for Improvement. To address these issues, we propose:

- **Step enumeration and coverage:** Enhance the model's ability to break down complex problems, especially in advanced domains like calculus and number theory.
- **Self-correction:** Introduce fine-tuning examples where the model corrects its own errors, fostering reflective reasoning.

Data Annotation Prompt

You are a professional math teacher, and your goal is to teach your student to learn a given math problem. Identify the type of the given problem and the name of the solution method. Then, generate a similar problem with its solution.

Example 1:
Original Problem:
Youngsville had a population of 684 people. The town had a growth spurt and the population increased by 25% then they witnessed that 40% of the population moved away. What is the current population?

Solution to the Original Problem:
The town had 684 people, and then had a 25% growth spurt, so the population increased by 684 * 0.25 = 171 people. This increase brought the population to 684 + 171 = 855 people. 40% of the population moved away, so 855 * 0.40 = 342 people moved away. The new population is 855 - 342 = 513 people. The answer is 513.

Type of Problem and Solution Method:
This is a consecutive percentage change problem. The solution method involves applying the growth factor first, then applying the reduction factor. The key to solve the problem is to understand the concept of relative increase and decrease percentages.

Similar Problem:
A company started with 800 employees. After one year, the workforce increased by 20%, and then 30% of the new workforce decided to quit due to relocation. How many employees remain at the company?

Solution to the Similar Problem:
1. Initial employees: 800.
2. Increase by 20%: 20% of 800 is 160, so the workforce grows to 800 + 160 = 960.
3. Then, 30% of these 960 employees quit: 30% of 960 is 288, so 288 employees leave.
4. Remaining employees: 960 - 288 = 672. Therefore, the final number of employees is 672.

Example 2:
Original Problem:
Solve the equation (x - 99)(x - 101) = 8.

Solution to the Original Problem:
Let t=x-100. Then the equation becomes (t - 1)(t + 1) = 8, which transforms into t^2 - 1 = 8. Therefore, t = 3 or t = -3, and accordingly we get x = 97 or x = 103.

Type of Problem and Solution Method:
This is a quadratic equation problem solved by introducing a substitution to simplify the expression. The solution method involves recognizing a suitable substitution that transforms the equation into a simpler form. The key to solving the problem is understanding how to use algebraic manipulation to facilitate solving equations.

Similar Problem:
Solve the equation (x - 50)(x - 52) = 4.

Solution to the Similar Problem:
1. Notice that the middle point between 50 and 52 is 51, so let t = x - 51.
2. Then (x - 50) = (t + 1) and (x - 52) = (t - 1).
3. The equation becomes (t + 1)(t - 1) = 4 $\Rightarrow$ t^2 - 1 = 4 $\Rightarrow$ t^2 = 5 $\Rightarrow$ t = $\sqrt{5}$ or t = $-\sqrt{5}$.
4. Substituting back: x = t + 51.
- If $t = \sqrt{5}$, then $x = 51 + \sqrt{5}$.
- If $t = -\sqrt{5}$, then $x = 51 - \sqrt{5}$.
Hence, the solutions are $x = 51 + \sqrt{5}$ or $x = 51 - \sqrt{5}$.

Original Problem:
{problem}

Solution to the Original Problem:
{solution}

Table 11: Annotation Prompt.

Answer Augmentation Prompt

You are a professional math teacher, and your goal is to teach your student to learn a given math problem. Please augment the original solution with a step-by-step breakdown.

Example 1:
Problem: Youngsville had a population of 684 people. The town had a growth spurt and the population increased by 25% then they witnessed that 40% of the population moved away. What is the current population?

Original Solution:
The town had 684 people, and then had a 25% growth spurt, so the population increased by 684 * 0.25 = 171 people. This increase brought the population to 684 + 171 = 855 people. 40% of the population moved away, so 855 * 0.40 = 342 people moved away. The new population is 855 - 342 = 513 people. The answer is 513.

Augmented Solution:
1. Initial population: 684.
2. Growth by 25%: 25% of 684 is 171, so the population increases to 684 + 171 = 855.
3. 40% of 855 moved away: 40% of 855 is 342, so 342 people moved away.
4. The remaining population is 855 - 342 = 513.
Therefore, the final population is 513.

