Adaptive Chain-of-Thought Framework

Saurav Krishnakumar
The University of Auckland
Email: skri486@aucklanduni.ac.nz

***Abstract*—This research investigates the effectiveness of adaptive Chain-of-Thought (CoT) prompting compared to standard uniform CoT approaches in mathematical reasoning tasks. The study addresses a critical gap in current CoT implementations, which apply the same reasoning structure regardless of problem complexity. We developed a three-dimensional difficulty taxonomy that classifies mathematical problems across semantic, procedural, and cognitive complexity dimensions, then implemented an adaptive framework that dynamically selects appropriate CoT strategies based on problem characteristics. The methodology was evaluated on three mathematical reasoning datasets (GSM8K, SVAMP, and ASDiv) using multiple language models, including DeepSeek-R1-Distill-Qwen-1.5B, LLaMA 3.2-1B, and FLAN-T5-base. Results demonstrate that adaptive CoT prompting significantly outperforms standard approaches, particularly for low-complexity problems, with accuracy improvements of up to 20.4 percentage points. The DeepSeek model showed the most consistent gains, while smaller models demonstrated limited effectiveness. However, no problems were classified as high-complexity, limiting the evaluation scope. This work establishes a framework for difficulty-aware prompting in mathematical reasoning, hence showing that tailoring reasoning complexity to problem characteristics can meaningfully improve performance when matched with appropriate model architectures.**

***Index Terms*—Adaptive Chain-of-Thought (CoT) prompting, Multi-dimensional difficulty taxonomy, Mathematical reasoning, Large Language Models (LLMs), Problem complexity classification**

I. Introduction

As mentioned by Sun et al. [1], Math Word Problems (MWPs) are commonly used to evaluate the arithmetic reasoning abilities of language models. While these issues may appear uncomplicated to humans, these models demonstrate remarkable fluency in language generation, their ability to solve multi-step reasoning problems, particularly in mathematical and arithmetic domains has been limited by their tendency to produce answers without explicit reasoning processes.

A notable finding to solve MWPs through LLMs is that employing chain-of-thought (CoT) prompted by Wei et al. [2], along with a language model containing 540 billion parameters, yields performance comparable to task-specific fine-tuned models across multiple tasks. CoT prompting has emerged as a transformative technique that addresses this fundamental challenge by eliciting step-by-step reasoning from LLMs. Rather than generating direct answers, CoT prompting encourages models to articulate their reasoning process through intermediate steps, mimicking human problem-solving strategies With the shift to chain of thought (CoT) prompting mathematical reasoning tasks, most existing implementations continue to adopt a one size fits all strategy, applying the same reasoning structure uniformly across all problems, regardless of their complexity or characteristics. This methodological limitation becomes particularly evident when considering the diverse nature of mathematical problems, which can differ significantly across multiple dimensions. These dimensions include the linguistic complexity and readability of the problem statement, the mathematical operations and number of steps required to reach a solution, and the level of abstract reasoning or domain knowledge involved. Hence, we have not seen many implementations of splitting datasets on a certain taxonomy of difficulty when applying CoT. In datasets that contain problems of varying difficulty levels, it might be assumed that a sophisticated CoT model capable of solving complex arithmetic tasks would naturally perform well on simpler problems too. However, this is not always the case. For example, Zhou et al. [3] proposed the "least to most prompting" strategy, which decomposes complex problems into a sequence of simpler subproblems and solves them step by step. This approach has shown improvements in tasks involving symbolic manipulation and compositional generalisation. Nevertheless, even Zhou [3] acknowledges that such elaborate decomposition is primarily beneficial for more challenging problems. When applied to simpler tasks, it can introduce unnecessary computational load and increase the risk of error propagation. This reveals a fundamental mismatch in current CoT strategies, which is that using overly complex reasoning for simple problems may reduce efficiency and accuracy, while using overly simplistic strategies for complex problems may fail to provide the depth of reasoning required. A straightforward arithmetic problem that requires only basic addition operates quite differently from a multi-step word problem involving ratios, percentages, or conditional logic, yet most current methods apply the same CoT strategy to both. This uniform approach fails to optimise reasoning effectiveness across the full range of problem complexity. Therefore, we can see a clear gap in the lack of a taxonomy of difficulty in arithmetic datasets, and applying a singular CoT framework on them regardless of difficulty is a flaw. This research addresses a critical gap in the current understanding of how reasoning complexity should align with problem difficulty in mathematical domains.

The central research question guiding this investigation is: **How much more effective is an adaptive chain-of-thought (CoT) prompting strategy, guided by a multi-dimensional difficulty taxonomy in mathematical reasoning, compared to standard CoT prompting across problems of varying complexity levels?**

To answer this question, we would need to develop a

framework that could classify mathematical problems across a few dimensions of difficulty and select appropriate reasoning strategies based on problem characteristics. The approach should move beyond static prompting by introducing an adaptive style methodology that tailors reasoning complexity to match problem complexity.

We hypothesise that this difficulty-aware approach will demonstrate better performance, particularly on medium and high-complexity problems where standard CoT often underperforms due to misaligned reasoning complexity, while maintaining efficiency on simpler problems through appropriately scaled reasoning steps.

The primary contribution of this work is the development and validation of the first systematic framework for adaptive Chain-of-Thought prompting in mathematical reasoning, which combines a three-dimensional difficulty taxonomy with dynamic strategy selection to optimise reasoning complexity based on problem characteristics. Our framework classifies problems across multiple dimensions, then automatically selects appropriate CoT strategies ranging from minimal reasoning for simple problems to decomposition-based approaches for complex ones. Through evaluation on GSM8K, SVAMP, and ASDiv datasets using multiple language models, we demonstrate that this adaptive approach can achieve substantial performance improvements over standard CoT prompting with accuracy gains of up to 20.4 percentage points on low-complexity problems. This work establishes a new paradigm for difficulty-aware prompting that moves beyond uniform reasoning strategies toward context-sensitive frameworks that can be applied across diverse mathematical reasoning domains.

