Doctoral Dissertation

Prompt-Guided Navigation in the Reasoning Space of Large Language Models

(大規模言語モデルの推論空間におけるプロンプト誘導型ナビゲーション)

September, 2026

Zhu Mengyao

Graduate School of Sciences and Technology for Innovation,
Yamaguchi University

Abstract

Mathematical definition generation represents a foundational yet cognitively demanding task in college STEM education, requiring precise formal language, conceptual completeness, and adherence to disciplinary conventions. As large language models become increasingly integrated into educational and technical applications, understanding how different prompt strategies influence the quality of generated mathematical definitions becomes essential. This dissertation presents three sequential studies that systematically investigate the effects of reasoning structure and persona specification on mathematical definition generation. The first study introduces the Cumulative Thinking Prompt, a linear reasoning method $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ that instructs the language model to perform five sequential knowledge recall steps from essential to abstract understanding before generating the final definition. Evaluated across 439 mathematical terms spanning 18 subjects, this method achieved higher performance compared to conventional baselines, suggesting that structured knowledge accumulation prior to generation is associated with improved definition quality

The second study proposes the AUDIT prompt, an iterative reasoning framework $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\to} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\to} \iota_3)^{\delta_2}\}$ that incorporates recursive quality audits and refinement cycles. Across an expanded dataset of 921 mathematical terms, AUDIT V3 achieved a holistic relative performance of 114.42% against the baseline, outperforming the Cumulative Thinking Prompt by 11.61 percentage points while demonstrating exceptional stability with a standard deviation of 0.058. This establishes that iterative reasoning with explicit feedback mechanisms yields substantially higher and more consistent quality than linear reasoning approaches.

The third study examines the role of persona specificity through the Professor-Rank-Aware Prompt Framework, proposing eight fine-grained professor-rank specifications ranging from a distinguished professor to a lecturer. The results reveal a clear performance hierarchy aligned with academic seniority, with distinguished professor achieving 106.3% relative performance and lecturer achieving 97.2%, demonstrating that precise persona specifications serve as effective navigation signals within the reasoning space. Critically, the coarse-grained "excellent professor" baseline was outperformed by four higher-ranked variations (namely, distinguished professor, emeritus professor, full professor, and research

professor), indicating that the coarse-grained "excellent professor" specification yielded lower relative performance than four fine-grained alternatives

Together, these studies establish that both the structure of reasoning (linear versus iterative) and the precision of context specifications fundamentally shape language model navigation of the reasoning space $\mathfrak{R}$, with iterative reasoning and fine-grained persona settings yielding superior, more stable performance. The findings contribute evidence-based methodologies for prompt design in knowledge-intensive domains, demonstrating that deliberate reasoning preparation, recursive refinement, and contextually optimized persona specifications offer practical, training-free approaches to enhancing language model outputs for educational and technical applications.

Acknowledgments

First and foremost, I wish to express my deepest gratitude to Nguyen Sensei for his unwavering support, patience, and tolerance for my continuous learning explorations and failures throughout my journey from master's to PhD studies. For a PhD student, finding the optimal knowledge subspaces $\mathcal{R}_{\text{opt}} \subseteq \mathfrak{R}$ and directions $\rho_{\text{opt}} \in \mathcal{R}_{\text{opt}}$ within the whole learning landscape is a profoundly challenging endeavor, one that requires intellectual rigor and resilience in the face of repeated setbacks. His consistent kindness and my continuous efforts have enabled me to transform moments of difficulties into opportunities for growth and discovery.

Algorithm 1 Pseudocode: PhD Learning Journey

Require: Nguyen Sensei's unwavering support $S_{\text{Nguyen}} = \{s_1, s_2, \ldots, s_m\}$ where each s_k represents patience, tolerance for continuous failures, and guidance, A genuine curiosity to learn about neural networks $\mathcal{N}_{\text{net}}$ and mathematics $\mathcal{M}_{\text{math}}$, A learning pipeline that is both linear $\mathcal{L}_{\text{learn}} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ and iterative $\mathcal{I}_{\text{learn}} = \{\iota_1(\xrightarrow{\circlearrowleft} \iota_2)^{\delta_1}(\xrightarrow{\circlearrowleft} \iota_3)^{\delta_2} \cdots (\xrightarrow{\circlearrowleft} \iota_T)^{\delta_{T-1}}\}$, A resilient mindset with never-give-up spirit $\mathcal{R}_{\text{mind}}$, Flexible behaviors $\mathcal{B}_{\text{flex}} = \{b_1, b_2, \ldots, b_K\}$ to overcome difficulties and seize opportunities, The ability to find and navigate optimal knowledge subspaces $\mathcal{R}_{\text{opt}} \subseteq \mathfrak{R}$ and directions $\rho_{\text{opt}} \in \mathcal{R}_{\text{opt}}$ within the holistic knowledge space $\mathfrak{R}$

Ensure: Completion of PhD journey with accumulated knowledge and insights $\mathcal{K}_{\text{PhD}}$

Initialize knowledge state with neural networks $\mathcal{N}_{\text{net}}$, mathematics $\mathcal{M}_{\text{math}}$, language models $\mathcal{L}_{\text{LM}}$, academic research $\mathcal{A}_{\text{res}}$, reading $\mathcal{R}_{\text{read}}$, and writing $\mathcal{W}_{\text{write}}$ fundamentals
Initialize mindset $\mathcal{M}_{\text{set}}$ with curiosity, resilience, and adaptability
for each phase $\phi_p \in \Phi$ of the PhD journey (progressing linearly through stages) **do**
    Receive guidance and support from Nguyen Sensei S_{Nguyen}
    for each challenge $\chi_c \in \mathcal{C}_{\text{challenge}}$ encountered during the current phase ϕ_p **do**
        Engage in iterative learning exploration $\mathcal{I}_{\text{learn}}$:
            Believe in the possibility of understanding
            Accept existing knowledge without overthinking
            Learn new concepts and procedures with repetition
            Refine and improve understanding in different scenarios
            Innovate within the constraints of the problem domain
        if falling into knowledge trap $\tau_k \in \mathcal{T}_{\text{trap}}$ **then**
            Seek Nguyen Sensei's understanding and acceptance S_{Nguyen}
            Identify escape strategies $\mathcal{E}_{\text{escape}} = \{e_1, e_2, \ldots, e_E\}$
            Navigate toward better knowledge directions $\rho_{\text{new}} \in \mathcal{R}_{\text{opt}}$
            Continue exploration with refined approach
        end if
    end for
    Consolidate learning and advance to next phase ϕ_{p+1}
    Maintain consistent hard work and never-give-up spirit $\mathcal{R}_{\text{mind}}$
end for
Synthesize accumulated knowledge $\mathcal{K}_{\text{PhD}}$ into dissertation
return PhD completion with gratitude for the journey

I am also deeply indebted to the foundational milestones in natural language processing, particularly Word2Vec and BERT, which served as critical compass points in my navigation of the complex learning spaces of neural networks $\mathcal{N}_{\text{net}}$, mathematics $\mathcal{M}_{\text{math}}$, and large language models $\mathcal{L}_{\text{LM}}$. These seminal contributions not only provided the theoretical scaffolding necessary to understand the mechanics of informed generation but also opened an entirely new intellectual horizon. They paved the way for my exploration into why and how language models possess such remarkable capabilities, inspiring the curiosity that drove this research into prompt engineering and reasoning navigation.

The essential learning experience that shaped this journey can be understood as both a linear $\mathcal{L}_{\text{learn}} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ and iterative $\mathcal{I}_{\text{learn}} = \{\iota_1(\overset{\circlearrowleft}{\to} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\to} \iota_3)^{\delta_2} \cdots (\overset{\circlearrowleft}{\to} \iota_T)^{\delta_{T-1}}\}$ procedure, a process of believing, accepting, learning, experimenting, refining, and innovating with constraints in mind. This long journey has been filled with great challenges, both physical and mental. Falling into knowledge traps $\tau_k \in \mathcal{T}_{\text{trap}}$ has been the norm, but finding escape strategies $\mathcal{E}_{\text{escape}} = \{e_1, e_2, \ldots, e_E\}$ and locating better directions $\rho_{\text{new}} \in \mathcal{R}_{\text{opt}}$ with more knowledge has been genuinely enjoyable.

It's time to start a new chapter in life with family, to be awesome.

Table of Contents

Abstract i
Acknowledgments iii
List of Tables vii
List of Figures viii
Notations ix
Chapter 1 Introduction 1
1.1 Research Motivation 1
1.2 Research Objectives 2
1.3 Dissertation Outline 3
Chapter 2 Cumulative Thinking Prompt with Linear Reasoning: Sequential Reflections Before Mathematical Definition Generation 4
2.1 Introduction 4
2.2 Related Works 5
2.2.1 Linear-Reasoning Prompt Applications 7
2.2.2 Linear-Reasoning Prompt Construction 15
2.3 Methodology 22
2.3.1 Problem Definition 22
2.3.2 Dataset 23
2.3.3 Methods 25
2.3.4 Experiment Setup 28
2.3.5 Evaluation Metrics 28
2.4 Results and Analytics 30
2.4.1 Absolute Performance Analytics 31
2.4.2 Relative Performance Analytics 32
2.4.3 Automatic Evaluation Reliability Analytics 34
2.4.4 Concrete Examples 35
2.5 Conclusion 36
2.5.1 Research Contributions 36
2.5.2 Future Directions 38
Chapter 3 AUDIT Prompt with Iterative Reasoning: On-the-Fly Mathematical Definition Generation with Cyclic Quality Refinement 40
3.1 Introduction 40
3.2 Related Works 41
3.2.1 Iterative-Reasoning Prompt Applications 42
3.2.2 Iterative-Reasoning Prompt Construction 48
3.3 Methodology 49
3.3.1 Problem Definition 49
3.3.2 Dataset 50
3.3.3 Methods 51
3.3.4 Experiment Setup 53
3.3.5 Evaluation Metrics 53
3.4 Results and Analytics 54
3.4.1 Holistic Performance Analytics 54
3.4.2 Subject-Level Performance Analytics 56

3.4.3 AUDIT Prompt vs. Cumulative Thinking Prompt . . . 59
3.4.4 Concrete Examples . . . 62
3.5 Conclusion . . . 64
3.5.1 Research Contributions . . . 64
3.5.2 Future Directions . . . 65

Chapter 4 Reasoning Assumptions: What We Expect vs. What We Trust in Persona-Aware Prompt Engineering . . . 68
4.1 Introduction . . . 68
4.2 Related Works . . . 69
4.2.1 Persona-Aware Prompt with Role Settings . . . 70
4.2.2 Persona-Aware Prompt with Competency Settings . . . 74
4.3 Persona-Aware Reasoning Assumptions . . . 76
4.4 Case Study: Persona-Aware Prompt Does not Always Make Sense . . . 86
4.4.1 Problem Definition . . . 86
4.4.2 Dataset . . . 86
4.4.3 Methods . . . 87
4.4.4 Experiment Setup . . . 89
4.4.5 Evaluation Metrics . . . 89
4.5 Results and Analytics . . . 90
4.5.1 Holistic Performance Analytics . . . 90
4.5.2 Subject-Level Absolute Performance Analytics . . . 92
4.5.3 Subject-Level Relative Performance Analytics . . . 97
4.5.4 Performance Stability Analytics . . . 99
4.5.5 Concrete Examples . . . 101
4.6 Conclusion . . . 104
4.6.1 Research Contributions . . . 104
4.6.2 Future Directions . . . 105

Chapter 5 Holistic Conclusion and Discussions . . . 108
5.1 What we Have Achieved . . . 108
5.2 What we Should Improve . . . 109

Chapter 6 Holistic Future Research Directions . . . 111
6.1 What We Can Expect . . . 111

References . . . 114

Appendix A Appendix . . . 126
A.1 Statistical Verification Results . . . 126
A.1.1 Pairwise Comparison Between Cumulative Thinking Prompt and Baselines . . . 126
A.1.2 Pairwise Comparison between AUDIT Prompt and Baselines . . . 126
A.1.3 Pairwise Comparison between Professor-Rank-Aware Prompt and Baseline . . . 127
A.2 Generation Demonstrations for Three Proposed Prompt Methods . . . 127
A.2.1 Definition Demonstrations for Study 1: Cumulative Thinking Prompt . . . 127
A.2.2 Definition Demonstrations for Study 2: AUDIT Prompt . . . 130
A.2.3 Definition Demonstrations for Study 3: Professor-Rank-Aware Prompt . . . 133
A.3 Prompt Construction and Experiment Reproduction . . . 138
A.3.1 Considerations for Prompt Engineering . . . 138
A.3.2 Evolving LLM Capabilities . . . 138
A.3.3 Probabilistic Nature of LLM Generation . . . 139
A.3.4 LLM-Specific Behaviors . . . 139

List of Tables

1 Notation: Essential Expressions ix
2 Notation: Language Models and Agents x
3 Notation: Language Model Reasoning xi
4 Notation: Prompt Methods xii

2.1 Cumulative Thinking Prompt: Linear Reasoning-Aware Prompt Citations 7
2.2 Cumulative Thinking Prompt Dataset: Dataset 24
2.3 Large Language Models for All Three Studies 28
2.4 Cumulative Thinking Prompt: Hyperparameters 28
2.5 Cumulative Thinking Prompt: Evaluation Framework 30
2.6 Cumulative Thinking Prompt: Holistic Absolute Performance (All Variations) 32
2.7 Cumulative Thinking Prompt: Relative Absolute Performance (All Variations) 34

3.1 AUDIT Prompt: Citations of Iterative Reasoning and Constructions 42
3.2 AUDIT Prompt Dataset: Extended Dataset 50
3.3 AUDIT Prompt: Holistic Relative Performance (All Variations) 55
3.4 AUDIT Prompt: Holistic Absolute Performance (All Variations) 56
3.5 AUDIT Prompt: Subject-Level Absolute Performance (All Variations) 57
3.6 AUDIT Prompt: Subject-Level Performance (All Vairations) 58
3.7 AUDIT Prompt: Subject-Level Performance (Selected Variations) 60
3.8 AUDIT Prompt: Subject-Level Relative Performance (Selected Variations) 62

4.1 Persona-Rank-Aware Prompt: Citations of Persona-Aware Prompt Variations 70
4.2 Persona-Rank-Aware Prompt: Citations of Persona-Aware Prompt Reasoning Assumptions 78
4.3 Professor-Rank-Aware Prompt: Holistic Relative Performance (All Variations) 91
4.4 Professor-Rank-Aware Prompt: Absolute Performance Difference (Proposed vs. Baseline) 92
4.5 Professor-Rank-Aware Prompt: Subject-Level Absolute Performance (All Variations) . . . 93
4.6 Professor-Rank-Aware Prompt: Subject-Level Absolute Performance (Selected Variations) 95
4.7 Professor-Rank-Aware Prompt: Subject-Level Relative Performance (Selected Variations) 99
4.8 Professor-Rank-Aware Prompt: Subject-Level Ranking Stability (All Variations) 101