Example 2:
Problem:
Solve the equation (x - 99)(x - 101) = 8.

Original Solution:
Let $t = x - 100$. Then the equation becomes $(t - 1)(t + 1) = 8$, which transforms into $t^2 - 1 = 8$. Therefore, $t = 3$ or $t = -3$, and accordingly we get x = 97 or x = 103.

Augmented Solution:
1. Notice that the middle point between 99 and 101 is 100, so let t = x - 100.
2. Then $(x - 99) = (t + 1)$ and $(x - 101) = (t - 1)$.
3. The equation becomes $(t + 1)(t - 1) = 8 \Rightarrow t^2 - 1 = 8 \Rightarrow t^2 = 9 \Rightarrow t = 3 or t = -3$.
4. Substituting back: x = t + 100.
- If t = 3, then x = 100 + 3.
- If t = -3, then x = 100 - 3.
Hence, the solutions are x = 103 or x = 97.

Problem:
{problem}

Original Solution:
{solution}

Table 12: Answer Augmentation Prompt.

Training and Evaluation Prompt

Instruction:
{question}

Response: Let's think step by step.

Table 13: Training and Evaluation Prompt.

Method	#Sample	GSM8K	MATH	ASDiv	CM	GE	DM
CoT	15K	82.4	59.1	90.1	43.4	52.5	60.0
MetaLadder	15K	**85.9**	**62.4**	**92.0**	**44.5**	52.2	**67.7**

Table 14: Comparison of CoT and MetaLadder on Qwen2.5-Math-7B. All methods are trained with 15K samples.

Model	GSM8K	MATH	ASDiv	CM	GE	DM	Avg.
LLaMA-CoT	61.5	19.4	73.2	16.6	19.5	26.3	36.08
LLaMA-MetaLadder	**65.0**	**20.9**	**77.7**	**16.7**	**20.0**	**26.8**	**37.85**
DeepSeek-CoT	64.2	34.3	81.5	31.4	29.4	43.0	47.30
DeepSeek-MetaLadder	**68.1**	**35.8**	**82.5**	30.1	**36.9**	**45.6**	**49.83**

Table 15: Comparison of CoT and MetaLadder using data generated by `LLaMA3.1-70B-Instruct`.

Example D.1: Case in Math: Polynomial Roots Problem

Question: Compute $a+b+c$, given that a, b, and c are the roots of $\frac{1}{x}+5x^2=6x-24$.

CoT: We can rewrite the equation as $5x^3-6x^2-x+24=0$. By the rational root theorem, the only possible rational roots are ±1, ±2, ±3, ±4, ±6, ±8, ±12, and ±24. We can check that $x=2$ is a root, so $5x^2-13x+12=0$. The roots of this quadratic are $x=\frac{13\pm\sqrt{169-240}}{10}=\frac{13\pm\sqrt{-71}}{10}=\frac{13\pm i\sqrt{71}}{10}$. Thus, the sum of the roots is $2+\frac{13+i\sqrt{71}}{10}+\frac{13-i\sqrt{71}}{10}=3$.
The answer is $\boxed{3}$.

MetaLadder: This is a polynomial equation problem where we need to find the sum of the roots of a polynomial. The solution method involves rearranging the equation to a standard form and using Vieta's formulas to find the sum of the roots.

Similar Problem: Compute $a+b+c$, given that a, b, and c are the roots of $\frac{1}{x}+4x^2=5x-15$.

Solution to the Similar Problem: We can rewrite the equation as $4x^3-5x^2+5x-1=0$. By Vieta's formulas, the sum of the roots is $\frac{5}{4}$.

Original Problem: ...

Solution to the Original Problem: We can rewrite the equation as $5x^3-6x^2+6x-1=0$. By Vieta's formulas, the sum of the roots is $\frac{6}{5}$. Thus, the answer is $\boxed{\frac{6}{5}}$.

Table 16: Model prediction example on MATH.

Example D.2: Error Case

Question: There are two pints in one quart and four quarts in one gallon. How many pints are there in $2\frac{1}{2}$ gallons of milk?