II. Literature Review

The foundation of structured reasoning in large language models was established by Wei [2], whose work introduced Chain-of-Thought (CoT) prompting as a method to elicit step-by-step reasoning from LLMs. Their approach demonstrated that providing examples of multi-step reasoning before presenting new problems enabled LLMs to generate intermediate reasoning steps, leading to improved performance on complex tasks. One issue in the study is that they are applying uniform reasoning strategies regardless of individual problem characteristics. This limitation will be addressed in our research to avoid a one-size-fits-all approach.

Building upon these foundations, Kojima et al. [4] addressed the context limitation through their zero-shot CoT approach using simple prompting phrases like "Let's think step by step." Their methodology demonstrated that explicit examples were not necessary to elicit structured reasoning in sufficiently large models, achieving comparable performance gains compared to Wei's approach while eliminating context window consumption. However, it perpetuated the fundamental issue of treating all problems with uniform prompting, missing optimisation opportunities for problems of varying difficulty levels. This will provide inspiration for solving low-level problems due to its improved performance from the standard CoT prompting, but also a simplistic approach.

The recognition that individual reasoning paths may contain errors led Wang et al. [5] to shift focus toward reasoning path diversity through their Self-Consistency method. By sampling multiple reasoning paths and applying majority voting, they achieved significant improvements over standard CoT. However, their approach treated all sampled reasoning paths equally in the voting mechanism, without considering that certain structural patterns might be inherently more reliable for specific problem types, and provided no clear guidance on optimal reasoning path quantities for different complexity levels. This uniform treatment of reasoning paths, regardless of problem characteristics, represents a key limitation that our multi-dimensional difficulty taxonomy addresses by enabling complexity-aware reasoning strategy selection.

The relationship between reasoning complexity and performance was directly investigated by Fu et al. [6], whose "complexity-based voting" approach explicitly modeled the interaction between problem difficulty and optimal reasoning structure. Unlike previous uniform approaches, their work demonstrated that optimal reasoning complexity varies systematically with problem difficulty, simpler problems benefit from concise reasoning while complex problems require more elaborate reasoning paths. This study validates the purpose of our methodology to deviate from a one-size-fits-all approach and look into approaches for different problem difficulties. We draw inspiration from their complexity metrics, primarily focused on quantitative measures like step count and semantic complexity. However, unlike Fu [6], we also focus on qualitative aspects such as reasoning strategy selection.

A significant advancement in addressing the easy-to-hard generalisation challenge was introduced by Zhou et al. [3] through their least-to-most prompting strategy. Recognising that traditional CoT approaches perform poorly on problems harder than examples shown in prompts, they proposed a two-stage approach: first decomposing complex problems into simpler subproblems, then solving them sequentially using answers from previously solved subproblems. Their experimental results demonstrated remarkable improvements over standard CoT, achieving 99% accuracy on the compositional generalisation benchmark SCAN using only 14 examples, compared to 16% with CoT prompting. This idea by Zhou [3] will be useful for us to tackle hard problems as his strategy of breaking them down will provide a good accuracy.

A crucial limitation across existing CoT implementations is the lack of comprehensive frameworks for classifying problem difficulty, which prevents systematic matching of reasoning strategies to problem characteristics. Zhang et al. [7] implicitly recognised this through their semantic similarity clustering using Sentence-BERT encodings and K-means clustering, acknowledging that different problem structures benefit from different reasoning approaches. On the other hand Zheng et al. [8] proposed a more quantitative approach, classifying GSM8K problems according to the number of expressions in reference solutions, though this approach failed to capture other dimensions of difficulty such as cognitive complexity or required background knowledge. These unidi-

mensional approaches highlight the need for comprehensive multi-dimensional difficulty taxonomies that consider semantic, procedural, and cognitive aspects rather than relying solely on structural or quantitative features.

More importantly, the challenge of systematically assessing problem difficulty extends beyond AI research into educational assessment theory. In a study, Benedetto et al. [9] presents Question Difficulty Estimation from Text (QDET) has emerged as a specialised field addressing "question calibration", which looks into the idea of using Natural Language Processing to estimate question difficulty as numerical or categorical. Benedetto [9] provided comprehensive surveys demonstrating that NLP techniques can effectively estimate difficulty in educational settings. This approach directly informs our methodology's use of categorical difficulty bins (low, medium, high) derived from normalised numerical complexity scores ranging from 1-5.

Pongsakdi et al. [10] aimed to deepen the understanding of the associations of linguistic and mathematical word problem characteristics affecting difficulty. They provide evidence that text complexity, cognitive understanding of the text, and procedural requirements to solve the problems all contribute to problem difficulty. Therefore, from the ideas we explored from Zhang [7] and Zheng [8] along with Pongsakdi [10], our methodology considers semantic complexity as a crucial dimension.

The broader NLP literature on text complexity assessment provides crucial methodological validation for automated difficulty assessment. Modern approaches by Vajjala et al. [11] validate the use of lexical diversity, syntactic complexity, and reading difficulty metrics through analysis of word length and sentence structure as reliable complexity indicators. Their work demonstrates that linguistic features can effectively capture cognitive processing demands, supporting our use of similar metrics in the semantic complexity dimension.

Crossley et al. [12] advanced this field by demonstrating that linguistic variables related to cognitive reading processes contribute significantly more to readability prediction than traditional surface-level formulas. Their research identified three key correlates with psycholinguistic theory: lexical coreferentiality, syntactic sentence similarity, and word frequency measures. These findings validate our multi-dimensional approach to complexity assessment, as they demonstrate that effective difficulty estimation requires consideration of multiple linguistic and cognitive factors rather than simple surface-level metrics.

III. Methodology

The full code of this report is here.[1]

A. Research Question

With several past methods and prompting techniques established, the research question guiding this investigation is to find how much more effective is an adaptive chain-of-thought (CoT) prompting strategy, guided by a multi-dimensional difficulty taxonomy in mathematical reasoning, compared to standard CoT prompting across problems of varying levels.

This question investigates whether tailoring the structure and complexity of reasoning steps to the difficulty level of each problem leads to measurable improvements in the performance of large language models (LLMs) on arithmetic tasks. The core idea is to move beyond a one-size-fits-all CoT prompting strategy by developing a difficulty-aware framework. This involves evaluating mathematical problems across multiple dimensions such as the operations required, linguistic complexity, and cognitive challenge and classifying them into difficulty tiers. The adaptive CoT approach will then match the reasoning prompt complexity to the estimated problem difficulty.