List of Figures

2.1 Cumulative Thinking Prompt: Dataset Distribution by Subject 24
2.2 Cumulative Thinking Prompt: Holistic Relative and Absolute Performances 30
2.3 Cumulative Thinking Prompt: Subject-Level Absolute Performance (All Variations) . . . 31
2.4 Cumulative Thinking Prompt: Subject-Level Relative Performance (All Variations) 33

3.1 AUDIT Prompt: Dataset Distribution by Subject 51
3.2 AUDIT Prompt: Holistic Relative and Absolute Performances (All Variations) 54
3.3 AUDIT Prompt: Subject-Level Absolute Performance (All Variations) 56
3.4 AUDIT Prompt: Subject-Level Relative Performance (All Vairations) 58
3.5 AUDIT Prompt: Subject-Level Absolute Performance (Selected Variations) 60
3.6 AUDIT Prompt: Subject-Level Relative Performance (Selected Variations) 61

4.1 Professor-Rank-Aware Prompt: Dataset Distribution by Subject 87
4.2 Professor-Rank-Aware Prompt: Holistic Relative and Absolute Performance (All Variations) 90
4.3 Professor-Rank-Aware Prompt: Subject-Level Absolute Performance (All Variations) . . . 93
4.4 Professor-Rank-Aware Prompt: Subject-Level Absolute Performance (Selected Variations) 94
4.5 Professor-Rank-Aware Prompt: Subject-Level Performance (All Variations) 98
4.6 Professor-Rank-Aware Prompt: Subject-Level Performance (Selected Variations) 99

Notations

Notation for Essential Components

Table 1: Notation: Essential Expressions

Title	Notation	Convention
Math Subjects	$S = \{s_1, s_2, \ldots, s_m\}$	Set of m mathematical subjects, where each s_i $(i = 1, 2, \ldots, m)$ represents a specific mathematical subject
Math Terms	$T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$	Set of n_i mathematical terms for subject s_i, where each $t_{(i,j)}$ $(j = 1, 2, \ldots, n_i)$ represents a specific mathematical term within subject s_i
Training Dataset	$\mathcal{D}_{\text{train}} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$	Set of N training examples, where each (x_i, y_i) $(i = 1, 2, \ldots, N)$ represents an input-output pair used for model training. The dataset may be used for supervised learning, fine-tuning, or demonstration construction for few-shot prompt
Seed Content	$\mathcal{C}_{\text{seed}} = \{c_1, c_2, \ldots, c_N\}$	Set of N seed content items used to initialize, guide, or bootstrap generation and reasoning processes. Seed content may include initial questions, prompt instructions, demonstration examples, QA pairs, or any foundational artifacts that establish the starting point for subsequent operations such as prompt engineering, few-shot learning, iterative refinement, or multi-agent collaboration
Generated Definitions	$\mathcal{G} = \bigcup_{p \in \mathcal{P}} \mathcal{G}_p$ where $\mathcal{G}_p = \{g^{(p)}_{(i,j)} \mid s_i \in S, t_{(i,j)} \in T_{s_i}\}$	Union of definition sets produced across all prompt method variations. For each prompt method $p \in \mathcal{P}$, $\mathcal{G}_p$ contains definitions generated by $M_{\text{auto-gen}}$, where each $g^{(p)}_{i,j}$ corresponds to the definition for term $t_{(i,j)}$ in subject s_i using prompt method p
Evaluation Metrics	$\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$	Set of 11 evaluation metrics employed to assess the quality of generated definitions. Each e_k $(k = 1, 2, \ldots, 11)$ represents a distinct metric targeting specific aspects of definition quality
Evaluation Results	$\mathcal{R} = \{r_{(i,j,p,k)} \mid s_i \in S, t_{(i,j)} \in T_{s_i}, p \in \mathcal{P}, e_k \in \mathcal{E}\}$	Set of evaluation scores obtained from assessing generated definitions. Each $r_{(i,j,p,k)}$ represents the score for definition $g^{(p)}_{(i,j)}$ evaluated by metric e_k, obtained via either $M_{\text{auto-eval}}$ (automatic evaluation) or $M_{\text{man-eval}}$ (manual evaluation)

Notation for Language Models and Agents

Table 2: Notation: Language Models and Agents

Title	Notation	Convention
Language Models	$\mathcal{M} = \{M_{(\text{a-gen})}, M_{(\text{a-eval})}, M_{(\text{m-eval})}, \pi_\theta\}$	Set of language models used for generation, evaluation, and reinforcement learning tasks. $M_{(\text{a-gen})}$ denotes the model for automatic definition generation, $M_{(\text{a-eval})}$ denotes the model for automatic evaluation against metrics, $M_{(\text{m-eval})}$ denotes the model supporting human evaluators for manual evaluation, and π_θ denotes the policy model parameterized by θ used in reinforcement learning to optimize generation behaviors through interaction with the environment
Agents	$\mathcal{A} = \{A_1, A_2, \ldots, A_K\}$ where $A_k = (\mathcal{M}_k, \mathcal{C}_k, \mathcal{R}_k)$	Set of K language model instances operating as distinct agents, where each agent A_k is defined by its language model configuration $\mathcal{M}_k$ (which may be an instance of a base model with specific parameters), its assigned role or persona $\mathcal{C}_k$, and its behavioral constraints or optimization objectives $\mathcal{R}_k$. Multiple agents may share the same base model but differ in settings such as temperature, top-p, system prompts, task specialization, or learning strategies, enabling collaborative or competitive interactions in multi-agent frameworks
Collaboration Strategy	$\mathcal{S}_{\text{collab}} = \{s_1, s_2, \ldots, s_L\}$ where each $s_\ell : \mathcal{A} \times \mathcal{T} \to \mathcal{A}_{\text{coord}}$	Set of L collaboration strategies governing how multiple agents interact to accomplish shared objectives. Each strategy s_ℓ defines a coordination mechanism that maps the set of available agents $\mathcal{A}$ and the task $\mathcal{T}$ to a coordinated agent configuration $\mathcal{A}_{\text{coord}}$, specifying interaction patterns such as sequential consensus, parallel deliberation, hierarchical delegation, debate, or voting. These strategies determine communication protocols, information sharing pathways, conflict resolution methods, and the aggregation of individual agent outputs into collective outcomes

Notation for Language Model Reasoning

Table 3: Notation: Language Model Reasoning

Title	Notation	Convention
Linear Reasoning	$\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$	Set of R sequential reasoning steps connected by forward progression, where each ℓ_r $(r = 1, 2, \ldots, R)$ represents a discrete intermediate thought generated by the LLM in chronological order, with $\to$ symbolizing the unidirectional flow from one reasoning step to the next without feedback loops
Iterative Reasoning	$\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\to}\ \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\to}\ \iota_3)^{\delta_2} \cdots (\overset{\circlearrowleft}{\to}\ \iota_T)^{\delta_{T-1}}\}$ where $\delta_t \in \{0, 1\}$	Set of T reasoning steps where iterative refinement is conditionally applied, with each δ_t serving as an indicator function determined by evaluating previous reasoning against threshold metrics $\mathcal{E}_{\text{iter}} \subseteq \mathcal{E}$. Steps connected by $\overset{\circlearrowleft}{\to}$ undergo cyclic feedback and refinement when $\delta_t = 1$, examining previous outputs to identify improvement points and incorporate refinements; steps without the iterative symbol ($\delta_t = 0$) proceed as linear progression when quality metrics meet predefined thresholds
Reasoning Space	$\mathfrak{R} = \bigcup_{u=1}^{U} \mathcal{R}_u$ where each $\mathcal{R}_u = \{\rho_{(u,1)}, \rho_{(u,2)}, \cdots, \rho_{(u,W_u)}\}$	Reasoning space comprises the union of all reasoning subspaces. The reasoning space $\mathfrak{R}$ is structured as a collection of U reasoning subspaces $\mathcal{R}_u$, with each reasoning subspace containing W_u reasoning directions, collectively representing the complete conceptual landscape of possible reasoning paths
Reasoning Subspace	$\mathcal{R}_u = \{\rho_{(u,1)}, \rho_{(u,2)}, \ldots, \rho_{(u,W_u)}\} \subset \mathfrak{R}$	A reasoning subspace $\mathcal{R}_u$ is a set of W_u reasoning directions within a specific conceptual region of the reasoning space. Each reasoning subspace groups related reasoning paths that share common conceptual foundations or approaches, and may contain multiple optimal or near-optimal reasoning directions, contain a single acceptable reasoning direction, or contain no viable reasoning directions
Reasoning Direction	$\rho_{(u,w)} \in \mathcal{R}_u \subset \mathfrak{R}$	A specific reasoning direction $\rho_{(u,w)}$ is a reasoning path or trajectory within reasoning subspace $\mathcal{R}_u$ that the LLM can follow to generate an output. A reasoning direction may represent an optimal or near-optimal solution for a given task; multiple reasoning directions may exist within a single reasoning subspace, and some reasoning directions may lead to suboptimal or unacceptable outputs
Reasoning Navigation	$\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D}) = \mathfrak{R}_{\mathcal{I},\mathcal{D}} \to \mathcal{R}_{\mathcal{I},\mathcal{D}} \to \rho_{\mathcal{I},\mathcal{D}}$ where $\mathfrak{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$, $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}_{\mathcal{I},\mathcal{D}}$, $\rho_{\mathcal{I},\mathcal{D}} \subseteq \mathcal{R}_{\mathcal{I},\mathcal{D}}$	Sequential navigation process executed by the large language model $\mathcal{M}$ as instructed by prompt instructions $\mathcal{I}$ and optionally prompt demonstrations $\mathcal{D}$. The model first identifies feasible reasoning subspaces $\mathfrak{R}_{\mathcal{I},\mathcal{D}}$ within the reasoning space $\mathfrak{R}$, then selects relevant subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}}$ from them, and finally follows reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ within those subspaces to generate solutions. The arrow notation $\to$ indicates progressive refinement from broader exploration to focused selection through nested subsets, where multiple subspaces and directions may be explored at each stage

Notation for Prompt Methods

Table 4: Notation: Prompt Methods

Title	Notation	Convention
Candidate Prompts	$\mathcal{P}_{\text{cand}} \subseteq \mathcal{P}$	Set of candidate prompt methods selected for experimental evaluation, where $\mathcal{P}_{\text{cand}} = \{p_{(c,1)}, p_{(c,2)}, \ldots, p_{(c,C)}\}$ with each $p_{(c,c')}$ $(c' = 1, 2, \ldots, C)$ representing a specific prompt method under consideration
Prompt Instructions	$\mathcal{I} = \{i_1, i_2, \ldots, i_L\}$	Set of L prompt instructions that guide the LLM's generation process. Each i_ℓ $(\ell = 1, 2, \ldots, L)$ represents a specific instruction component (e.g., task description, output format requirements, reasoning constraints) that collectively direct the model's behavior
Prompt Demonstrations	$\mathcal{D} = \{d_1, d_2, \ldots, d_K\}$	Set of K prompt demonstrations serving as examples for few-shot prompt. Each d_k $(k = 1, 2, \ldots, K)$ represents a specific demonstration, with K being a variable number adjustable across experimental configurations
Prompt Methods	$\mathcal{P} = \mathcal{P}_{\text{prop}} \cup \mathcal{P}_{\text{base}}$	Set of prompt method variations, where $\mathcal{P}_{\text{prop}} = \{p_{(p,1)}, p_{(p,2)}, \ldots, p_{(p,a)}\}$ is the set of a proposed prompt methods with each $p_{(p,i)}$ $(i = 1, 2, \ldots, a)$ representing a specific proposed method, and $\mathcal{P}_{\text{base}} = \{p_{(b,1)}, p_{(b,2)}, \ldots, p_{(b,b)}\}$ is the set of b baseline prompt methods with each $p_{(b,j)}$ $(j = 1, 2, \ldots, b)$ representing a specific baseline method
Professor-Rank-Aware Prompt	$\mathcal{P}_{\text{profs}} \subseteq \mathcal{P}$	Set of n Professor-Rank-Aware Prompt method variations as $\{p_{(\text{profs},1)}, p_{(\text{profs},2)}, \ldots, p_{(\text{profs},n)}\}$, where each $p_{(\text{profs},k)}$ $(k = 1, 2, \ldots, n)$ represents a distinct prompt method with a specific professor rank as context for generating definitions of term $t_{(i,j)}$ in subject s_i. Each method follows the output structure $\mathcal{O}_{(\text{prof},k)} = \{d\}$ where d is the final definition with LaTeX mathematical notation
Direct Prompt	$p_{\text{direct}} \in \mathcal{P}$	Baseline prompt method where the LLM directly generates the definition of term $t_{(i,j)}$ in subject s_i without intermediate reasoning steps or persona adoption. The output follows the structure $\mathcal{O}_{\text{direct}} = \{d\}$ where d is the final definition with LaTeX mathematical notation
Step-by-Step Prompt	$p_{\text{sbs}} \in \mathcal{P}$	Prompt method where the LLM engages in sequential reasoning steps to progressively build toward the solution before generating the final definition of term $t_{(i,j)}$ in subject s_i. The output follows the structure $\mathcal{O}_{\text{sbs}} = \{s_1, s_2, \ldots, s_n, d\}$ where each s_k represents an intermediate reasoning step and d is the final definition with LaTeX mathematical notation
Excellent-Professor-Aware Prompt	$p_{\text{e-pro}} \in \mathcal{P}$	Prompt method where the LLM adopts the persona of an excellent math professor to generate the definition of term $t_{(i,j)}$ in subject s_i. The output follows the structure $\mathcal{O}_{\text{e-pro}} = \{d\}$ where d is the final definition with LaTeX mathematical notation
Cumulative Thinking Prompt	$p_{\text{ct}} \in \mathcal{P}$	Prompt method where the LLM performs five sequential recalls from essential to abstract understanding of term $t_{(i,j)}$ in subject s_i before generating the definition. The output follows the structure $\mathcal{O}_{\text{ct}} = \{r_1, r_2, r_3, r_4, r_5, d\}$ where r_k represents the k-th sequential thinking and d is the final definition with LaTeX mathematical notation
AUDIT Prompt	$p_{\text{audit}} \in \mathcal{P}$	Prompt method where the LLM iteratively generates definitions followed by quality audits and refinements across multiple rounds for term $t_{(i,j)}$ in subject s_i. The output follows the structure $\mathcal{O}_{\text{audit}} = \{d_1, a_1, d_2, a_2, d_3\}$ where d_1 is the initial definition, a_1 is the first quality audit with refinement suggestions, d_2 is the first refined definition, a_2 is the second quality audit, and d_3 is the final refined definition.