Model Prediction: This is a unit conversion problem involving the conversion of gallons to pints. The solution method involves converting the mixed number of gallons to a decimal and then multiplying by the conversion factor to find the total number of pints.
Similar Problem: There are two pints in one quart and four quarts in one gallon. How many pints are there in $3\frac{1}{4}$ gallons of juice?
Solution to the Similar Problem:
1. Convert $3\frac{1}{4}$ gallons to a decimal: $3\frac{1}{4}$ = 3.25 gallons.
2. Convert gallons to pints: 3.25 gallons $\times$ 4 quarts/gallon $\times$ 2 pints/quart = 13 pints.
Thus, there are $\boxed{13}$ pints in $3\frac{1}{4}$ gallons of juice.
Solution to the Original Problem:
$2\frac{1}{2}$ gallons is the same as 2.5 gallons.
There are 2.5 gallons $\times$ 4 quarts/gallon $\times$ 2 pints/quart = $\boxed{10}$ pints in $2\frac{1}{2}$ gallons of milk.
Final Answer: $\boxed{10}$.

Table 17: Error Case.

Example D.3: Case in MATH

Question: Exits on a highway are numbered consecutively from 1 to 50. The distance from exit 41 to exit 50 is 100 km. If each exit is at least 6 km from the next exit, what is the longest possible distance, in kilometers, between exit 47 and exit 48?

CoT: The distance between exit 41 and exit 50 is 100 km. If each exit is at least 6 km from the next exit, the distance between exit 47 and exit 48 is at least 12 km. The longest possible distance between exit 47 and exit 48 is 12 km.
Final Answer: $\boxed{12}$.

MetaLadder: This is a problem involving the arrangement of objects (exits) with constraints on the distances between them. The solution method involves setting up inequalities based on the given conditions and then solving for the maximum distance between two specific exits.

Similar Problem:
Exits on a highway are numbered consecutively from 1 to 40. The distance from exit 31 to exit 40 is 80 km. If each exit is at least 5 km from the next exit, what is the longest possible distance, in kilometers, between exit 37 and exit 38?

Solution to the Similar Problem:
1. The distance from exit 31 to exit 40 is 80 km, which means the distance from exit 31 to exit 32 is at least 5 km.
2. The distance from exit 32 to exit 33 is at least 5 km, and so on.
3. The total distance from exit 31 to exit 38 is:

- From exit 31 to exit 32: 5 km
- From exit 32 to exit 33: 5 km
- From exit 33 to exit 34: 5 km
- From exit 34 to exit 35: 5 km
- From exit 35 to exit 36: 5 km
- From exit 36 to exit 37: 5 km
- From exit 37 to exit 38: 5 km

4. Therefore, the total distance from exit 31 to exit 38 is:

$$5 \text{ km} \times 7 = 35 \text{ km}$$

5. The distance from exit 38 to exit 40 is:

$$80 \text{ km} - 35 \text{ km} = 45 \text{ km}$$

6. The distance from exit 37 to exit 38 is:

$$45 \text{ km} - 5 \text{ km} = 40 \text{ km}$$

Thus, the longest possible distance between exit 37 and exit 38 is **40 km**.

Original Problem:
Exits on a highway are numbered consecutively from 1 to 50. The distance from exit 41 to exit 50 is 100 km. If each exit is at least 6 km from the next exit, what is the longest possible distance, in kilometers, between exit 47 and exit 48?

Solution to the Original Problem:
The distance from exit 41 to exit 50 is 100 km, which means the distance from exit 41 to exit 42 is at least 6 km.
The distance from exit 42 to exit 43 is at least 6 km, and so on.
The total distance from exit 41 to exit 48 is:

- From exit 41 to exit 42: 6 km
- From exit 42 to exit 43: 6 km
- From exit 43 to exit 44: 6 km
- From exit 44 to exit 45: 6 km
- From exit 45 to exit 46: 6 km
- From exit 46 to exit 47: 6 km
- From exit 47 to exit 48: 6 km

Therefore, the total distance from exit 41 to exit 48 is:
6 km * 7 = 42 km
The distance from exit 48 to exit 50 is:
100 km - 42 km = 58 km
The distance from exit 47 to exit 48 is:
58 km - 6 km = 52 km
Final Answer: $\boxed{52}$.