By comparing this adaptive strategy with a uniform CoT prompting baseline, the study aims to assess the effectiveness of difficulty-sensitive reasoning prompts. This will not only provide a quantitative evaluation of performance differences but also offer insight into how difficulty-aware reasoning impacts LLM capabilities across varying levels of problem complexity.

B. Multi-Dimensional Problem Difficulty Taxonomy

The first methodological idea addresses the limitation identified in existing literature regarding the lack of comprehensive frameworks for classifying arithmetic problem difficulty. Our methodology develops a three-dimensional taxonomy that would take natural language processing as this can analyse the language to look for features in a question such as mathematical operations involved (e.g., multi-step operations, use of division or percentages), Linguistic complexity (e.g., length of problem statements, syntactic depth, or presence of domain-specific jargon), and cognitive challenge (e.g., problems requiring understanding of more abstract topics like ratios or percentages versus basic addition). We believe that these are plausible elements that would make up a problem and scoring based on these features would give a fair level of difficulty. The first category that we will look at is the Semantic Complexity (S). This dimension analyses lexical diversity, syntactic structure, reading difficulty and conditional language patterns. This is important because sentence structure, length, and wording all contribute to how challenging a problem is to read and interpret. By evaluating these linguistic features, we can better understand how much effort is required to comprehend the problem statement before any mathematical reasoning begins.

The semantic complexity score is computed as:

$$S = \frac{L + Y + R + C}{4} \tag{1}$$

Where:

- **Lexical diversity score** (L):

$$L = \frac{\text{unique words}}{\text{total words}}$$

  Measures vocabulary variety, excluding stop words like "a", "an", and "the".

[1] https://github.com/SauravK12/Compsci_703.git

- **Syntactic complexity score** (Y):

$$Y = \text{avg\_word\_length} + \left(\frac{\text{comma\_count}}{\text{word\_count}}\right)$$

Reflects sentence intricacy. Longer words and more clauses (indicated by commas) imply greater complexity.

- **Reading difficulty** (R):

$$R = 1 + \left(\frac{\text{avg\_words\_per\_sentence}}{10}\right) + (\text{difficult\_word\_ratio} \times 4)$$

Where "difficult words" are those with more than six characters.

- **Conditional language complexity** (C): A binary or weighted score for the presence of phrases such as "if", "then", "unless". These increase interpretive difficulty.

The next category to consider is Procedural Complexity (P). This dimension evaluates the mathematical structure of the problem, specifically the types of operations involved, the sequence of steps required, and how interdependent those steps are. This aspect is crucial because it reflects the actual problem-solving workload. The operations that need to be performed, their order, and the logical flow between them. By analysing these components, we gain insight into how challenging a problem is from a computational standpoint. For instance, problems requiring multiple operations, advanced mathematical concepts (like fractions or ratios), or a precise order of execution tend to be significantly more difficult. Understanding procedural complexity helps uncover the layers of reasoning required and allows for a more informed assessment of overall difficulty.

The procedural complexity score (P) is computed as:

$$P = \frac{O + T + D}{3}$$

where:

- **Operations count score** (O): Number of distinct mathematical operations (e.g., addition, subtraction, etc.) required in the solution. More operations generally correspond to higher complexity.
- **Operation type complexity** (T): A weighted score that accounts for the difficulty of operation types. For example, basic arithmetic is scored lower, while fractions, ratios, or exponents are scored higher. E.g., basic operations (e.g., addition) count as 1 and advanced operations (e.g., percentages, fractions, ratios, algebra) count as 2.
- **Step dependency score** (D): Estimated number of procedural steps required. This is inferred from the length of the problem and number of operators. Problems that require multiple intermediate steps receive higher scores.

The final dimension is Cognitive Complexity (C), which measures the extent to which a problem relies on deeper levels of reasoning, abstract thinking, and prior domain knowledge. This is important because not all math problems require the same kind of mental processing, some involve straightforward calculation, while others demand cognitive understanding, strategic thinking, and logical inference. Cognitive complexity captures how much abstract or generalisable knowledge is needed to interpret the problem, recognise patterns, and apply appropriate strategies. The Cognitive Complexity score is computed as:

$$C = \frac{K + A + M}{3}$$

where:

- **Domain knowledge requirements** (K): Number of specialised knowledge areas or contexts (e.g., geometry, time, money) required to comprehend and solve the problem. This is determined by searching for domain-specific keywords and counting the number of unique domains present.
- **Abstract reasoning indicators** (A): A qualitative score reflecting the need for abstract reasoning, such as identifying patterns, making generalisations, or interpreting implicit relationships. This is based on the presence of words like "pattern", "rule", "general", "abstract", or "formula".
- **Multi-step dependency complexity** (M): Reflects the sequential nature of the reasoning process. This is inferred by searching for transition words such as "first", "then", "next", and "finally". Problems that require backtracking or managing multiple strands of logic score higher.

When calculating the overall semantic, procedural, and cognitive complexity scores, we average several sub-dimensions, each of which captures a different aspect of problem difficulty. However, these sub-dimensions are often measured on entirely different scales. For instance, lexical diversity is expressed as a proportion, conditional language complexity may be a simple count of certain phrases. Because these measurements vary so widely in scale and units, averaging them directly would produce skewed results. To address this, we normalise all features to a common 1–5 scale using a linear mapping function:

$$\text{score} = \left(\frac{\text{value} - \text{min\_val}}{\text{max\_val} - \text{min\_val}}\right) \times 4 + 1$$

This formula rescales the original value so that it reflects its relative position between the observed minimum and maximum, then stretches that normalised value to a 1–5 range. The subtraction of min_val centers the scale, the division by the range spreads the values across a 0–1 interval, and multiplying by 4 followed by adding 1 shifts the final result into the desired 1–5 interval. This approach ensures that each sub-dimension contributes equally and fairly to the final complexity score, regardless of its original unit or distribution. It preserves proportional differences between values, avoids zero scores and enables meaningful aggregation.

Lastly the overall problem complexity is computed as the arithmetic mean of the three dimensional scores:

$$\text{Overall Complexity} = \frac{S + P + C}{3}$$

We assume all of these parameters to have an equal effect. Hence, we are only averaging them and have not utilised any weights.