Chapter 1

Introduction

1.1 Research Motivation

The emergence of large language models has created unprecedented opportunities for learners with complex and systematic educational needs, particularly those who previously faced barriers due to diverse knowledge foundations or specific learning constraints. By combining the vast parametric knowledge and extensive reasoning capabilities of these models with carefully designed prompt strategies, we can now instruct artificial intelligence to search, accumulate, and synthesize relevant knowledge at lightning speed while maintaining high levels of accuracy and reliability. This technological synergy transcends traditional educational limitations, offering a scalable solution that democratizes access to high-quality, rigorous academic content. Consequently, this shift holds profound positive social impacts, enabling equitable educational opportunities, reducing instructional burdens, and supporting lifelong professional development in an increasingly knowledge-driven society.

This dissertation proposes three prompt methods, structured knowledge accumulation (Cumulative Thinking Prompt), iterative refinement (AUDIT prompt), and fine-grained persona specification (Professor-Rank-Aware Prompt), which offer substantial social and economical benefits for STEM mathematics education. First, these methodologies can be deployed without costly fine-tuning or specialized infrastructure. This enables under-resourced institutions, online platforms, and self-directed learners to access mathematically precise definitions previously available only through expert tutoring or elite lecture notes, thereby leveling the playing field for students from diverse socioeconomic backgrounds or regions with scarce STEM faculty.

Second, the proposed methods benefit both instructors and students by significantly reducing the cognitive and temporal overhead associated with definition generation and comprehension. For instructors, these tools serve as reliable assistants that rapidly produce well-structured, formally accurate definitions, freeing them from repetitive manual

drafting to focus on higher-level instructional activities such as curriculum design and personalized feedback. For students, on-demand access to multiple, adaptively tailored definitions facilitates self-paced learning, clarifies subtle conceptual distinctions, and provides immediate support for unfamiliar terms, thereby enhancing both understanding and confidence in mathematical reasoning.

Third, this research supports lifelong learning and workforce development in an AI-driven economy. The ability to generate precise, customized mathematical definitions on demand empowers professionals in data science, engineering, finance, and healthcare to efficiently refresh or deepen their knowledge. This enables adaptation to rapidly evolving technical demands without requiring formal degree programs, contributing to economic mobility and sustained employability. Collectively, these impacts demonstrate that the proposed prompt innovations are not merely academic explorations but have concrete potential to make STEM mathematics education more effective, adaptable, and accessible for learners and educators alike.

1.2 Research Objectives

This dissertation pursues three sequential prompt engineering proposals designed to investigate whether and how different reasoning structures and context specifications can fundamentally alter the reasoning behaviors of large language models as they navigate the vast reasoning space $\mathfrak{R}$. The first objective is to develop and validate the Cumulative Thinking Prompt, a linear reasoning method $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ that structures knowledge accumulation through sequential recalling steps before final generation, enabling the model to leave less informative reasoning subspaces $\mathcal{R}_u \subset \mathfrak{R}$ and locate more valuable reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$. The second objective is to design and evaluate the AUDIT prompt, an iterative reasoning framework $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\to} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\to} \iota_3)^{\delta_2}\}$ that incorporates recursive quality audits and refinement cycles, allowing the model to dynamically adjust its navigation based on self-assessed output quality. The third objective is to examine the Professor-Rank-Aware Prompt Framework, investigating whether fine-grained persona specifications across eight academic ranks $\mathcal{P}_{\text{profs}} = \{p_{(\text{profs},1)}, p_{(\text{profs},2)}, \ldots, p_{(\text{profs},8)}\}$ can guide the model toward reasoning subspaces containing more valuable mathematical information compared to coarse-grained baseline specifications.

This research faces several key challenges. First, each proposed prompt method must demonstrate its unique reasoning advantages over established baselines, requiring careful experimental design that isolates the effects of the specific reasoning mechanism, whether linear knowledge accumulation, iterative refinement, or fine-grained persona specification, while controlling for confounding variables. Second, the research requires the construction of a customizable and generalizable generation and evaluation dataset spanning 18

mathematical subjects with 921 terminologies, ensuring balanced representation across domains while maintaining scalability for future applications. Third, a comprehensive evaluation framework must be established with trustworthy and applicable evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$ developed through automatic collaboration among different LLM agents with distinct roles, suggesting, evaluating, and optimizing, to make the mathematical term definition generation and evaluation procedure more reliable, less dependent on manual labeling, and fully transferable and customizable to new tasks and domains. Finally, methodological consistency must be maintained across all three studies to enable meaningful comparative analysis while allowing sufficient flexibility to accommodate the distinct characteristics of each proposed prompt approach.

1.3 Dissertation Outline

This dissertation is organized into six chapters. Chapter 1 introduces the motivation, objectives, and structure of the research. Chapter 2 presents the Cumulative Thinking Prompt study, investigating linear reasoning with structured knowledge accumulation through sequential recalling before definition generation, covering related works, methodology, results and analytics, and concluding with contributions and future directions. Chapter 3 presents the AUDIT prompt study, examining iterative reasoning with recursive quality audits and refinement cycles for on-the-fly definition optimization, including related works, methodology, results and analytics with head-to-head comparison against the Cumulative Thinking Prompt, and concluding with contributions and future directions. Chapter 4 presents the Professor-Rank-Aware Prompt study, investigating fine-grained persona specifications across eight academic ranks compared to a coarse-grained baseline, including related works, theoretical analysis of reasoning assumptions, a comprehensive case study with methodology, results and analytics covering holistic and subject-level performance with stability analysis, and concluding with contributions and future directions. Chapter 5 synthesizes the findings across all three studies, discussing what has been achieved and what should be improved. Chapter 6 envisions extending the proposed methodologies beyond mathematics into natural and social sciences, adapting them to diverse learner needs through flexible configurations, and applying them to professional workplace environments. And Appendix with extra information requested by the committee members. All the chapters are combined together to work as a cohesive narrative for three prompt methods that progressively builds from foundational linear reasoning to iterative refinement and finally to persona-aware context optimization, establishing a comprehensive framework for prompt-guided navigation in the reasoning space of large language models.

Chapter 2

Cumulative Thinking Prompt with Linear Reasoning: Sequential Reflections Before Mathematical Definition Generation

2.1 Introduction

The rapid advancement of large language models $\mathcal{M} = \{M_{(\text{a-gen})}, M_{(\text{a-eval})}, M_{(\text{m-eval})}, \pi_\theta\}$ has opened new possibilities for automating complex cognitive tasks, including the generation of precise mathematical definitions, a foundational activity in mathematics education and technical communication. Mathematical definitions serve as the building blocks of formal reasoning, requiring not only factual accuracy but also conceptual completeness, structural coherence, and adherence to disciplinary conventions. Yet the capacity of language models to generate high-quality mathematical definitions that meet the rigorous standards of college STEM education remains underexplored, particularly regarding how different prompt strategies influence the model's ability to navigate its vast reasoning space $\mathfrak{R} = \bigcup_{u=1}^{U} \mathcal{R}_u$ toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$. While existing prompt methods such as direct prompt, persona-aware prompt, and step-by-step reasoning have demonstrated utility across various tasks, they typically approach definition generation as a one-pass process without explicit mechanisms for structured knowledge accumulation prior to final output.

This study introduces the Cumulative Thinking Prompt p_{ct}, a novel linear reasoning method $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ designed to address the limitations of existing prompt strategies for mathematical definition generation. Unlike conventional approaches that generate definitions directly or decompose reasoning without preparatory knowledge scaffolding, the Cumulative Thinking Prompt instructs the language model to perform

five sequential recalling steps $\{r_1, r_2, \ldots, r_5\}$ that progress from essential to abstract understanding of the target mathematical term before generating the final definition. This structured pre-generation phase functions as a deliberate reasoning preparation mechanism, enabling the model to accumulate relevant knowledge and establish a coherent conceptual foundation that guides its subsequent navigation of the reasoning space $\mathfrak{R}$. By explicitly separating knowledge accumulation from definition generation, the proposed method aims to ensure that the model's final reasoning direction is informed by a comprehensive preparatory exploration rather than an implicit, one-step traversal.

To evaluate the effectiveness of the Cumulative Thinking Prompt, this study conducts a systematic comparison against three widely used baseline prompt methods: the direct prompt p_{direct}, which generates definitions without intermediate reasoning; the excellent-professor-aware prompt $p_{\text{e-pro}}$, which adopts a persona-based contextualization approach; and the step-by-step prompt p_{sbs}, which employs sequential reasoning decomposition. The evaluation utilizes a rigorously constructed dataset comprising 439 mathematical terminologies $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ distributed across 18 mathematical subjects $S = \{s_1, s_2, \ldots, s_{18}\}$, with definitions assessed across 11 evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$ spanning dimensions such as formal mathematical precision, conceptual completeness, and structural coherence. Through this comparative framework, the study seeks to determine whether structured knowledge accumulation prior to generation yields superior definition quality compared to existing linear reasoning strategies, and to provide insights into how different reasoning navigation patterns influence language model performance on complex cognitive tasks requiring both foundational understanding and formal precision.

2.2 Related Works

Following are extensive interpretations of typical prompt methods $\mathcal{P} = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$ that instruct the large language models $\mathcal{M} = \{M_{(\text{a-gen})}, M_{(\text{a-eval})}, M_{(\text{m-eval})}, \pi_\theta\}$ to linearly navigate in its vast reasoning space $\mathfrak{R} = \bigcup_{u=1}^{U} \mathcal{R}_u$ for multiple downstream tasks, which can be organized into two primary categories, each comprising three subcategories. The first main category, "Prompt for Linear Reasoning," encompasses methods that guide models through reasoning processes via specific reasoning strategies (such as architectural planning or path sampling), content distillation (progressive information refinement through sequential extraction), and context generation (creating intermediate knowledge prior to final answer formulation). The second main category, "Prompt Construction," focuses on techniques for building effective prompts through specific reasoning patterns (manual rationales or pattern-based selection), quality metrics (optimization via consistency, clustering, or adaptive selection), and reinforcement learning from human feedback (learning prompt effectiveness from pairwise comparisons or offline prediction models).

A defining characteristic shared across all 21 reviewed prompt method variations $\mathcal{P}_{\text{cand}} \subseteq \mathcal{P}$ is their reliance on linear reasoning $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ as the fundamental mechanism for constraining and navigating the holistic reasoning space $\mathfrak{R}$ of large language models $\mathcal{M}$. Within the framework of reasoning spaces $\mathfrak{R}$, subspaces $\mathcal{R}_u \subset \mathfrak{R}$, and directions $\rho_{(u,w)} \in \mathcal{R}_u$, linear reasoning can be understood as a structured approach to progressively narrow the vast holistic reasoning space $\mathfrak{R}$ into increasingly focused reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$, ultimately guiding the model toward specific, verifiable reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$. This architectural commitment manifests across the identified subcategories through distinct but complementary mechanisms. Some methods explicitly construct reasoning subspaces $\mathcal{R}_u$ by enforcing predetermined sequences of operational stages, effectively partitioning the holistic space into ordered regions that share common procedural characteristics. Others decompose complex problems into hierarchical sub-tasks, recursively refining the reasoning subspace $\mathcal{R}_{\mathcal{I},\mathcal{D}}$ as each sub-problem is solved and its output becomes the foundation for subsequent steps. Still others employ progressive content distillation, creating cascading constraints that systematically reduce the subspace dimensionality until a single, well-defined reasoning direction $\rho_{\mathcal{I},\mathcal{D}}$ emerges. Methods focused on prompt construction $\mathcal{P}_{\text{prop}} \cup \mathcal{P}_{\text{base}}$ externalize this subspace navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D}) = \mathfrak{R}_{\mathcal{I},\mathcal{D}} \to \mathcal{R}_{\mathcal{I},\mathcal{D}} \to \rho_{\mathcal{I},\mathcal{D}}$ by selecting demonstrations $\mathcal{D} = \{d_1, d_2, \ldots, d_K\}$ that exemplify particular reasoning patterns, optimizing the selection criteria to ensure that the induced reasoning directions remain within desirable subspaces $\mathcal{R}_u$. Across all these approaches, the unifying principle is the explicit orchestration of cognitive steps that progressively structure the model's traversal from the undifferentiated holistic reasoning space $\mathfrak{R}$ through successively constrained subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}}$, culminating in specific reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ that yield accurate and interpretable solutions. This linear reasoning paradigm $\mathcal{L}$ thus functions as a spatial constraint mechanism, transforming the unbounded generative capacity of LLMs $\mathcal{M}$ into a disciplined search through hierarchically organized reasoning territories $\mathfrak{R}$.

Table 2.1: Cumulative Thinking Prompt: Linear Reasoning-Aware Prompt Citations

Main Category	Subcategory	Prompt Methods
Prompt for Linear Reasoning	Prompt for Linear Reasoning with Specific Reasoning Strategy	• Overarching Prompting [1] • PEARL (Planning with Executable Actions for Reasoning over Long Documents) [2] • Self-Consistency [3]
	Prompt for Linear Reasoning with Content Distillation	• R^3 (Review, Rephrase and Resolve) [4] • SCoT (Structured Chain-of-Thought) [5] • CoF-CoT (Coarse-to-Fine Chain-of-Thought) [6] • Least-to-Most Prompting [7]
	Prompt for Linear Reasoning with Context Generation	• Cue-CoT (Cue-Chain-of-Thought) [8] • TSGP (Two-Stage Generative Prompting) [9] • Thought Experiments [10] • Generated Knowledge Prompting [11]
Prompt Construction	Prompt Construction with Specific Reasoning Strategy	• Chain-of-Thought Prompting [12] • Pattern-CoT [13] • Active-Prompt [14]
	Prompt Construction with Quality Metrics	• Prompt Demonstration Explanation Selection [15] • COSP (Consistency-based Self-adaptive Prompting) [16] • Auto-CoT [17] • Dynamic Program Prompting [18] • IAP (Instance-Adaptive Prompting) [19]
	Prompt Construction with RLHF Language Model	• Prompt-OIRL [20] • ParetoPrompt [21]

2.2.1 Linear-Reasoning Prompt Applications

Prompt for Linear Reasoning with Specific Reasoning Strategy

Overarching Prompting [1] is a few-shot text prompt method that combines both specific instructions and example demonstrations to teach a language model how to build and use a high-level summary, called an "archetype," for solving complicated problems. The instructions are the clear, manually-written sentences that tell the model what to do, such as "Let's start with some high-level thinking," "The problem statements can be abstracted as follows," and "From a high-level perspective, the problem could be addressed as follows." These guide the model through the two main steps: first creating the archetype and then forming a strategy. The demonstrations are the two examples provided within the prompt for each task. These examples show the model exactly what a good archetype and a good strategy look like for a specific type of problem, such as a math word problem or a physics question. By seeing these examples, the model learns the expected level of abstraction and the style of the high-level plan. Together, the instructions tell the model what to do, and the demonstrations show the model how to do it. This combination effectively guides the model to first strip away distracting low-level details to grasp the core of the problem, and then use that clearer understanding to plan its reasoning, which ultimately helps it solve complex tasks more accurately. On

the GSM8K [22] mathematical reasoning dataset, applying Overarching Prompting to ChatGPT improved its solve rate to 81.34%, compared to 79.08% for standard Chain-of-Thought. On the MMLU College Physics task [23], Overarching Prompting with ChatGPT achieved a 57.84% solve rate, outperforming Chain-of-Thought at 54.70%. These results demonstrate consistent improvements across different models and task types.