Example D.4: Case in MATH: Trigonometry

Question: Find $\csc(-120^\circ)$.

CoT: We have $\csc(-120^\circ) = \frac{1}{\sin(-120^\circ)} = \frac{1}{\sin(120^\circ)} = \frac{1}{\frac{\sqrt{3}}{2}} = \boxed{\frac{2\sqrt{3}}{3}}$.

Final Answer: $\boxed{\frac{2\sqrt{3}}{3}}$.

MetaLadder: This is a trigonometric function evaluation problem. The solution method involves using the properties of the cosecant function and the reference angle to find the value of the cosecant at a given angle.

Similar Problem:
Find $\csc(-135^\circ)$.

Solution to the Similar Problem:
1. The angle -135° is in the third quadrant, where the cosecant function is negative.
2. The reference angle for -135° is $135^\circ - 180^\circ = -45^\circ$.
3. The cosecant of -45° is $\csc(-45^\circ) = -\sqrt{2}$.
4. Therefore, $\csc(-135^\circ) = -\sqrt{2}$.

Thus, the final answer is $\boxed{-\sqrt{2}}$.

Original Problem:
Find $\csc(-120^\circ)$.

Solution to the Original Problem:
The angle -120° is in the third quadrant, where the cosecant function is negative. The reference angle for -120° is $120^\circ - 180^\circ = -60^\circ$. The cosecant of -60° is $\csc(-60^\circ) = -\frac{2}{\sqrt{3}}$. Therefore, $\csc(-120^\circ) = \boxed{-\frac{2}{\sqrt{3}}}$.

Final Answer: $\boxed{-\frac{2}{\sqrt{3}}}$.

Example D.5: Case in GSM8K: Algebra

Question: A football team played 22 games. They won 8 more than they lost. How many did they win?

CoT: If they won 8 more than they lost, then they lost 22-8=14 games. So they won 22-14=8 games.
Final Answer: $\boxed{8}$.

MetaLadder: This is a problem involving a system of equations based on relationships between wins, losses, and total games played. The solution method involves setting up equations based on the given information and solving for the unknowns.

Similar Problem:
A basketball team played 30 games. They won 10 more than they lost. How many did they win?

Solution to the Similar Problem:
1. Let the number of games won be w.
2. The number of games lost is $w - 10$.
3. The total number of games played is 30, so we have the equation:

$$w + (w - 10) = 30$$

4. Simplifying this gives:

$$2w - 10 = 30$$

5. Adding 10 to both sides:

$$2w = 40$$

6. Dividing by 2:

$$w = 20$$

Thus, the team won 20 games.

Original Problem:
A football team played 22 games. They won 8 more than they lost. How many did they win?

Solution to the Original Problem:
Let the number of games won be w. The number of games lost is $w - 8$. The total number of games played is 22, so we have the equation:

$$w + (w - 8) = 22$$

$$2w - 8 = 22$$

$$2w = 30$$

$$w = 15$$

Final Answer: $\boxed{15}$.

Example D.6: Case in Math

Question: Simplify $[\frac{1}{1-\tan^2 x}+\frac{1}{1-\cot^2 x}]$.

CoT: We have$[\frac{1}{1-\tan^2 x}+\frac{1}{1-\cot^2 x}]=\frac{1}{\cos^2 x-\sin^2 x}+\frac{1}{\sin^2 x-\cos^2 x}=\frac{1}{\cos^2 x-\sin^2 x}-\frac{1}{cos^2x-sin^2x}=0$]. Final Answer: $\boxed{0}$

MetaLadder: This is a trigonometric simplification problem. The solution method involves using the Pythagorean identities to express the given expression in terms of sine and cosine, and then simplifying the resulting expression.

Similar Problem:
Simplify $[\frac{1}{1-\sin^2 x}+\frac{1}{1-\cos^2 x}]$.