Problems are then categorised into complexity tiers:

- Low Complexity: Overall Complexity ≤ 2.5
- Medium Complexity: $2.5 <$ Overall Complexity ≤ 3.5
- High Complexity: Overall Complexity > 3.5

C. Adaptive Framework

The second methodological component addresses the need for systematic evaluation of different CoT reasoning structures on the different levels of difficulty we set. This step is crucial to answering our research question as we need to uncover some methods of suitable CoT methods to build an adaptive framework. Building on the insights from Fu [6] regarding the relationship between reasoning complexity and performance, our framework implements and evaluates multiple CoT strategies: Hence, to create our adaptive framework we will need to create some prompts aligning to different CoT frameworks for varying levels of complexity. Below are the prompts used in the framework inspired from previous literature.

This approach follows the traditional step-by-step reasoning framework introduced by Wei [2], which has become a cornerstone for improving performance in complex reasoning tasks. The prompt encourages the model to explicitly articulate each logical and computational step, reducing the likelihood of skipping over important intermediate inferences. It's especially useful for moderately difficult problems that benefit from a detailed, linear breakdown. The phrasing "step-by-step" sets the expectation for methodical progression, helping to simulate human-like problem solving.

Prompt Template:

> Solve this math problem step-by-step:
> Problem: {problem}
> Step-by-step solution:

This variation trims the explanation down to the essentials. It assumes that the problem at hand doesn't demand verbose reasoning and that a brief chain of logic suffices. This can be useful for simpler problems or for efficiency in large-scale inference settings. The prompt still encourages step-wise logic but in a more concise form, avoiding over-elaboration that could slow down inference or introduce unnecessary complexity.

Prompt Template:

> Solve this math problem using minimal step-by-step reasoning:
> Problem: {problem}
> Let me solve this concisely:

This strategy introduces a modular structure to tackle procedurally complex problems by explicitly breaking them into smaller, more manageable subproblems. The goal is to reduce cognitive load by encouraging the model to isolate different components of the problem before integrating them into a final solution. The prompt guides the model through a logical decomposition process like Zhou [3]: identifying key elements, partitioning the problem, and solving each piece independently.

Prompt Template:

> Solve this math problem by breaking it into subproblems:
> Problem: {problem}
> 1) First, I'll identify the key variables and unknowns.
> 2) Then, I'll break this into smaller subproblems.
> 3) Finally, I'll solve each part and combine the results.

This strategy prompts the model to apply domain-specific knowledge, shortcuts, or mental math techniques. Rather than following a generic reasoning path, the model is encouraged to leverage more strategic thinking such as estimation, common patterns, symmetry, or formula recall to simplify the problem. It emulates how experienced problem-solvers often bypass exhaustive calculations by recognising structure or reusing known methods.

Prompt Template:

> Solve this math problem using mathematical shortcuts and domain-specific techniques:
> Problem: {problem}
> I'll apply relevant math heuristics and techniques:

Now that we have outlined each Chain-of-Thought (CoT) prompting strategy along with its specific strengths, we move toward building a principled framework that can dynamically align these strategies with the varying levels of complexity identified earlier. This represents the adaptive component of our research question of how prompting can become context-sensitive by tailoring reasoning styles to the cognitive, procedural, and semantic demands of individual problems. The core idea is to move beyond a one-size-fits-all CoT formulation and instead adaptively select the most effective reasoning structure based on a problem's estimated difficulty profile. This allows the model to conserve resources on simpler tasks by using concise reasoning, while deploying deeper, more structured reasoning on more complex ones, thereby optimising accuracy.

The algorithm for the Adaptive Strategy takes as input a math problem and a taxonomy vector representing its semantic (S), procedural (P), and cognitive (C) complexity scores, each scaled between 1 and 5. A composite score is computed by averaging these three dimensions to reflect the overall difficulty calculated earlier in the Multi-Dimensional Problem Difficulty Taxonomy. Below is the pseudo code for how the adaptive framework will select the CoT strategy based on the difficulty score.

Algorithm 1 Adaptive CoT Strategy Selection

Require: problem_text, taxonomy_vector (S, P, C)
Ensure: optimised_prompt

composite_score $\leftarrow \frac{S+P+C}{3}$
if composite_score ≤ 2.0 **then**
  return minimal_cot_prompt(problem_text)
else if composite_score ≤ 3.5 **then**
  if $P > C$ **then**
    return decomposition_cot_prompt(problem_text)
  else
    return standard_cot_prompt(problem_text)
  end if
else
    {High complexity} **if** $S > 4.0$ **then**
    return linguistic_complex_prompt(problem_text)
  else if $P > 4.0$ **then**
    return decomposition_cot_prompt(problem_text)
  else
    return heuristic_cot_prompt(problem_text)
  end if
end if

For low-complexity problems (composite score ¡= 2.0), the Minimal CoT strategy is selected, as these problems typically do not require elaborate reasoning. For medium-complexity problems (2.0 ¡ composite score ¡= 3.5), the algorithm chooses between the Standard and Decomposition CoT strategies depending on whether cognitive or procedural demands dominate. When procedural complexity is higher, the Decomposition CoT is used to modularise the reasoning. If cognitive complexity is greater, the Standard CoT is used to walk through reasoning in a linear fashion.

For high-complexity problems (composite score ¿ 3.5), the algorithm introduces further granularity. If semantic complexity (S) is very high (¿ 4.0), indicating significant linguistic or conditional complexity, a Linguistic Complexity CoT (an extension of the Standard CoT that incorporates parsing and rephrasing) is applied. If procedural complexity (P) is dominant, the Decomposition CoT is chosen again to address multi-step logic. Otherwise, when the complexity is driven by abstract or cognitive reasoning, the Heuristic CoT strategy is employed to leverage domain-specific shortcuts and problem-solving techniques. This adaptive selection mechanism directly supports our research question by enabling a data-driven, difficulty-aware prompting approach. By aligning reasoning strategies with the true structure of the task, our framework provides a means to evaluate whether adaptive CoT strategies can outperform static prompting methods in terms of effectiveness and reasoning fidelity.

D. Experimental design

With all key components defined, this section outlines the experimental pipeline we will use to evaluate our adaptive prompting framework and assess its effectiveness in addressing the research question. Specifically, we aim to determine whether dynamically selecting reasoning strategies based on a multi-dimensional complexity taxonomy leads to improved problem-solving performance compared to standard Chain-of-Thought (CoT) prompting. Datasets and Preprocessing.