PEARL (Planning with Executable Actions for Reasoning over Long Documents) [2] is a multi-stage prompt framework specifically designed to improve how large language models reason over lengthy texts. The process begins with an action mining stage, where the model is prompted, using a few manually crafted seed actions as examples, to generate a broader set of task-specific, reusable actions from training questions. Each action is formatted like a program function with a natural language definition. Next, in the plan generation stage, the model is given a new question and, through few-shot prompt with demonstrations of good plans, formulates a sequence of these mined actions. This plan acts as a blueprint, where the output of one action can be used as an input for another. Finally, in the plan execution stage, the model executes the plan step-by-step. For each step, it is given a zero-shot prompt containing the long document, the specific action to perform, and the results from previous steps, allowing it to generate a detailed output. To ensure the quality of the few-shot demonstrations used in plan generation, the framework incorporates a self-refinement mechanism where the model corrects its own plans based on error messages or execution results before they are used as examples. The principal evidence for PEARL's effectiveness comes from experiments on a challenging subset of the QuALITY [24] dataset. On questions requiring long-context understanding, PEARL achieved an accuracy of 70.9%, significantly outperforming the zero-shot GPT-4 [25] baseline which scored 64.3%. For the most complex questions that required the full document, PEARL's performance was even stronger at 72.4% compared to the baseline's 61.9%, demonstrating that its structured, multi-stage approach leads to more accurate answers.

Self-Consistency [3] is a decoding strategy designed to replace the standard greedy decoding approach used with Chain-of-Thought (CoT) [12] prompt. It functions as an extension to manually-crafted, few-shot CoT prompts by altering how the language model generates its response after the prompt is given. The process involves three main steps. First, the language model is prompted with the manually written chain-of-thought exemplars. Second, instead of the model generating a single, most probable reasoning path (greedy decoding), the researchers sample multiple diverse outputs from the model's decoder, creating a set of different reasoning paths that might lead to various final answers. Third, these generated answers are aggregated, typically by taking a majority vote, to select the single most consistent answer from the set. This method is grounded in the intuition that for complex reasoning tasks, there are often several valid ways to arrive at

a correct solution, and these correct solutions are more likely to agree on a final answer than incorrect ones. By marginalizing over the sampled reasoning paths, self-consistency [3] aims to identify the answer with the strongest support, thereby improving the overall reasoning accuracy of the model.

Algorithm 2 Pseudocode: Linear Reasoning with Specific Reasoning Strategy

Require: Source of demonstrations $\mathcal{D} = \{d_1, d_2, \ldots, d_K\}$, Target LLM $M_{\text{a-gen}} \in \mathcal{M}$ for generation, Reasoning strategy configuration $\mathcal{C}_{\text{reason}}$

Ensure: Optimized prompt with reasoning strategy

Candidate Generation
Generate reasoning demonstrations from $\mathcal{D}$ according to $\mathcal{C}_{\text{reason}}$:
    - High-level abstraction construction
    - Action or plan formulation
    - Step-by-step rationale generation
Demonstration Selection
Select K demonstrations $\mathcal{D}_{\text{selected}} \subseteq \mathcal{D}$ based on $\mathcal{C}_{\text{reason}}$:
    - Representativeness of reasoning patterns
    - Diversity of problem-solving approaches
    - Quality of intermediate reasoning steps
Inference with Reasoning Strategy
Construct prompt with $\mathcal{I}$ and $\mathcal{D}_{\text{selected}}$ according to $\mathcal{C}_{\text{reason}}$
Apply reasoning mechanism according to $\mathcal{C}_{\text{reason}}$:
    - Generate structured reasoning path
    - Execute sequential reasoning steps
    - Aggregate multiple reasoning paths if applicable
Generate output via $M_{\text{a-gen}}$
return Optimized reasoning-enhanced prompt

Prompt for Linear Reasoning with Content Distillation

The R^3 **(Review, Rephrase and Resolve)** [4] prompt method operates through three distinct and sequential stages of interaction with the language model, each with a specific instruction. In the first stage, called Review, the model is given a prompt containing several example problems (few-shot demonstrations) that include both relevant and irrelevant information. The instruction for this stage asks the model to read the question and extract only the key sentences from the noisy context that are essential for finding the answer. For the problem of interest, the model then outputs a numbered list of these crucial sentences, explicitly noting which pieces of information are irrelevant. In the second stage, Rephrase, the model is provided with a new prompt that includes the original problem along with the key sentences it just extracted. This prompt instructs the model to translate those natural language sentences into a more abstract representation using explicit variables and their numerical values (e.g., "Number of lost marbles = 8"). The final stage, Resolve, gives the model the original question along with the generated variables, asking it to form an equation and calculate the final answer. This structured approach forces the model to focus, simplify, and then solve, preventing it from being derailed by distracting details. The principal evidence for the effectiveness of this method is that, when tested on five different reasoning datasets containing misleading context, R^3 prompting achieved an average accuracy improvement of 3.7% over the next best prompt baseline when using the

GPT-3.5-turbo [26] engine. Specifically, on the SVAMP [27] dataset, which is known for its tricky in-topic noise, it raised the state-of-the-art performance from 83.6% to 87.3%.

SCoT (Structured Chain-of-Thought) [5] prompt is a method where the overall task of generating a document-grounded conversation is deliberately divided into a sequence of smaller, more manageable sub-tasks. For each of these sub-tasks, the researchers manually create text-based prompts that include specific instructions and a few examples (few-shot) to show the language model exactly what is expected. In practice, this means before the model writes a final answer, it first goes through guided steps: one prompt asks it to determine if the user's question can actually be answered from the given document (answerability classification), and another prompt asks it to identify the specific sentences within the document that contain the relevant information (sentence selection). Only after these steps are completed does a final prompt guide the model to generate the response, now equipped with a clear understanding of whether an answer exists and where it is located. This structured, step-by-step approach is designed to prevent the model from skipping ahead and making up information, instead forcing it to follow a logical and verifiable reasoning path. The principal evidence for its effectiveness is that the SCoT method, when including the answerability classification and sentence selection states with a FLAN-UL2-20B [28] assistant model, increased the agent's faithfulness to the source document by up to 16.8% compared to a simpler method that generated the conversation in one step, as measured by lexical precision and factual consistency checks.

The **CoF-CoT (Coarse-to-Fine Chain-of-Thought)** [6] method is a manually-crafted, text-based prompt approach designed to improve how large language models handle complex language understanding tasks. Instead of asking the model to produce an answer in one go, CoF-CoT breaks the overall task into a specific sequence of five smaller, more manageable steps that progress from broad, sentence-level understanding to precise, word-level details. As detailed in Section 3 of the paper, these steps are: 1) generating an Abstract Meaning Representation [29] (AMR) graph, which captures the core concepts and relationships in the sentence; 2) predicting the overall intent of the utterance; 3) identifying the key phrases or slot values; 4) assigning a type to each of those key phrases from a given vocabulary; and finally 5) assembling everything into a final structured output called a Logic Form. A key feature of this method is that the prompt for each step explicitly includes the outputs generated from all previous steps, a process called conditioning, which ensures the later, finer-grained decisions are informed by the earlier, broader understanding. For example, the model uses the AMR graph and predicted intent from the first two steps to help it identify the correct slot values in step three. The primary evidence for its effectiveness is presented in Table 1, where CoF-CoT consistently outperforms all other prompt methods across multiple metrics. For instance, on the MTOP [30] dataset in a zero-shot setting, CoF-CoT achieved

57.67% intent accuracy, a substantial improvement over the next best method's 45.67% (Least-to-Most [7]). Similarly, on the MASSIVE dataset [31] under the same conditions, it reached 89.00% intent accuracy, significantly higher than the 73.66% achieved by the best baseline method, Complex CoT [32].

Least-to-Most Prompting [7] is a manually-crafted, few-shot text prompt strategy designed to help language models solve complex problems by breaking them down into simpler parts. The process is explicitly divided into two stages, both implemented through carefully designed prompts. In the first stage, called decomposition, the user provides the model with a few examples that demonstrate how to break a complicated question into a list of smaller, easier subproblems. For instance, a prompt might show how a multi-step math question can be split into individual questions, each asking for a single piece of information. The model then applies this learned pattern to decompose the new, unseen problem it is given. In the second stage, called subproblem solving, the model is guided through a sequence of prompts to solve each of these smaller subproblems one by one. Critically, the answer generated for each subproblem is included in the prompt for the next one, allowing the model to build on its previous work and eventually arrive at the final answer for the original, complex problem. The principal evidence for the effectiveness of this method comes from its performance on the SCAN [33] benchmark, a test of compositional generalization. When tested on the most challenging "length split," where the commands in the test set are longer than any shown in the prompts, least-to-most [7] prompt achieved an accuracy of 99.7% using only 14 examples. This dramatically outperformed chain-of-thought prompt, which only reached 16.2% accuracy on the same test.

Algorithm 3 Pseudocode: Linear Reasoning with Content Distillation

Require: Input query x, Target LLM $M_{\text{a-gen}} \in \mathcal{M}$ for generation, Number of distillation stages S, Distillation strategy $\mathcal{T}_{\text{distill}}$ (e.g., sequential decomposition, coarse-to-fine, structured sub-tasks)
Ensure: Final output y

Content Distillation
Decompose input x into S sequential components:
    - Extract essential information while filtering irrelevant content
    - Transform natural language into structured or abstract representations
    - Break complex problems into ordered subproblems
Sequential Reasoning
for stage $s = 1$ **to** S **do**
    Condition on outputs from previous stages
    Generate stage-specific output via $M_{\text{a-gen}}$
end for
Resolution
Synthesize final output based on all distilled components
return Final output y

Prompt for Linear Reasoning with Context Generation

The **Cue-CoT (Cue-Chain-of-Thought)** [8] prompt framework is designed to enhance Large Language Model (LLM) responses in dialogue by explicitly incorporating reasoning about a user's underlying status, which includes their personality, emotional state, and psychological condition as inferred from the conversation context. This framework consists of two distinct variations. The first, One-step Cue CoT (**O-Cue CoT**), uses a single, comprehensive prompt that instructs the LLM to perform two tasks in one go: first, to reason about and generate a description of the user's status based on the dialogue, and second, to immediately generate a final response that takes this inferred status into account. The primary advantage of this method is its simplicity, but the paper notes it can lead to less detailed reasoning or cause the model to struggle with long and complex instructions. The second variation, Multi-step Cue CoT (**M-Cue CoT**), decomposes the process into two separate, sequential prompts. The first prompt instructs the model solely to analyze the dialogue and output an inference about the user's status. The second prompt then provides the model with the original dialogue plus this inferred user status, and instructs it to generate a final response. This step-by-step approach offers greater clarity, allows for the intermediate reasoning to be verified or stored, and makes it easier for the model to follow instructions. The principal evidence supporting the superiority of this multi-step approach comes from extensive comparative experiments. For instance, across six different datasets, the M-Cue CoT method consistently achieved higher win rates than both standard prompt and O-Cue CoT. In a specific zero-shot evaluation on the Chinese Zhihu dataset with the ChatGPT model, M-Cue CoT achieved a helpfulness win rate of 95.57% and an acceptability win rate of 65.22%, demonstrating its robustness and effectiveness.

TSGP (Two-Stage Generative Prompting) [9] is a framework designed to answer commonsense questions without any training on labeled data. It operates in two distinct stages, both driven by manually-crafted text prompts. In the first stage, called Knowledge Generation, the researchers feed a pre-trained language model a prompt that includes a simple instruction, such as "Generate some knowledge about the concepts in the input," followed by a few handwritten examples (few-shot demonstrations) of questions paired with helpful knowledge statements. When a new question is inserted into this prompt, the model generates several pieces of commonsense knowledge related to that question, making the implicit information stored in its parameters explicit. Because this generated knowledge can sometimes be noisy or unhelpful, the framework then selects the single most relevant piece by calculating its pointwise mutual information with the original question. In the second stage, called Answer Generation, the framework uses another manually-crafted prompt. This prompt contains instructions, the selected knowledge statement from the first stage, the original question, and more few-shot demonstrations showing how to derive answers from knowledge. The language model then generates

numerous possible answers, or pseudo-answers, based on this combined context. Finally, to predict the correct answer from the given choices, the framework computes the semantic similarity between each of the generated pseudo-answers and each provided answer option, using a separate sentence-embedding model, and votes for the option with the highest overall similarity. The principal evidence supporting this framework's effectiveness comes from its performance on three different commonsense reasoning benchmarks. On the CommonsenseQA [34] dataset, TSGP achieved 49.1% accuracy using the GPT2-XL model, which is a substantial 16.8% improvement over the baseline model that simply scored answer options without any prompt. It also outperformed the previous best unsupervised method, SEQA [35], by 14.3%. On OpenBookQA [36], TSGP reached 44.4% accuracy, beating the baseline by 21.6% and SEQA by 11.0%. On SocialIQA [37], it achieved 51.5% accuracy, outperforming the baseline by 8.7% and SEQAA by 4.0%. These consistent improvements across diverse tasks demonstrate that by using structured prompts to make implicit knowledge explicit, the TSGP framework enables language models to perform unsupervised commonsense reasoning more effectively without any fine-tuning.