Solution to the Similar Problem:
Using the Pythagorean identity, we have $1-\sin^2 x=\cos^2 x$ and $1-\cos^2 x=\sin^2 x$. Thus,$\frac{1}{1-\sin^2 x}+\frac{1}{1-\cos^2 x}=\frac{1}{\cos^2 x}+\frac{1}{\sin^2 x}=\frac{\sin^2 x+\cos^2 x}{\sin^2 x\cos^2 x}=\frac{1}{\sin^2 x\cos^2 x}$.

Original Problem:
Simplify $\left[\frac{1}{1-\tan^2 x}+\frac{1}{1-\cot^2 x}\right]$.

Solution to the Original Problem
1. Start with the given expression: $[\frac{1}{1-\tan^2 x}+\frac{1}{1-\cot^2 x}]$.
2. Use the Pythagorean identity: $[1-\tan^2 x=\frac{1}{\cos^2 x}$ and $1-\cot^2 x=\frac{1}{\sin^2 x}]$.
3. Substitute these identities into the expression: $\frac{1}{\frac{1}{\cos^2 x}}+\frac{1}{\frac{1}{\sin^2 x}}=\cos^2 x+\sin^2 x$.
4. Since $\cos^2 x+\sin^2 x=1$, we have: $\cos^2 x+\sin^2 x=1$.
5.Final Answer: $\boxed{1}$

Table 18: Case in MATH.

Example D.7: Case in MATH: Exponentiation

Question: Compute $(2^3)(5^5)$ in your head.

CoT: We have $(2^3)(5^5) = (2^3)(5^3)(5^2) = (10^3)(25) = 100000$.
Final Answer: $\boxed{100000}$.

MetaLadder: This is a multiplication problem involving powers of different bases. The solution method involves simplifying the expression by breaking it down into smaller parts and then multiplying the results.

Similar Problem:
Compute $(3^2)(4^3)$ in your head.

Solution to the Similar Problem:
1. First, calculate 3^2:

$$3^2 = 9$$

2. Next, calculate 4^3:

$$4^3 = 4 \times 4 \times 4 = 16 \times 4 = 64$$

3. Now, multiply the results:

$$9 \times 64 = 576$$

Thus, the final answer is $\boxed{576}$.

Original Problem:
Compute $(2^3)(5^5)$ in your head.

Solution to the Original Problem:
We have $2^3 = 8$ and $5^5 = 5 \cdot 25 \cdot 25 = 5 \cdot 625 = 3125$.
Thus, the answer is $8 \cdot 3125 = \boxed{25000}$.
Final Answer: $\boxed{25000}$.

Table 19: Case in MATH: Exponentiation

Example D.8: Case in GSM8K: Basic Algebra

Question: A treasure hunter found a buried treasure chest filled with gems. There were 175 diamonds, 35 fewer rubies than diamonds, and twice the number of emeralds than the rubies. How many of the gems were there in the chest?

CoT: There were $175 - 35 = 140$ rubies.
There were $140 \times 2 = 280$ emeralds.
There were $175 + 140 + 280 = 600$ gems in the chest.
Final Answer: $\boxed{600}$.

MetaLadder: This is a problem involving basic arithmetic operations (addition and multiplication) to find the total number of items based on given relationships. The solution method involves calculating the quantities of each type of gem based on the relationships provided.

Similar Problem:
A collector found a collection of antique coins. There were 200 silver coins, 40 fewer gold coins than silver coins, and three times the number of copper coins than the gold coins. How many coins were there in total?

Solution to the Similar Problem:
1. Silver coins: 200.
2. Gold coins: $200 - 40 = 160$.
3. Copper coins: $160 \times 3 = 480$.
4. Total coins: $200 + 160 + 480 = 840$.
Therefore, the total number of coins in the collection is $\boxed{840}$.

Original Problem:
A treasure hunter found a buried treasure chest filled with gems. There were 175 diamonds, 35 fewer rubies than diamonds, and twice the number of emeralds than the rubies. How many of the gems were there in the chest?

Solution to the Original Problem:
There were $175 - 35 = 140$ rubies.
There were $140 \times 2 = 280$ emeralds.
There were $175 + 140 + 280 = 595$ gems in the chest.
Final Answer: $\boxed{595}$.

Table 20: Case in MATH: Basic Algebra