Our evaluation begins by applying the multi-dimensional taxonomy to three widely-used mathematical reasoning datasets: GSM8K, ASDiv, and SVAMP, accessed via the Hugging Face API. These datasets were selected to ensure diversity in problem structure and complexity. GSM8K offers grade-school problems, ASDiv includes algebraic and verbal questions across multiple templates, and SVAMP introduces variations that test robustness to problem phrasing and semantics.

Before applying our taxonomy, we perform preprocessing to standardise answer formats. We will only use 100 questions and answers for each dataset to manage the run time. Specifically, we convert textual answers into pure numerical formats where possible. This step is critical, as it allows for consistent comparison across models and avoids inaccuracies that may arise from answer format mismatches. The Numbers from non-numeric or free-text answers are extracted to isolate the final numerical solution . Next, our custom-built taxonomy scorer is used to compute semantic (S), procedural (P), and cognitive (C) complexity scores for each problem. These are normalised to a [1, 5] scale and used to derive an overall composite score. Based on this score, problems are categorised into low, medium, or high difficulty tiers. The taxonomy not only helps structure the experimental groups but also provides granular insights into the complexity dimensions most correlated with success or failure, which will be critical during error analysis.

After scoring and binning, we apply both standard and adaptive prompting strategies. For the adaptive strategy, we use the taxonomy-informed algorithm described in Section b, which dynamically selects one of several CoT prompting strategies based on each problem's (S, P, C) profile. For baseline comparison, we also run the Standard CoT prompt across all problems. We evaluate prompts using three lightweight open-source LLMs: DeepSeek-R1-Distill-Qwen-1.5B, LLaMA 3.2–1B, and google/flan-t5-base. These models were chosen due to their good reasoning compatibility and their strong performance relative to model size. Testing across multiple architectures ensures our conclusions generalise beyond a single model type.

E. Evaluation Metrics

Our primary evaluation metric is accuracy, defined as the proportion of problems answered correctly. Accuracy is calculated for each model, dataset, and difficulty bin, allowing us to compare:

- Adaptive CoT vs. Standard CoT across all three difficulty levels
- Performance trends across the three LLMs
- Differences in how each dataset responds to adaptive prompting

This structured evaluation enables us to assess the central claim of our research: whether tailoring the reasoning style to the difficulty profile leads to improved performance.

Before we evaluate our accuracy, we will look at the distribution of semantic, procedural, and cognitive in our datasets. This will allow us to conduct qualitative error analysis. For our datasets, we will examine their semantic, procedural, and cognitive distribution to identify patterns to understand where a model might fall short. For example, poor results on high-cognitive tasks could suggest limitations in the model's abstract reasoning capabilities. Through this, we aim to uncover which dimensions most impact model success and whether adaptive CoT prompts mitigate these weaknesses.

IV. Results

A. Dataset Implementation and Complexity Analysis

As part of our evaluation, we will be looking into the distribution of semantic, procedural, and cognitive scores for qualitative analysis on the accuracy of the models.

The complexity analysis revealed meaningful differentiation across difficulty levels, as shown in the box-and-whisker plots. For GSM8K in fig1 , the semantic complexity exhibited a clear stratification between the low and medium categories, with median values increasing progressively from approximately 2.7 to 2.9. Procedural complexity demonstrated a similar pattern, albeit with greater variance within the medium difficulty category. Cognitive complexity also displayed some distinction, as evidenced by a noticeable shift in the plot. Regarding overall difficulty, the median value for the low category is approximately 2.1, while for the medium category it is around 2.7. This suggests that the taxonomy has successfully implemented a hierarchical system of difficulty.

SVAMP from fig 2 exhibited different complexity patterns, with semantic scores showing greater variance. Compared to GSM8K, the average semantic complexity is similar, suggesting that SVAMP may have comparable readability or lexical diversity. The procedural complexity distribution revealed some interesting characteristics, with medium-difficulty problems displaying a more compressed range than those in the low and high categories. Overall, the procedural complexity for SVAMP is lower than that of GSM8K, indicating that SVAMP's problems may involve fewer procedural steps. Cognitive complexity for low-difficulty problems appears similar to GSM8K, although GSM8K has slightly higher average scores, which could suggest that its problems require a deeper cognitive understanding. Overall, the average low-complexity score in GSM8K is higher, indicating that its easier problems are more challenging than those in SVAMP. For medium complexity, the two datasets are similar, although GSM8K shows greater variability.

ASDiv from fig3 has very similar semantic scores to GSM8K and SVAMP for low-complexity problems; however, for medium complexity, the semantic score for ASDiv is slightly higher than that of the other two datasets. The procedural score for low-complexity problems shows high variability compared to the other datasets, indicating that ASDiv exhibits less uniform difficulty characteristics at this level. Ironically, the medium-complexity category is much more consistent. Both ASDiv and SVAMP have lower procedural scores for medium-difficulty problems, suggesting that GSM8K involves more calculations and steps. In terms of cognitive complexity, ASDiv appears to have the lowest scores among the three datasets, indicating that less abstract or cognitive understanding is required to solve its problems. Overall, GSM8K has the highest complexity scores for both low and medium levels. ASDiv ranks higher than SVAMP in terms of low-complexity scores and is similar to SVAMP for medium complexity, although ASDiv shows high variability in the low-complexity category.