Thought Experiments [10] prompt framework is a method designed to help language models reason through moral questions by guiding them to think about alternative scenarios and hypothetical situations. The approach uses a series of manually written text instructions that break down the reasoning process into five clear steps. First, the model is asked to generate "what if" questions about the given moral situation, questions that explore different possibilities or missing details. Second, the model answers those questions in detail. Third, it summarizes what it has learned from considering these counterfactuals and decides whether the original action was morally wrong. Fourth, because there can be multiple ways to think about a moral issue, the model generates several different summaries and then selects the best one. Finally, using that chosen summary, the model answers the original multiple-choice question about the moral scenarios. This framework can be used in two ways: in a zero-shot setting where the model receives only the instructions and no examples, or in a few-shot setting where it first sees five manually crafted examples that demonstrate the process. Unlike simpler reasoning methods that follow a single straight line of thought, this approach encourages the model to explore multiple perspectives by asking hypothetical questions. For instance, when considering a scenario about cutting children's hair, the model asks questions like "Were the children okay with it?" or "What was the reason for cutting their hair?", questions that address details not clearly stated in the original scenario. By exploring these different possibilities, the model can make a more informed and accurate moral judgment based on a fuller understanding of the situation.

Generated Knowledge Prompting [11] operates as a two-stage text-prompt method designed to improve commonsense reasoning. In the first stage, knowledge generation, the

researchers use a large language model like GPT-3 [26] and prompt it with a manually-crafted few-shot prompt. This prompt contains a clear instruction and five human-written demonstration pairs, each showing a question and a corresponding knowledge statement that helps answer that question without directly giving away the answer. By feeding a new question into this few-shot prompt, the language model generates multiple relevant knowledge statements about the topic of the question. In the second stage, knowledge integration, these generated statements are then used as additional input for a separate inference model, such as T5 [38] or UnifiedQA [39]. The inference model makes predictions by looking at each knowledge statement combined with the original question, and the method automatically selects the prediction that gets the strongest support from any single knowledge statement. This two-stage prompt approach is flexible because it does not depend on having a structured knowledge base like ConceptNet [40] to look up facts, nor does it require training the model specifically to work with external knowledge. Instead, it leverages the knowledge already stored inside large language models and makes that knowledge explicit through prompt, effectively turning complex commonsense questions into simpler reasoning tasks such as deduction, analogy, or numerical comparison that standard language models can handle more reliably. The principal evidence for the effectiveness of this method comes from the experimental results reported in Table 3 of the paper, where Generated Knowledge Prompting achieves new state-of-the-art performance on three out of four commonsense reasoning benchmarks, improving accuracy on NumerSense [41] from 66.18% to 72.47%, on CommonsenseQA 2.0 [42] from 70.2% to 73.03%, and on QASC [43] from 76.74% to 80.33% when applied to existing top-performing models.

Algorithm 4 Pseudocode: Linear Reasoning with Context Generation

Require: Input query x, Target LLM $M_{\text{a-gen}} \in \mathcal{M}$ for generation, Number of generated contexts K, Context selection strategy $\mathcal{S}_{\text{select}}$
Ensure: Final output y
Stage 1: Context Generation
Generate context information relevant to input x via $M_{\text{a-gen}}$:
    - User status inference (e.g., personality, emotion, psychological condition)
    - Commonsense knowledge statements
    - Counterfactual or hypothetical scenarios
    - Missing details or alternative perspectives
Obtain generated context set $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$
Stage 2: Context Selection
if $\mathcal{S}_{\text{select}} =$ single **then**
    Select optimal context c^* from $\mathcal{C}$ using:
        - Pointwise mutual information with input
        - Semantic similarity scoring
        - Heuristic or learned ranking
else if $\mathcal{S}_{\text{select}} =$ multiple **then**
    Keep all contexts for ensemble reasoning
end if
Stage 3: Output Generation
Generate final output via $M_{\text{a-gen}}$ using:
    - Original input x
    - Selected context(s) c^* or $\mathcal{C}$
    - Task-specific instructions
if $\mathcal{S}_{\text{select}} =$ multiple **then**
    Aggregate outputs via voting or similarity-based selection
end if
return Final output y

2.2.2 Linear-Reasoning Prompt Construction

Prompt Construction with Specific Reasoning Strategy

Chain-of-Thought Prompting [12] is a few-shot, text-based prompt technique where each demonstration provided to the language model is manually augmented with a natural language "chain of thought." This means the prompt does not just show the model a question and its final answer, but also includes a written-out, step-by-step explanation of the reasoning process needed to proceed from the question to the final answer. For example, instead of using a prompt simply showing "Q: 15 + 15 = 30" and "A: 30", a chain-of-thought [12] demonstration would show its reasoning rationals "Q: 15 + 15 = 30? A: The problem is 15 plus 15. Adding the ones column, 5 plus 5 equals 10, so you write down 0 and carry over 1. Then, in the tens column, you add the carried-over 1 to the two tens digits, which are 1 and 1, making 1 + 1 + 1 = 3. The final number is 30. The answer is 30." The core idea is that this specific prompt structure explicitly models for the model how a complex problem can be broken down into simpler, sequential steps. By seeing this structure in a few examples, a sufficiently large language model can then apply the same type of step-by-step reasoning to new, unseen problems, effectively unlocking its latent capacity for multi-step tasks like math, commonsense, and symbolic reasoning. The proposed method demonstrates significant effectiveness, particularly when

applied to sufficiently large language models. For instance, on the challenging GSM8K benchmark of math word problems, prompting the PaLM 540B [44] model with just eight chain-of-thought exemplars achieved a new state-of-the-art accuracy, surpassing even a finetuned GPT-3 with a verifier. Furthermore, this method improved performance on the StrategyQA [45] commonsense reasoning task, with PaLM 540B reaching 75.6% accuracy, outperforming the prior state-of-the-art of 69.4%. Lastly, in symbolic reasoning tasks like sports understanding, the same model achieved a score of 95.4%, which is notably higher than the unaided sports enthusiast baseline of 84%.

Pattern-CoT [13] is a prompt demonstration generation and selection method designed to construct few-shot examples for reasoning tasks by prioritizing the logical structure of problem-solving over surface-level semantics. The method begins by collecting a pool of seed demonstrations, each consisting of a question, its intermediate reasoning steps (rationale), and the answer. These can be sourced from existing training data or, when unavailable, generated by a large language model using a zero-shot prompt approach. The core innovation lies in extracting "reasoning patterns" from these rationales, which are sequences of fundamental operation tokens specific to the task domain, such as mathematical symbols ($+, -, \times, \div$) for arithmetic problems, or terms like "day," "week," and "month" for date-related tasks. These operation tokens are identified either through prior domain knowledge or by prompt a large model to generate them. Once extracted, each rationale is converted into a pattern sequence (e.g., $\times, -$ for a two-step calculation). These patterns are then encoded into vector representations using a sentence transformer model [46] and grouped into clusters via k-means clustering. This clustering ensures that the selected demonstrations are diverse in terms of the reasoning approaches they represent. From each cluster, a representative demonstration is sampled to form the final prompt set, where the number of clusters k is determined adaptively based on the number of identified operation types and the total sample size using a formula $k = \lceil \frac{1}{2} \times n \times (1 + \log N) \rceil$ (where n is the number of operation types and N is the number of samples). This process yields a prompt that exposes the language model to varied reasoning paths, reducing the risk of bias from repetitive patterns and improving the model's ability to select the correct approach for downstream tasks. The effectiveness of Pattern-CoT is demonstrated through extensive experimental results reported in the paper. On the LLaMA-2-7B model [47], Pattern-CoT achieved 79.66% accuracy on the MultiArith [48] dataset, outperforming the semantic-based Auto-CoT [17] method which scored 76.00%. Similarly, on the Coin Flip dataset [12], Pattern-CoT reached 59.40% accuracy compared to Auto-CoT's 51.20%.

Active-Prompt [14] is a method designed to build better few-shot chain-of-thought prompts by focusing on the model's own uncertainty. Instead of relying on humans to guess which example questions might be helpful, the process begins by having the large language model itself analyze a pool of unlabeled questions. For each question, the model

is asked to generate an answer multiple times. If the generated answers vary widely, then that question would be flagged as having high "uncertainty." The logic behind this selection decision is that these are the questions where the model struggles most and where a clear, step-by-step example would be most beneficial for being few-shot demonstrations. Humans then step in to write a reasoning chain for only these carefully selected, high-uncertainty questions, showing exactly how to arrive at the correct answer. This new, small set of human-annotated examples becomes the demonstrations of the few-shot prompt. The principal evidence supporting this approach is found in the paper's experimental results, such as on the GSM8K [22] math dataset. Using the GPT-3.5-turbo [26] model, the standard chain-of-thought method with arbitrarily chosen examples achieved 74.2% accuracy. In contrast, Active-Prompt, which used its uncertainty-based selection to pick the examples to annotate, achieved a significantly higher accuracy of 77.1%, demonstrating that a prompt built around the model's uncertain questions is more effective.

Algorithm 5 Pseudocode: Prompt Construction with Specific Reasoning Strategy

Require: Source of demonstrations $\mathcal{S}$ (training data, unlabeled inputs, or manual examples), Target LLM $M_{\text{a-gen}} \in \mathcal{M}$ for generation, Reasoning strategy type $\mathcal{T}_{\text{reason}}$, Number of demonstrations K

Ensure: Optimized prompt with reasoning demonstrations

Candidate Generation
Generate reasoning demonstrations from $\mathcal{S}$ via:
    - Manual crafting with step-by-step rationales
    - Pattern extraction and clustering from existing rationales
    - Uncertainty-based selection from unlabeled inputs
Demonstration Selection
Select K demonstrations based on:
    - Diversity of reasoning structures
    - Uncertainty or difficulty levels
    - Representativeness of reasoning patterns
Inference
for each test input x_{test} **do**
    Construct prompt with selected reasoning demonstrations
    Generate step-by-step reasoning and output via $M_{\text{a-gen}}$
end for
return Optimized reasoning-enhanced prompt

Prompt Construction with Quality Metrics

Prompt Demonstration Explanation Selection [15] functions as an optimization framework designed to refine the explanatory content within few-shot text prompts. The research employs a leave-one-out generation scheme. Using the other $K-1$ exemplars (complete with their seed explanations and answers) as context, they prompt a large language model (LLM) to generate new explanation-answer pairs for the held-out question q_i, keeping only those generations that produce the correct answer a_i. This process is repeated with temperature scaling to produce a diverse candidate pool $\hat{E}_i$ of up to 8 valid explanations per question, which also includes the original seed explanation. Subsequently, the selection phase aims to find the optimal combination $C = \{e_i\}_{i=1:K}$ (one candidate per

exemplar) from this discrete search space. Given the high computational cost of evaluating combinations directly on a development set, the method introduces two efficient proxy metrics to prioritize promising candidates: one based on one-shot silver accuracy (S_{OSAcc}) on an unlabeled development set V, and another based on one-shot log likelihood (S_{OSLL}) computed from the few-shot exemplars themselves. After using these metrics to rank and select a limited number of high-scoring combinations, the method performs a final, more expensive evaluation using the actual objective O, accuracy against a silver-labeled version of the development set, to identify the single best-performing combination of reasoning rationales. The principal evidence supporting this method is that the optimized prompts consistently outperform the original human-written ones across four different reasoning tasks. Quantitatively, the optimized explanations achieved substantial gains over the seed explanations, with accuracy improvements of 3.3% on GSM [22], 4.3% on ECQA [49], and 7.1% on E-SNLI [50]. Furthermore, the proxy metrics proved effective, as combinations ranked by $\mathcal{S}_{\text{OSAcc}}$ in the top 16 achieved up to 1.4% higher accuracy than random selection on E-SNLI, while those ranked by $\mathcal{S}_{\text{OSLL}}$ outperformed random selection by 2.0% on ECQA. The framework's robustness was further demonstrated by its ability to find better-performing explanations even with a reduced computation budget of 20 passes, yielding improvements of 2.0%, 4.0%, and 6.0% on GSM, ECQA, and E-SNLI respectively.

COSP (Consistency-based Self-adaptive prompt) [16] is a two-stage method designed to improve the reasoning performance of large language models without requiring any human-provided, labeled examples. In the first stage, the model is asked to solve each test question multiple times using a simple zero-shot chain-of-thought trigger phrase like "Let's think step by step," generating a pool of its own responses, each consisting of a reasoning path and a final answer. From this pool, COSP selects the most effective question-answer-reasoning pairs to serve as in-context demonstrations. The selection process is guided by a scoring function that incorporates three key criteria: consistency, which measures how often the model arrives at the same final answer for a given question across multiple attempts (assuming that answers with higher agreement are more likely to be correct); diversity, which ensures the selected set of demonstrations covers different types of questions and reasoning patterns to avoid narrow guidance; and repetitiveness, which penalizes demonstrations where the reasoning text repeats itself, as such repetitive patterns can mislead the model. In the second stage, these self-selected, high-quality demonstrations are prepended to each original test question, and the model is queried again to produce a final answer. The principal evidence for the method's effectiveness is found in the paper's experimental results. Across six arithmetic and logical reasoning tasks using the PaLM-62B model [44], COSP achieved an average accuracy of 66.28%. This represents a substantial improvement of 10.40% over the standard Zero-shot CoT with self-consistency [3] baseline, which averaged 55.88%. Furthermore, it achieved near

identical performance with the Few-shot CoT baseline that uses five handcrafted, labeled demonstrations, which averaged 66.37%, demonstrating that COSP can eliminate the need for manual prompt engineering while matching or exceeding its performance.

Auto-CoT [17] is a few-shot prompt method designed to eliminate the need for manually creating demonstration examples. Its process begins by taking all the available test questions and grouping them into clusters based on their semantic similarity, which ensures the selected examples will be diverse. From each of these clusters, it picks one question that is most representative of that group. For each selected question, it uses a simple zero-shot prompt, "Let's think step by step," to instruct a large language model to generate a reasoning chain that leads to an answer. The method then applies basic quality checks, such as preferring shorter questions and reasoning chains with a limited number of steps, to filter out potentially flawed generations. The final output of this process is a set of model-generated demonstrations, each consisting of a question, its step-by-step reasoning, and the answer. This set is then used as the few-shot examples in a prompt for answering new, unseen test questions. The principal evidence for the effectiveness of this method is shown in the paper's main results table, where on ten different reasoning tasks, Auto-CoT achieved an average accuracy competitive with Manual-CoT. For instance, on the MultiArith [48] dataset, Auto-CoT scored $92.0\% \pm 1.7\%$, compared to Manual-CoT's 91.7%, demonstrating that automatically generated demonstrations can perform as well as human-crafted ones without the need for human intervention.

Dynamic Program Prompting [18] is a few-shot method designed to improve how large language models solve math word problems by using training data more effectively. The process begins by automatically annotating each problem in the training dataset with a Python program that correctly solves it. This is done by prompting Codex [51] multiple times until it generates a program whose executed result matches the known correct answer. Once this annotated dataset is created, the method dynamically selects prompts for a new test question. Instead of using the same fixed set of example problems for every query, it retrieves the M most similar problems from the annotated training data based on semantic similarity by using sentence embeddings. These similar problems, along with their Python program solutions, are then provided as in-context examples in the prompt. This approach helps the model by showing it how to solve problems that are structurally similar to the new one, which is especially valuable for complex or diverse question types where a generic set of examples would be insufficient. The method's effectiveness is strongly supported by the empirical results; on the complex and noisy MathQA [52] dataset, this approach achieved a significant improvement, outperforming the strong PAL baseline by over 30 percentage points (61.7% vs. 30.0%), while this dynamic selection strategy consistently surpassed random exemplar selection, which scored only 34.0% on the same MathQA dataset, highlighting the critical role of relevance, and the benefit

extended to the SVAMP [27] dataset, where the proposed method improved performance by 2.9 points over the reproduced PAL baseline (80.3% vs. 77.4%).

IAP (Instance-Adaptive Prompting) [19] is a method designed to choose the optimal instruction-only prompt for each individual question when using large language models for step-by-step reasoning. Instead of using the same prompt for every question, IAP looks at how information flows inside the model when it processes a question with a candidate prompt. It calculates a single combined score based on three measurements: how well the prompt absorbs information from the question, how well the reasoning steps pull information directly from the question, and how well the reasoning steps incorporate guidance from the prompt. This score tells researchers whether a particular prompt is likely to lead to a correct answer for that specific question. IAP comes in two versions. The first, Sequential Substitution, tests prompts one by one in a fixed order until it finds one whose score exceeds a pre-set threshold, then uses that prompt to generate the answer. The second, Majority Vote, calculates scores for all available prompts, picks the top three with the highest scores, and selects the most common answer among them as the final result. Both versions draw from a diverse pool of candidate prompts that includes helpful instructions, misleading ones, and even completely unrelated phrases. When tested across mathematical word problems, logical reasoning tasks, and commonsense questions, both IAP approaches consistently outperformed the best single prompt designed for an entire task, as well as more complex prompt optimization methods like OPRO [53] and Self-Discover [54], demonstrating that tailoring prompts to individual instances yields better overall results.

Algorithm 6 Pseudocode: Prompt Construction with Quality Metrics

Require: Target LLM M, Task inputs $\mathcal{X}_{\text{test}} = \{x_j\}_{j=1}^{T}$, Candidate source $\mathcal{S}$ (training data, prompt templates, or self-generation), Quality metrics $\mathcal{Q}_{\text{proxy}}$ for efficient prioritization, Target objective $\mathcal{O}$ for final evaluation, Number of top candidates K

Ensure: Optimized prompt components and final outputs $\mathcal{Y}_{\text{test}}$ for all test inputs

Stage 1: Candidate Generation
Generate candidate pool $\mathcal{C}$ from source $\mathcal{S}$ via:
    - Retrieval from training data, or
    - Self-generation with temperature scaling, or
    - Predefined prompt templates
Stage 2: Quality-Aware Prioritization
for each test input $x \in \mathcal{X}_{\text{test}}$ **do**
    Compute proxy scores for candidates $c \in \mathcal{C}$ using $\mathcal{Q}_{\text{proxy}}$
        (e.g., consistency, diversity, similarity, information flow, or silver accuracy)
    Rank candidates and retain top-K candidates $\mathcal{C}_{\text{top}}$
    Stage 3: Final Selection & Generation
    Evaluate $\mathcal{C}_{\text{top}}$ using target objective $\mathcal{O}$ (e.g., answer agreement, silver-label accuracy)
    Select optimal candidate $c^* = \arg\max_{c \in \mathcal{C}_{\text{top}}} \mathcal{O}(c, x)$
    Generate final output y via M using selected prompt components
    Store y in $\mathcal{Y}_{\text{test}}$
end for
return $\mathcal{Y}_{\text{test}}$ with associated optimized prompt components

Prompt Construction with RLHF Language Model

Prompt-OIRL [20] is designed to suggest the most effective prompt for each individual question, rather than relying on a single prompt for all questions. It works in two main stages. First, it learns from an "offline" dataset, which is existing information collected from past experiments where various prompts, such as "Let's think step by step," were tested on arithmetic problems. This dataset contains records of which prompts led to correct or incorrect answers for specific questions. From this data, Prompt-OIRL builds a prediction model, a kind of simplified scoring system, that can estimate the likelihood of a given prompt successfully answering a particular question. In the second stage, during actual use (inference), when a new question is asked, the method uses this learned prediction model to instantly evaluate a set of candidate prompts and select the one with the highest predicted chance of success. The key advantage is that this selection happens without having to send each candidate prompt to the large language model, which is a time-consuming and costly process. Only the single, selected prompt is then sent to the language model to generate the final answer. This approach directly tackles two major challenges identified in the paper: the difficulty of evaluating a prompt's quality when the correct answer is unknown (Challenge 1) and the high cost of repeatedly interacting with language models to test different prompts (Challenge 2). The principal evidence for its effectiveness is demonstrated in their experiments. For instance, when only one prompt's worth of data was available for training, Prompt-OIRL improved the success rate of getting correct answers by 24.3% compared to simply using the best prompt from that limited training data. This shows its ability to learn and generalize effectively, even from scarce information, to make better, query-specific prompt choices.

ParetoPrompt [21] is a reinforcement learning method designed to automatically generate effective text prompts by learning from comparisons between prompt pairs. Instead of calculating a single combined score to evaluate a prompt's quality, which would require making assumptions about how to balance different goals, the method compares two prompts based on whether one "dominates" the other across multiple objectives. A prompt dominates another if it is better in at least one objective (such as accuracy or fluency) and not worse in any other. The policy model, which generates the prompts, is then updated using a loss function that encourages it to produce more of the dominating prompts. A key innovation is the use of a special "non-dominated loss function" for pairs of prompts where neither dominates the other. This loss function encourages the model to generate prompts that represent different, equally valid trade-offs between objectives, rather than clustering around a single type of solution. This combination ensures that the final set of prompts generated by the model is not only high-quality but also diverse, effectively covering the entire range of optimal compromise solutions, known as the Pareto front. The principal evidence supporting this definition is found in the text style transfer experiment results shown in Table 2, where ParetoPrompt-IPO generated the largest set of

non-dominated prompts (Pareto Set Size of 34.8) and achieved the highest Hypervolume (HV) score of 2406, demonstrating its ability to produce both diverse and high-performing prompts that cover the full range of optimal trade-offs.

Algorithm 7 Pseudocode: Prompt Construction with RLHF Language Model

Require: Training dataset $\mathcal{D}_{\text{train}} = \{(q, \mathcal{P}_{\text{cand}}, y)\}$ containing questions, candidate prompts, and ground-truth answers, RLHF language model π_θ for prompt optimization, Target LLM $M_{\text{a-gen}} \in \mathcal{M}$ for generation, Number of optimization iterations T

Ensure: Optimized prompt selection policy π_θ^*

Stage 1: Policy Learning from Offline Data
for iteration $t = 1$ **to** T **do**
    Sample batch of training examples from $\mathcal{D}_{\text{train}}$
    For each example, evaluate candidate prompts $\mathcal{P}_{\text{cand}}$ using learned reward model
    Update policy π_θ to maximize expected reward via RLHF
end for
Obtain optimized policy π_θ^* for prompt selection
Stage 2: Inference with Learned Policy
for each test input q **do**
    Identify candidate prompts $\mathcal{P}_{\text{cand}}$ for input q
    Select optimal prompt $p^* = \arg\max_{p \in \mathcal{P}_{\text{cand}}} \pi_\theta^*(p \mid q)$
    Generate output via $M_{\text{a-gen}}$ using prompt p^*
end for
return Optimized policy π_θ^*

2.3 Methodology

2.3.1 Problem Definition

Mathematical definition generation requires the language model $M_{\text{a-gen}}$ to navigate from the holistic reasoning space $\mathfrak{R}$ toward a specific reasoning direction $\rho_{\mathcal{I},\mathcal{D}}$ that yields a definition satisfying standards of formal precision, conceptual completeness, and structural coherence. Existing prompt strategies approach this navigation through various linear reasoning patterns $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$, yet each exhibits distinct limitations in how they structure the reasoning trajectory. The direct prompt p_{direct} instructs the model to generate a definition in a single step without any intermediate reasoning scaffolding, effectively requiring the model to identify the optimal reasoning direction $\rho_{\mathcal{I},\mathcal{D}}$ from the outset without preparatory exploration. The excellent-professor-aware prompt $p_{\text{e-pro}}$ introduces persona-based contextualization but still relies on a one-pass generation process, assuming that the assigned persona alone sufficiently constrains the reasoning space without providing explicit guidance on how to traverse it. The step-by-step prompt p_{sbs} decomposes the reasoning process into sequential steps, yet these steps remain loosely structured and do not ensure that the model accumulates progressively refined understanding before committing to a final definition.

The central problem addressed by the Cumulative Thinking Prompt is whether a more structured linear reasoning approach, one that explicitly sequences knowledge accumulation prior to generation, can more effectively guide the language model toward

optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$. While existing linear methods either bypass intermediate reasoning altogether or employ decomposition without explicit knowledge scaffolding, they do not guarantee that the model has established a sufficient conceptual foundation before generating the final output. This gap is particularly consequential for mathematical definition generation, where the quality of the final definition depends critically on the model's ability to integrate foundational understanding with formal precision.

The Cumulative Thinking Prompt addresses this problem by introducing a structured linear reasoning framework that separates the reasoning process into two distinct phases: a preparatory knowledge accumulation phase followed by a definition generation phase. The method instructs the model to perform five sequential recalling steps $\{r_1, r_2, \ldots, r_5\}$ that progress from essential to abstract understanding of the target term $t_{(i,j)}$ within its subject s_i, thereby establishing a progressively refined conceptual foundation. Only after completing this explicit knowledge accumulation does the model proceed to generate the final definition $g_{(i,j)}^{(p_{\text{ct}})}$. This design treats reasoning navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D}) = \mathfrak{R}_{\mathcal{I},\mathcal{D}} \to \mathcal{R}_{\mathcal{I},\mathcal{D}} \to \rho_{\mathcal{I},\mathcal{D}}$ not as an implicit byproduct of generation but as a deliberate, staged process where each phase prepares the model for the next. By making the knowledge accumulation explicit and sequential, the Cumulative Thinking Prompt aims to ensure that the model's final reasoning direction is informed by a comprehensive preparatory exploration of the reasoning space, thereby overcoming the limitations of linear methods that either skip this preparatory phase entirely or structure it insufficiently.

2.3.2 Dataset

The dataset employed for evaluating the proposed Cumulative Thinking Prompt p_{ct} against the baseline prompt methods comprises 439 math terms T_{s_i} systematically distributed across 18 distinct mathematical subjects $S = \{s_1, s_2, \ldots, s_{18}\}$, including algebra, geometry, calculus, linear algebra, probability, topology, and others. Designed to represent the typical mathematical terms that college STEM students encounter when learning mathematics for machine learning and neural network fundamentals, each subject contains between 23 and 26 terminologies with a mean of approximately 24 terms per subject, ensuring balanced representation across foundational mathematical domains.

Table 2.2: Cumulative Thinking Prompt Dataset: Dataset

Subject	Number of Math Terms
Abstract Algebra	26
Algebra	24
Calculus	24
Differential Geometry	25
Discrete Mathematics	25
Dynamical Systems	23
Geometry	24
Graph Theory	26
Information Theory	24
Linear Algebra	25
Logic	24
Measure Theory	23
Numerical Linear Algebra	24
Optimization	26
Probability	24
Statistics	24
Topology	24
Trigonometry	24
Total	439

This dataset is specifically constructed to characterize the distinct reasoning behaviors of different prompt methods $\mathcal{P} = \{p_{\text{direct}}, p_{\text{c-pro}}, p_{\text{sbs}}, p_{\text{ct}}\}$ and to investigate whether the proposed Cumulative Thinking Prompt's sequential reasoning order, which first initializes five sequential and accumulative knowledge recall and thinking steps $\{r_1, r_2, \ldots, r_5\}$ within the sequential reasoning framework $\mathcal{I}$ before generating the final mathematical term definition $g_{(i,j)}^{(p_{\text{ct}})}$, can effectively guide the language model $M_{\text{a-gen}}$ toward identifying the optimal reasoning subspace $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ within the holistic reasoning space $\mathfrak{R}$, thereby producing superior definition generation outcomes compared to baseline prompt methods that employ alternative reasoning navigational strategies $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$.

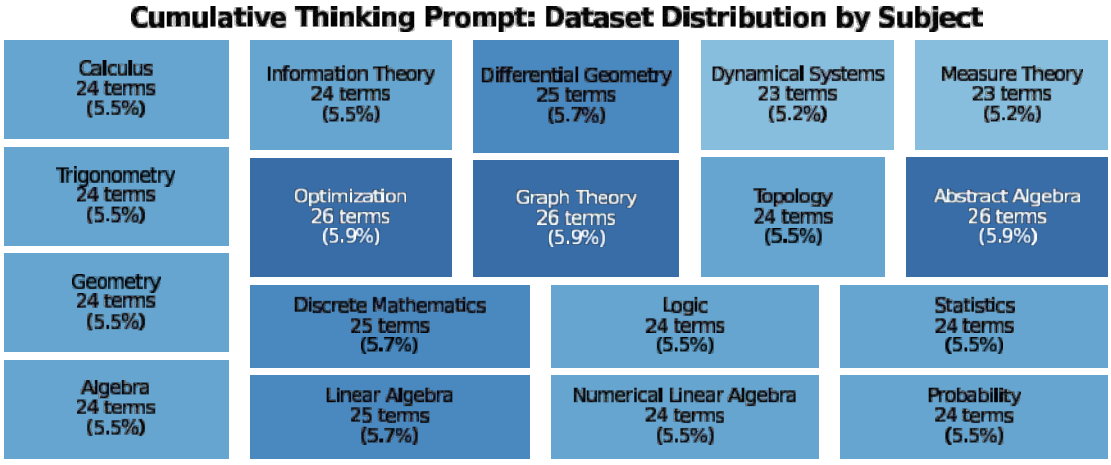


Figure 2.1: Cumulative Thinking Prompt: Dataset Distribution by Subject

The dataset construction followed a multi-agent collaborative optimization procedure, beginning with DeepSeek serving as the suggesting LLM agent $\mathcal{A}_{\text{suggest}}$ to propose an initial structure of mathematical subjects $S = \{s_1, s_2, \ldots, s_m\}$ and corresponding mathematical terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ for each subject. This initial structure was then iteratively reviewed and optimized through a collaborative framework involving Qwen, ChatGPT, and Gemini operating as evaluator agents $\mathcal{A}_{\text{eval}} \subseteq \mathcal{A}$ and optimizer agents $\mathcal{A}_{\text{optim}} \subseteq \mathcal{A}$, following the implicit LLM collaboration strategies $\mathcal{S}_{\text{collab}} = \{s_1, s_2, \ldots, s_L\}$ that enabled sequential consensus and parallel deliberation. The iterative refinement process continued until the optimized dataset structure, encompassing both the mathematical subjects and the mathematical terms within each subject, achieved mutual agreement across all participating language model agents, ensuring that the final dataset represented a collectively validated foundation for evaluating reasoning behaviors across different prompt methods $\mathcal{P}$.