One main issue we encountered was that no problems in

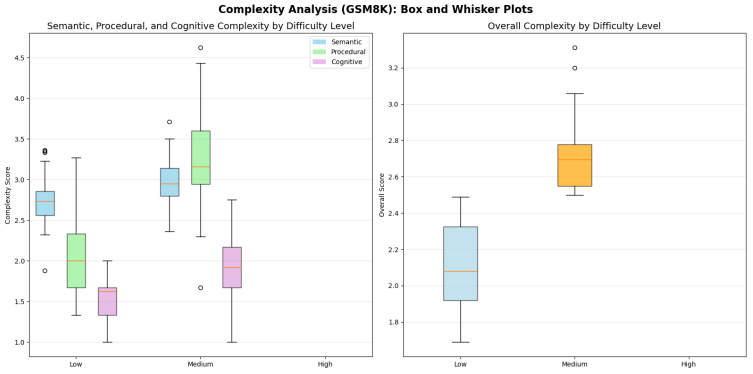

Fig. 1. GSM8K Dataset

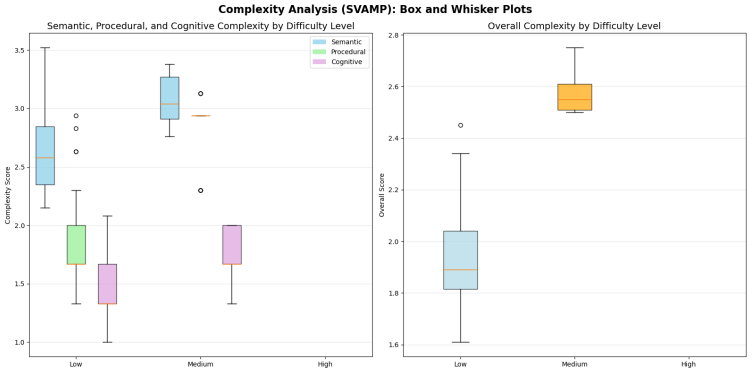

Fig. 2. SVAMP Dataset

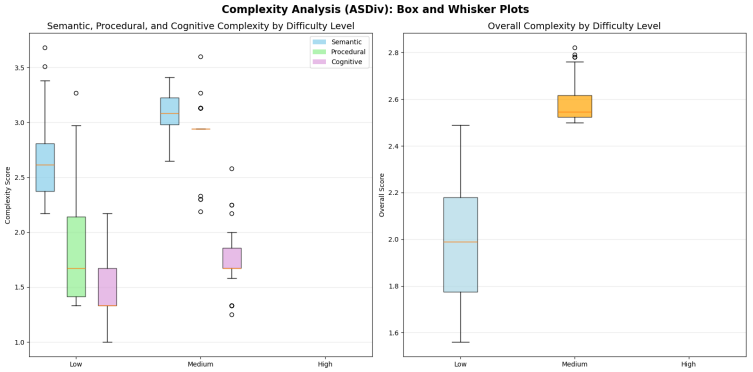

Fig. 3. ASDiv Dataset

any dataset were classified as high complexity; this is a small setback that we didn't account for, as we expected to see datasets identified into 3 bins. This could be due to treating semantic, procedural and cognitive equally. Hence, this will mean that we won't see any performance metrics for high difficulty categories.

B. Accuracy analysis

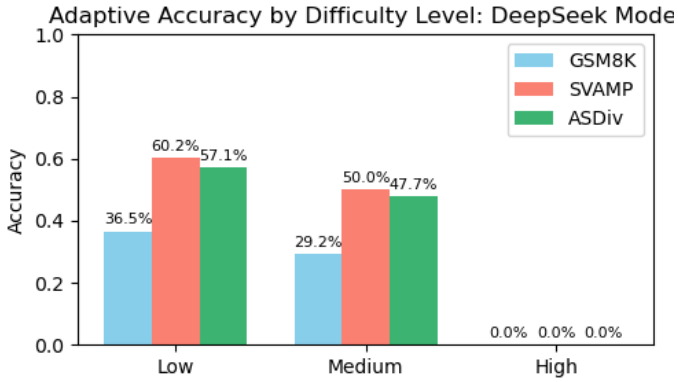

Fig. 4. Adaptive deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B summary

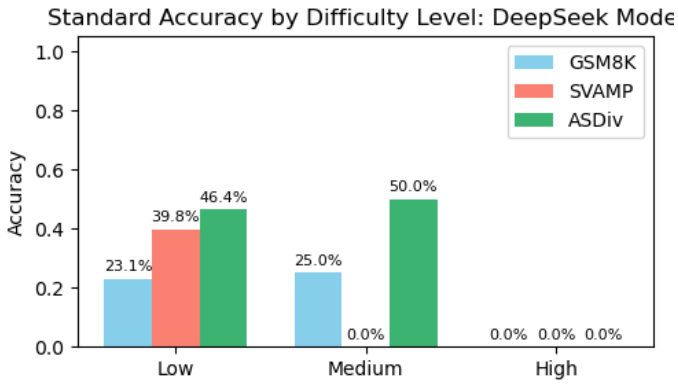

Fig. 5. Standard deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B summary

For the adaptive approach, the DeepSeek model demonstrated the most promising results within our adaptive CoT framework. For low-complexity problems, adaptive prompting led to substantial improvements, the accuracy on GSM8K increased from 23.1% to 36.5%, SVAMP from 39.8% to 60.2%, and ASDiv from 46.4% to 57.1%. These results strongly support our research question by showing that adaptive prompting, guided by a multi-dimensional difficulty taxonomy, can outperform standard CoT prompting, particularly for simpler problems. The improvements suggest that using minimal or streamlined CoT strategies tailored to problem complexity is more effective than applying a uniform reasoning strategy regardless of difficulty.

Furthermore, as shown in our earlier dataset analysis, GSM8K recorded the lowest accuracy gains in the adaptive setting. This may be attributed to its higher overall complexity scores, especially in the procedural dimension, which may carry greater influence in determining problem difficulty compared to semantic or cognitive complexity. This highlights the importance of weighting the dimensions appropriately within the taxonomy. For medium-complexity problems, the results were more varied. GSM8K showed a modest improvement (25.0% to 29.2%), SVAMP demonstrated a dramatic gain (0% to 50.0%), while ASDiv experienced a slight decrease (50.0% to 47.7%). The substantial improvement in SVAMP suggests that our adaptive decomposition strategy and standard CoT selector effectively address procedural and cognitive challenges in that dataset. In contrast, the reduced performance on ASDiv may be due to its higher semantic complexity, implying that for some problems, a standard CoT approach may already be well-suited.