2.3.3 Methods

Proposed Prompt Method

The proposed Cumulative Thinking Prompt p_{ct} is a sequential reasoning method designed for mathematical term definition generation, where the language model $M_{\text{a-gen}}$ first initializes five sequential and accumulative knowledge recall steps $\{r_1, r_2, \ldots, r_5\}$, progressing from essential to abstract understanding of the target term $t_{(i,j)}$ within its subject s_i, before generating the final definition $g_{(i,j)}^{(p_{\text{ct}})}$. These five recalling steps serve as implicit reasoning context and reasoning navigation constraints that guide the model toward relevant reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and optimal reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$, enabling the accumulation of pertinent knowledge that facilitates convergence to the most appropriate mathematical definition. As one of two proposed prompt methods that employ sequential reasoning as their fundamental reasoning behavior for the mathematical term definition generation task, p_{ct} leverages this structured pre-generation thinking process to navigate the holistic reasoning space $\mathfrak{R}$ more effectively than non-sequential or less constrained alternatives.

Cumulative Thinking Prompt

recall your understanding about [term] in [subject] from essential to abstract in sequential 5 times, then to generate definition of [term] in [subject] with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format, based on all 5 previous sequential thinking before generating the mathematical definition.

Algorithm 8 Pseudocode: Cumulative Thinking Prompt

Require: Math subjects $S = \{s_1, s_2, \ldots, s_m\}$, Math terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ for each subject, Cumulative Thinking Prompt p_{ct}, Number of sequential recalls N, Generation LLM $M_{\text{a-gen}} \in \mathcal{M}$ accessed via API, Evaluation LLM $M_{\text{a-eval}} \in \mathcal{M}$ accessed via API, Eleven evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$

Ensure: Evaluation results $\mathcal{R}$

for each subject $s_i \in S$ **do**
  for each term $t_{(i,j)} \in T_{s_i}$ **do**
    for sequential recall $k = 1$ **to** N **do**
      Generate recall r_k from essential to abstract understanding using p_{ct} via $M_{\text{a-gen}}$
    end for
    Generate final definition $g_{(i,j)}^{(p_{\text{ct}})}$ via $M_{\text{a-gen}}$, based on all sequential recalls $\{r_1, r_2, \ldots, r_N\}$
    for each evaluation metric $e_k \in \mathcal{E}$ **do**
      Evaluate generated final definition $g_{(i,j)}^{(p_{\text{ct}})}$ using metric e_k via $M_{\text{a-eval}}$
      Store evaluation score $r_{(i,j,p_{\text{ct}},k)}$
    end for
  end for
end for
Compute aggregate scores for Cumulative Thinking Prompt
return Evaluation results $\mathcal{R}$ for cumulative thinking method

Baseline Prompt Methods

To establish a comprehensive performance comparison with the proposed Cumulative Thinking Prompt p_{ct}, three baseline prompt methods are employed for the same mathematical term definition generation task: direct prompt p_{direct}, excellent-professor-aware prompt $p_{\text{e-pro}}$, and step-by-step prompt p_{sbs}. Each of these baseline methods represents widely used and well-established prompt techniques that have robustly demonstrated the advantages and strengths of their respective reasoning patterns for reasoning navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ to achieve strong generation performance. The direct prompt p_{direct} relies on straightforward, single-step generation without intermediate reasoning; the excellent-professor-aware prompt $p_{\text{e-pro}}$ leverages persona-driven contextualization to guide the model toward authoritative outputs; and the step-by-step prompt p_{sbs} employs sequential reasoning decomposition to systematically build toward the final definition. These baselines collectively provide a rigorous benchmark against which the sequential recalling behavior of the proposed Cumulative Thinking Prompt can be evaluated.

Three Baseline Prompt Methods

Direct Prompt:
definition of [term] in [subject] with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

Excellent-Professor-Aware Prompt:
you are an excellent math professor who are good at math teaching, generate definition of [term] in [subject] with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

Step-by-Step Prompt:
let's think step by step, definition of [term] in [subject] with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

Algorithm 9 Pseudocode: Direct Prompt

Require: Math subjects $S = \{s_1, s_2, \ldots, s_m\}$, Math terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ for each subject, Direct prompt p_{direct}, Generation LLM $M_{\text{a-gen}} \in \mathcal{M}$ accessed via API, Evaluation LLM $M_{\text{a-eval}} \in \mathcal{M}$ accessed via API, Eleven evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$

Ensure: Evaluation results $\mathcal{R}$

for each subject $s_i \in S$ **do**
  for each term $t_{(i,j)} \in T_{s_i}$ **do**
    Generate final definition $g_{(i,j)}^{(p_{\text{direct}})}$ via $M_{\text{a-gen}}$ using p_{direct}
    for each evaluation metric $e_k \in \mathcal{E}$ **do**
      Evaluate generated final definition $g_{(i,j)}^{(p_{\text{direct}})}$ using metric e_k via $M_{\text{a-eval}}$
      Store evaluation score $r_{(i,j,p_{\text{direct}},k)}$
    end for
  end for
end for
Compute aggregate scores for direct prompt
return Evaluation results $\mathcal{R}$ for direct method

Algorithm 10 Pseudocode: Excellent-Professor-Aware Prompt

Require: Math subjects $S = \{s_1, s_2, \ldots, s_m\}$, Math terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ for each subject, Excellent-professor-aware prompt $p_{\text{e-pro}}$, Generation LLM $M_{\text{a-gen}} \in \mathcal{M}$ accessed via API, Evaluation LLM $M_{\text{a-eval}} \in \mathcal{M}$ accessed via API, Eleven evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$

Ensure: Evaluation results $\mathcal{R}$

for each subject $s_i \in S$ **do**
  for each term $t_{(i,j)} \in T_{s_i}$ **do**
    Adopt excellent math professor persona via $p_{\text{e-pro}}$
    Generate final definition $g_{(i,j)}^{(p_{\text{e-pro}})}$ via $M_{\text{a-gen}}$
    for each evaluation metric $e_k \in \mathcal{E}$ **do**
      Evaluate generated final definition $g_{(i,j)}^{(p_{\text{e-pro}})}$ using metric e_k via $M_{\text{a-eval}}$
      Store evaluation score $r_{(i,j,p_{\text{e-pro}},k)}$
    end for
  end for
end for
Compute aggregate scores for excellent-professor-aware prompt
return Evaluation results $\mathcal{R}$ for excellent-professor-aware method

Algorithm 11 Pseudocode: Step-by-Step Prompt

Require: Math subjects $S = \{s_1, s_2, \ldots, s_m\}$, Math terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ for each subject, Step-by-step prompt p_{sbs}, Generation LLM $M_{\text{a-gen}} \in \mathcal{M}$ accessed via API, Evaluation LLM $M_{\text{a-eval}} \in \mathcal{M}$ accessed via API, Eleven evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$

Ensure: Evaluation results $\mathcal{R}$

for each subject $s_i \in S$ **do**
  for each term $t_{(i,j)} \in T_{s_i}$ **do**
    for reasoning step $k = 1$ **to** K **do**
      Generate step k reasoning using p_{sbs} via $M_{\text{a-gen}}$
    end for
    Generate final definition $g_{(i,j)}^{(p_{\text{sbs}})}$ via $M_{\text{a-gen}}$
    for each evaluation metric $e_k \in \mathcal{E}$ **do**
      Evaluate generated final definition $g_{(i,j)}^{(p_{\text{sbs}})}$ using metric e_k via $M_{\text{a-eval}}$
      Store evaluation score $r_{(i,j,p_{\text{sbs}},k)}$
    end for
  end for
end for
Compute aggregate scores for step-by-step prompt
return Evaluation results $\mathcal{R}$ for step-by-step method

2.3.4 Experiment Setup

Applied Large Language Models

Table 2.3: Large Language Models for All Three Studies

Large Language Model	Versions	Functions
DeepSeek	V 3.2	Used for math term definition generation and evaluation, and also for dataset and evaluation metrics construction.
Qwen	3.5 Plus	Used for optimizing the dataset and evaluation metrics.
ChatGPT	5.3 and 5.4 Instant	Used for optimizing the dataset and evaluation metrics.
Gemini	3.1 Flash	Used for optimizing the dataset and evaluation metrics.

Hyperparameters

Table 2.4: Cumulative Thinking Prompt: Hyperparameters

Hyperparameter	Value	Explanation
Max Retries	3	The system tries up to 3 times if an API call fails, balancing reliability against total processing time.
Retry Delay	2 seconds	The system waits 2 seconds between retry attempts to give servers time to recover and prevent overwhelming them.
API Keys Count	User input	Multiple API keys distribute requests across accounts to increase throughput and avoid hitting individual rate limits.
Model	deepseek-chat	DeepSeek v3.2, Used for definition generation and quality evaluation.
Temperature for Inference	0.25	Controls randomness in response generation; moderate temperature balances creativity with coherence for generating reasoning steps.
Temperature for Evaluation	0.1	A low temperature ensures consistent, deterministic evaluations by minimizing randomness in the AI's responses.
Max_Tokens	4000	This limits the AI's response length to ensure complete evaluation results fit within the allowed capacity.

2.3.5 Evaluation Metrics

The evaluation of mathematical term definition quality for both the proposed Cumulative Thinking Prompt p_{ct} and the baseline prompt methods p_{direct}, $p_{\text{c-pro}}$, and p_{sbs} was conducted using eleven automatic evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$. These metrics were implemented via the DeepSeek API, where the evaluation LLM $M_{\text{a-eval}} \in \mathcal{M}$ systematically assessed each generated definition $g_{(i,j)}^{(p)}$ across multiple dimensions of definition quality, producing evaluation scores $r_{(i,j,p,k)}$ for each metric e_k. This automated evaluation framework ensured consistent, scalable, and reproducible assessment across all generated definitions, enabling rigorous comparative analysis of reasoning navigation strategies employed by different prompt methods.

This collection of evaluation metrics was constructed through a multi-agent collaborative framework that parallels the dataset optimization procedure. Initially, DeepSeek served

as the suggesting agent $\mathcal{A}_{\text{suggest}}$ to propose a series of mathematical term definition evaluation metrics, each accompanied by detailed metric descriptions designed to enable the evaluator LLM $M_{\text{a-eval}}$ to clearly understand the meaning and scope of each evaluation criterion. Following this initial suggestion, Qwen, ChatGPT, and Gemini were deployed as evaluator agents $\mathcal{A}_{\text{eval}}$ and optimizer agents $\mathcal{A}_{\text{optim}}$ to collaboratively review and refine the proposed metrics from both holistic and specific perspectives. This iterative optimization process ensured that the evaluation metrics were rigorously aligned with the fundamental knowledge and skills expected of college STEM students learning mathematics for machine learning and neural network fundamentals, consistent with the design principles established during dataset construction.

A distinctive feature of the evaluation metrics, alongside the generation and evaluation dataset, is that both research components were proposed, reviewed, and optimized through the collaborative interaction of multiple LLM agents $\mathcal{A} = \{A_1, A_2, \ldots, A_K\}$, enabling iterative convergence toward improved quality in an automated manner. This agent-based collaboration strategy $\mathcal{S}_{\text{collab}}$ leveraged the powerful parametric knowledge and generation capabilities of proven large language models, allowing the evaluation framework to evolve through sequential consensus and parallel deliberation without requiring manual intervention. The resulting eleven evaluation metrics encompass a comprehensive set of dimensions for assessing mathematical definition quality. Collectively, these metrics provide a robust foundation for evaluating the reasoning navigation behaviors $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ exhibited by both the proposed Cumulative Thinking Prompt p_{ct} and the baseline prompt methods.

Table 2.5: Cumulative Thinking Prompt: Evaluation Framework

Evaluation Metrics	Evaluation Focus
Abstraction Level (AL)	Degree to which the explanation is formulated in terms of formal structures, properties, or logical relations rather than anchored to specific examples, visualizations, or intuitive analogies
Conceptual Completeness (CC)	Extent to which the definition includes all necessary and sufficient conditions, and omits no core attributes required to characterize the concept uniquely within its mathematical theory
Explanatory Hierarchy (EH)	Clarity and effectiveness of the logical ordering, from the initial definition to the statement of immediate implications, properties, or necessary consequences
Formal Mathematical Precision (FMP)	Degree of exactness in the formulation of mathematical statements, including unambiguous quantification, well-formed logical expressions, and precise specification of relations and operations
Definition Rigor (DR)	Completeness and explicitness of the scope, boundary conditions, domain restrictions, existence assertions, and uniqueness conditions (where applicable) accompanying a formal definition
Theoretical Correctness (TC)	Absence of logical errors, false implications, invalid equivalences, or statements that contradict established theorems or axioms within the relevant mathematical framework
Structural Coherence (SC)	Consistency and logical flow at macro-level (paragraph, section) and micro-level (sentence, clause), including non-contradiction, clear referents, and absence of abrupt conceptual jumps
Notation and Terminology Consistency (NTC)	Uniformity in the use of symbols, variable names, functional notation, and terminological choices throughout the definition, with no synonymous substitutions that imply distinction or conflicting usage
Use of Standard Academic Conventions (USAC)	Adherence to prevailing notational systems, symbolic idioms, and terminological norms accepted within the relevant mathematical subdiscipline, including conventional quantifier usage and symbol deployment
Academic Tone and Style (ATS)	Appropriateness of register, including impersonal construction, lexical formality, absence of colloquialism or rhetorical flourish, and maintenance of neutral, expository voice consistent with scholarly mathematical writing
Suitability for Textbook / Graduate Notes (STGN)	Fitness for direct inclusion in advanced expository prose without requiring immediate remediation, supplementation, or rephrasing; assumes graduate-level readiness but does not presuppose specialized prerequisite knowledge beyond the intended scope of the text