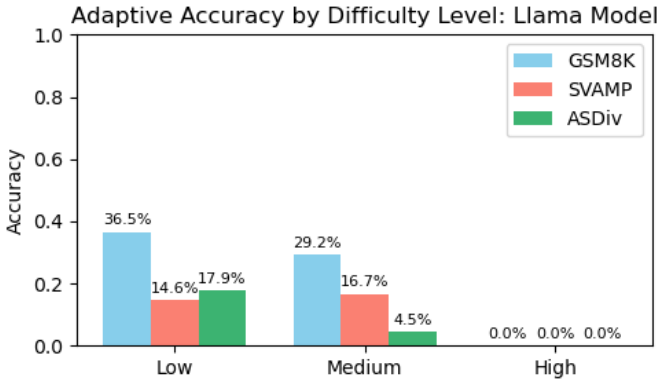

Fig. 6. Adaptive meta-llama/Llama-3.2-1B summary

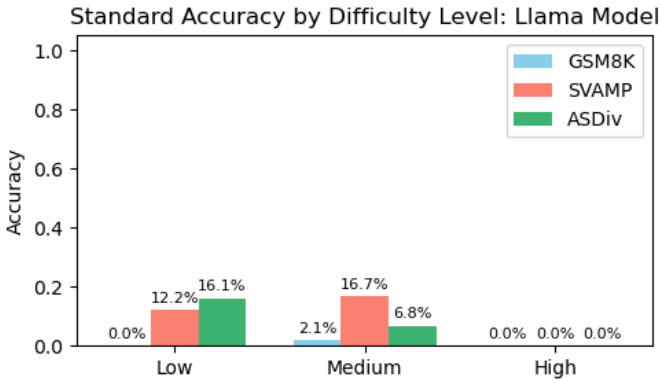

Fig. 7. Standard meta-llama/Llama-3.2-1B summary

The LLaMA 3.2 1B model exhibited different performance characteristics, offering further insights into the effectiveness of our adaptive CoT framework. This model performed less effectively than DeepSeek, which may be due to its smaller parameter size. Nevertheless, for low complexity problems, adaptive prompting continued to outperform standard CoT. GSM8K saw a notable improvement from 0% to 36.5%, while

ASDiv showed a modest gain from 16.1% to 179%. SVAMP, however, experienced a slight decrease from 16.1% to 14.6% . As with the DeepSeek model, LLaMA demonstrated significant improvement for GSM8K at the low complexity level, suggesting that the model benefitted from the minimal CoT prompts tailored to the higher procedural demands present in this dataset. For SVAMP and ASDiv, where semantic complexity is generally higher, the limited gains may indicate that LLaMA is less responsive to adaptive strategies when cognitive and linguistic complexity dominates. The results for medium complexity problems were particularly revealing. GSM8K showed a significant increase in accuracy from 2.1% to 29.2% , whereas SVAMP showed no change at 16.7% , and ASDiv experienced a decline from 6.8% to 4.5%. This mixed performance suggests that LLaMA's architecture interacts differently with adaptive prompting, especially at higher levels of complexity. The continued decline in ASDiv's performance may imply that for this dataset, which is more semantically oriented, standard CoT prompting may be more appropriate than a decompositional approach. Similarly, the unchanged result for SVAMP suggests that procedural complexity may be a more influential factor for LLaMA, and adaptive strategies may only yield improvements when the model is able to exploit procedural cues effectively.

These findings support our research question by highlighting that the effectiveness of adaptive CoT prompting varies not only by problem complexity but also by model architecture. While adaptive strategies clearly enhance performance in specific contexts, especially for low complexity problems, they must be tailored to the underlying strengths and limitations of the model in use.

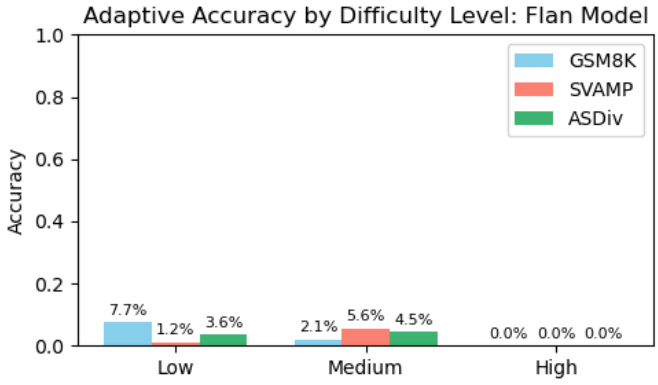

Fig. 8. Adaptive google/flan-t5-base summary

The google/flan-t5-base model showed the most limited overall performance, with generally low accuracy scores across both adaptive and standard approaches. However, despite its constrained reasoning capabilities, the model still demonstrated meaningful improvements with adaptive prompting for low-complexity problems. GSM8K showed a notable doubling of accuracy from 3.8% to 7.7%, representing a 103% relative improvement. While SVAMP maintained its performance level

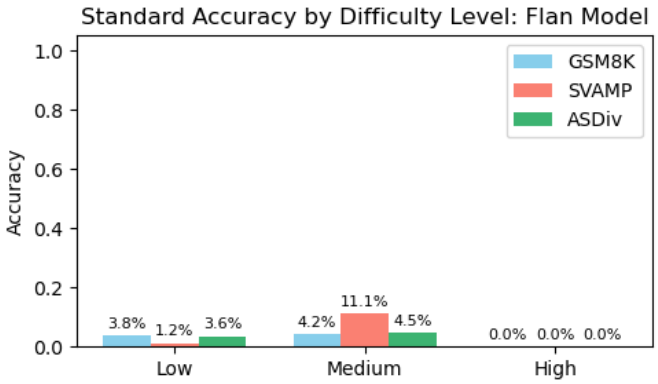

Fig. 9. Standard google/flan-t5-base summary

at 1.2% for both adaptive and standard approaches, this consistency suggests that the adaptive framework at least avoided degrading performance on this dataset. ASDiv maintained stable performance at 3.6% across both conditions. The limited absolute performance levels reflect the model's inherent capacity constraints, but the relative improvements, particularly the doubling of GSM8K performance, indicate that even smaller models can perform for low complexity problems.

For medium-complexity problems, the google/flan-t5-base model showed modest but interpretable patterns. GSM8K decreased from 4.2% to 2.1%, while SVAMP decreased from 11.1% to 5.6%, and ASDiv decreased from 4.5% to 4.5%. These mixed results suggest that the model's limited parameters create inconsistent responses. Therefore, from this change in model parameters, it suggests that our taxonomy is pointless and the poor performance in higher complexity does not highlight the strengths and weaknesses of the datasets. This is a small setback, however, it does show that small parameter models cannot be effective in arithmetic questions despite a taxonomy or adaptive framework like we saw with DeepSeek.

V. Limitations

One limitation of this study was the small dataset size, with only 100 data points per dataset. This restricts the generalisability of our findings and may have contributed to the observed variability when assessing semantic, procedural, and cognitive complexity. While our hypothesis regarding the presence of a hierarchical structure in problem difficulty was supported, a key limitation emerged from the fact that no problems were classified as high complexity. This was evident in the absence of plots for this category, limiting our evaluation to only low and medium difficulty problems. As a result, the full effectiveness of adaptive chain of thought prompting could not be assessed across the complete difficulty spectrum. This absence also points to a possible shortcoming in the current taxonomy. Specifically, cognitive complexity scores were consistently low across all three datasets, which may have disproportionately lowered the overall difficulty scores and prevented any problems from being classified as high

complexity. To address this issue, the taxonomy may require refinement, possibly through assigning greater weight to semantic, procedural, and particularly cognitive components, in order to better identify more challenging problems. Another limitation was the variability observed within individual difficulty categories, especially among low-complexity problems. This suggests that the classification method may lack robustness. Reducing this variability would allow for clearer stratification between difficulty levels and would support a more accurate evaluation of how adaptive prompting performs in comparison to standard prompting across genuinely distinct levels of mathematical reasoning complexity. Finally, the dependency on model architecture posed a further limitation. While the deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B produced encouraging results, the google flan t5 base model showed consistently poor performance under both adaptive and standard prompting. This suggests that our framework may be effective primarily with models above a certain parameter size. The mixed outcomes seen with the LLaMA 3.2 1B model further support this concern, indicating that adaptive strategies do not uniformly enhance performance across all model types.

VI. Future work

The most immediate improvement would be to increase the number and diversity of data points in each dataset. Using larger samples across a wider range of problem types will reduce the risk of overfitting, improve statistical validity, and provide more reliable complexity stratification. It may also increase the likelihood of identifying truly high-complexity problems, which are essential for evaluating the full range of the adaptive strategy. The current taxonomy may require recalibration, especially in how it balances semantic, procedural, and cognitive dimensions. One possible refinement is to assign weights to each component based on the nature of the dataset or target reasoning skill. In particular, from observation, reducing the influence of cognitive complexity could help focus more on the arithmetic steps and the actual linguistics of the problem, facilitating better classification across all three difficulty tiers. This should also solve the issue of the lack of high-complexity examples. Future studies should include more detailed qualitative assessments of the generated CoT responses. Analysing the nature of reasoning, error patterns, or step-by-step coherence can reveal where and why adaptive strategies succeed or fail, beyond what accuracy scores can convey. Lastly, although this study focused on mathematical reasoning, the adaptive prompting approach may also be valuable in other structured domains, such as physics problems, logical puzzles, or reading comprehension tasks. Evaluating the generalisability of the taxonomy across domains could extend its utility.

VII. Conclusion

Our research investigated whether adaptive Chain-of-Thought (CoT) prompting, guided by a multi-dimensional difficulty taxonomy, could outperform standard CoT approaches in mathematical reasoning tasks. We developed a three-dimensional framework classifying problems by semantic, procedural, and cognitive complexity, then applied adaptive CoT strategies based on difficulty bins.Our findings demonstrate that adaptive CoT prompting can outperform standard approaches, particularly for low-complexity problems. The deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B showed substantial improvements. These results support our hypothesis that tailoring reasoning strategies to problem complexity yields meaningful performance gains. However, significant limitations emerged as no problems were classified as high-complexity, restricting evaluation to low and medium difficulty categories. Furthermore, model architecture proved critical, while deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B benefited consistently, google/flan-t5-base showed limited improvements and LLaMA 3.2 1B had mixed results. This suggests adaptive CoT strategies work best with models above a certain parameter threshold. Despite these limitations, our research demonstrates that adaptive prompting strategies can meaningfully improve mathematical reasoning performance when matched to appropriate model architectures. The multi-dimensional taxonomy provides a systematic framework extending beyond previous step-counting approaches, while revealing that different datasets respond differently to adaptive strategies. This work establishes a foundation for more sophisticated difficulty-aware reasoning approaches, with future research needed to expand dataset sizes, refine taxonomy weights, and evaluate generalisability across other structured reasoning domains.

References

[1] J. Sun et al., "A survey of reasoning with foundation models," *arXiv preprint arXiv:2312.11562*, 2023.

[2] J. Wei et al., "Chain of thought prompting elicits reasoning in large language models," in *Proc. NeurIPS*, 2022, pp. 24824–24837.

[3] D. Zhou et al., "Least-to-most prompting enables complex reasoning in large language models," *arXiv preprint arXiv:2205.10625*, 2022.

[4] T. Kojima, S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," in *Proc. NeurIPS*, 2022, pp. 22199–22213.

[5] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2022.

[6] Y. Fu, W. Y. Wang, and H. Cai, "Complexity-based prompting for multi-step reasoning," in *Proc. ICLR*, 2022.

[7] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic chain of thought prompting in large language models," *arXiv preprint arXiv:2210.03493*, 2022.

[8] X. Zheng et al., "Critic-CoT: Boosting the reasoning abilities of large language model via chain-of thoughts critic," *arXiv preprint arXiv:2408.16326*, 2024.

[9] L. Benedetto et al., "A survey on recent approaches to question difficulty estimation from text," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–37, 2022.

[10] N. Pongsakdi, A. Kajamies, K. Veermans, K. Lertola, M. Vauras, and E. Lehtinen, "What makes mathematical word problem solving challenging? exploring the roles of word problem characteristics, text comprehension, and arithmetic skills," *ZDM*, vol. 52, no. 1, pp. 33–44, 2020.

[11] S. Vajjala and D. Meurers, "On improving the accuracy of readability classification using insights from second language acquisition," in *Proc. 7th Workshop on Building Educational Applications Using NLP*, 2012, pp. 163–173.

[12] S. A. Crossley, M. S. Greenfield, and D. S. McNamara, "Assessing text readability using cognitively based indices," *TESOL Quarterly*, vol. 42, no. 3, pp. 475–493, 2008.