2.4 Results and Analytics

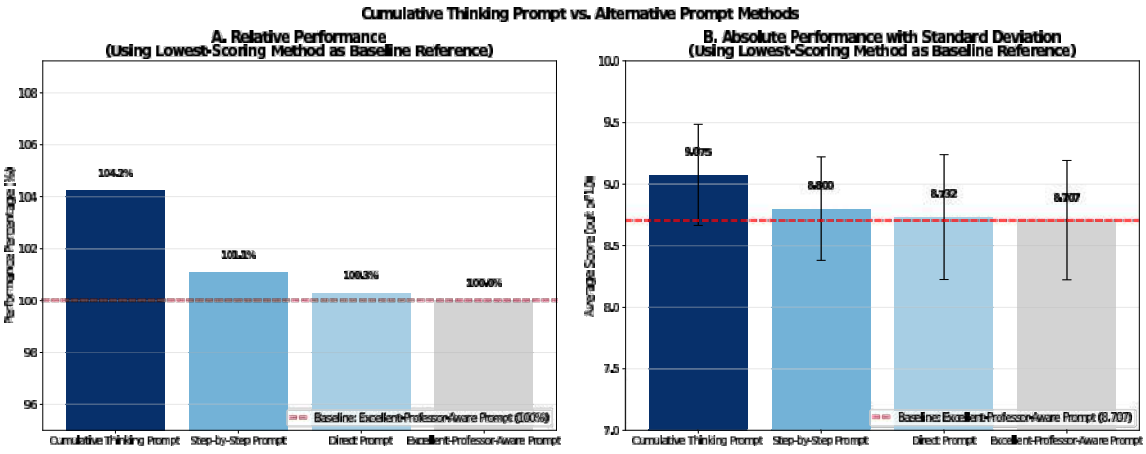


Figure 2.2: Cumulative Thinking Prompt: Holistic Relative and Absolute Performances

2.4.1 Absolute Performance Analytics

The Cumulative Thinking Prompt p_{ct} demonstrates superior performance across the vast majority of subjects, achieving the highest average scores in all of 18 math subjects. This method excels particularly in Differential Geometry (9.2727), Probability (9.2443), and Measure Theory (9.2431), where the sequential five-step recalling process $\{r_1, r_2, \ldots, r_5\}$ appears most effective for navigating complex, abstract mathematical domains. The method's strongest relative advantage over baselines is observed in Algebra (8.9261) and Trigonometry (9.0170), suggesting that the cumulative knowledge accumulation prior to definition generation $g_{(i,j)}^{(p_{\text{ct}})}$ is especially beneficial for foundational mathematical subjects. The lowest performance within this method occurs in Numerical Linear Algebra (8.9716), though this still ranks competitively against baseline methods.

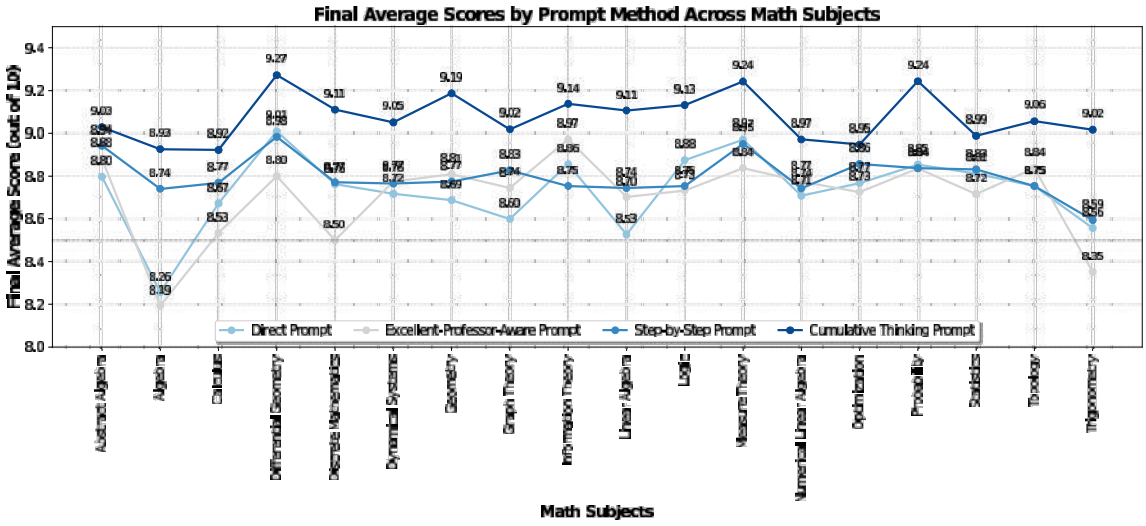


Figure 2.3: Cumulative Thinking Prompt: Subject-Level Absolute Performance (All Variations)

The step-by-step prompt p_{sbs} serves as the second strongest performer overall, consistently ranking above the direct prompt and excellent-professor-aware prompt across nearly all subjects. This method achieves its highest scores in Abstract Algebra (8.9423) and Measure Theory (8.9506), where sequential reasoning decomposition $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ aligns well with the hierarchical structure of these subjects. The method demonstrates particular strength in Differential Geometry (8.9836), closely approaching the performance of the proposed method. The step-by-step approach shows its most modest performance in Trigonometry (8.5947), suggesting that subjects with strong visual-spatial components may benefit less from purely sequential verbal reasoning.

The direct prompt p_{direct} exhibits consistent but moderate performance across the subject spectrum, functioning as a stable baseline without the additional reasoning constraints of other methods. This method achieves its highest scores in Measure Theory (8.9704) and Abstract Algebra (8.7972), where direct generation without intermediate

reasoning appears sufficient for producing acceptable definitions. The lowest performance occurs in Algebra (8.2557) and Numerical Linear Algebra (8.7083), indicating that subjects requiring precise formal notation and structural clarity benefit substantially from additional reasoning scaffolding.

The excellent-professor-aware prompt $p_{\text{e-pro}}$ shows the most variable performance across subjects, ranking lowest among all four methods in the majority of subject areas. This method achieves its highest scores in Information Theory (8.9716) and Abstract Algebra (8.8811), where the "excellent professor" persona may provide beneficial contextualization. However, it performs notably lower in Algebra (8.1932) and Trigonometry (8.3523), suggesting that persona assignment alone, without explicit reasoning guidance, is insufficient for consistently eliciting high-quality definitions. The variability in performance indicates that persona-based prompt $\mathcal{P}_{\text{profs}}$ may interact unpredictably with subject-specific content characteristics.

Table 2.6: Cumulative Thinking Prompt: Holistic Absolute Performance (All Variations)

Prompt Method	**Holistic Absolute Performance**	**Rank**	**Performance Characteristic**
Cumulative Thinking Prompt	9.075	1	Consistently highest across 16 subjects; strongest in abstract and measure-theoretic domains
Step-by-Step Prompt	8.800	2	Stable second-place performance; excels in algebraically structured subjects
Direct Prompt	8.732	3	Moderate, consistent baseline; weakest in subjects requiring precise formal structure
Excellent-Professor-Aware Prompt	8.707	4	Most variable; underperforms in foundational subjects despite persona assignment

The analytics indicate that the Cumulative Thinking Prompt's sequential recalling mechanism $\{r_1, r_2, \ldots, r_5\}$ effectively guides the language model $M_{\text{a-gen}}$ toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$, consistently outperforming baseline methods that employ either no reasoning scaffolding, linear decomposition only, or persona-based contextualization without structured pre-generation thinking.

2.4.2 Relative Performance Analytics

To enable systematic comparison across prompt methods, the excellent-professor-aware prompt $p_{\text{e-pro}}$ is selected as the baseline reference point, as it exhibits the lowest overall performance among all evaluated methods with a holistic average score of 8.707, ranking fourth across the 18 mathematical subjects. This method serves as the baseline against which the performance improvements of other prompt methods, direct prompt p_{direct}, step-by-step prompt p_{sbs}, and the proposed Cumulative Thinking Prompt p_{ct}, are measured. Each method's performance for a given subject is expressed as a percentage relative to

the baseline, where values exceeding 100% indicate performance gains over the lowest-performing method, providing a clear measure of improvement achieved through different reasoning navigation strategies $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$.

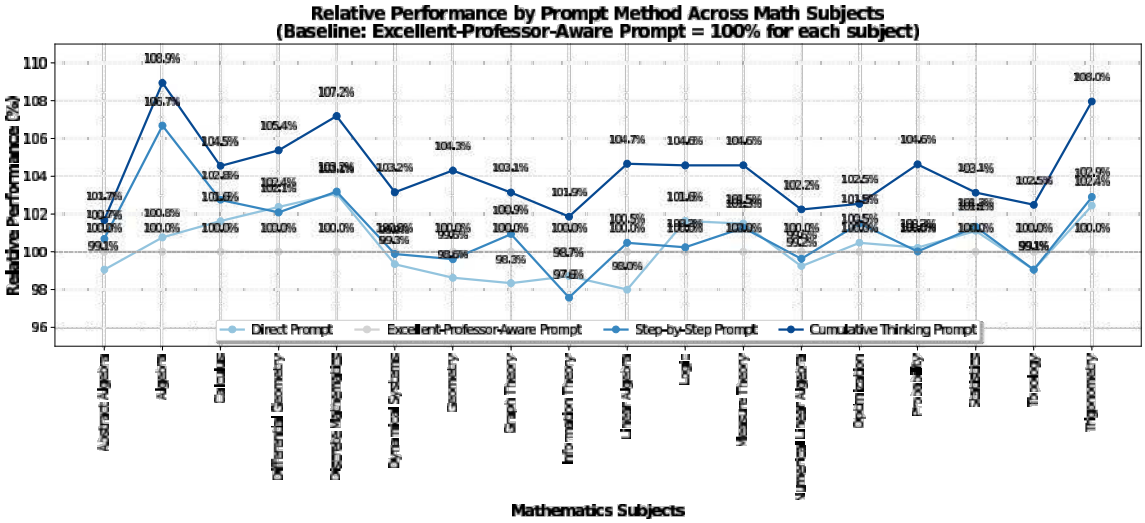


Figure 2.4: Cumulative Thinking Prompt: Subject-Level Relative Performance (All Variations)

The Cumulative Thinking Prompt p_{ct} achieves substantial and consistent performance gains over both the baseline excellent-professor-aware prompt and the alternative prompt methods across all 18 mathematical subjects. Relative to the baseline, improvements range from 1.67% in Abstract Algebra to 8.95% in Algebra, with an average holistic gain of 4.11%. When compared against the step-by-step prompt, the second strongest performer, p_{ct} demonstrates superior performance in 17 out of 18 subjects, with the most pronounced advantages in Discrete Mathematics (107.19% vs. 103.19%), Geometry (104.30% vs. 99.61%), and Algebra (108.95% vs. 106.68%). Against the direct prompt, p_{ct} outperforms in all 18 subjects, with average gains of 3.80%. The most significant advantages over alternative methods are observed in Algebra, Trigonometry (107.96%), and Probability (104.63%), where the sequential five-step recalling process $\{r_1, r_2, \ldots, r_5\}$ most effectively guides the language model toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$.

The step-by-step prompt p_{sbs} ranks as the second strongest performer overall, achieving positive gains over the baseline in 13 of 18 subjects with an average relative score of 101.05%. This method demonstrates its strongest advantage over the baseline in Algebra (106.68%), Discrete Mathematics (103.19%), and Trigonometry (102.90%), where sequential reasoning decomposition $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ effectively structures the definition generation process. However, p_{sbs} underperforms relative to the proposed Cumulative Thinking Prompt in all but one subject, with the largest performance gap observed in Geometry (4.69% lower) and Algebra (2.27% lower), indicating that linear reasoning alone, without

cumulative knowledge accumulation, yields suboptimal navigation of the reasoning space.

The direct prompt p_{direct} ranks third overall with an average relative score of 100.31%, demonstrating selective gains over the baseline in 11 of 18 subjects. While this method surpasses the baseline in subjects such as Discrete Mathematics (103.08%) and Trigonometry (102.45%), it consistently underperforms relative to the Cumulative Thinking Prompt across all 18 subjects, with an average deficit of 3.80%. The most substantial gaps are observed in Algebra (8.19% lower), Probability (4.42% lower), and Linear Algebra (6.67% lower), confirming that the absence of reasoning scaffolding constrains the model's ability to navigate toward optimal reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$.

Table 2.7: Cumulative Thinking Prompt: Relative Absolute Performance (All Variations)

Prompt Method	Average Relative Score	Subjects Above Baseline	Subjects Below Baseline
Cumulative Thinking Prompt	**104.11%**	18 of 18	0 of 18
Step-by-Step Prompt	101.05%	13 of 18	5 of 18
Direct Prompt	100.31%	11 of 18	7 of 18

The Cumulative Thinking Prompt consistently achieves superior performance over both the baseline excellent-professor-aware prompt and the alternative prompt methods across all mathematical subjects, with an average relative score of 104.11%. The sequential recalling mechanism $\{r_1, r_2, \ldots, r_5\}$ enables the language model $M_{\text{a-gen}}$ to accumulate relevant knowledge prior to definition generation, effectively navigating from the holistic reasoning space $\mathfrak{R}$ toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$. This structured pre-generation thinking provides a distinct reasoning advantage over the linear decomposition of step-by-step prompt, the persona-based contextualization of excellent-professor-aware prompt, and the direct generation approach, confirming that cumulative knowledge accumulation before final output yields optimal mathematical definition generation quality.

2.4.3 Automatic Evaluation Reliability Analytics

To establish the reliability of automatic evaluation using the DeepSeek API as the evaluator $M_{\text{a-eval}} \in \mathcal{M}$ for mathematical term definition quality scoring, a validation study was conducted. A random sample comprising 25% of all math terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ was selected, with sampling stratified to ensure even representation across all 18 mathematical subjects $S = \{s_1, s_2, \ldots, s_{18}\}$. For this sampled subset, the same evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$ were applied through manual evaluation by the dissertation author as human assessor to independently score the generated definitions $g^{(p)}_{(i,j)}$ across all prompt methods $\mathcal{P} = \{p_{\text{direct}}, p_{\text{e-pro}}, p_{\text{sbs}}, p_{\text{ct}}\}$.

The rankings obtained from manual evaluation for each prompt method were compared

against the rankings derived from automatic evaluation via DeepSeek. The comparative analytics obtained a 90% consistency between the automated and manual evaluation approaches, demonstrating that the automatic evaluation framework reliably replicates human judgment across diverse mathematical domains.

The strong alignment between automatic and manual evaluation validates that large language models such as DeepSeek, with their vast learned parametric knowledge and powerful capability in navigating their reasoning space $\mathfrak{R}$, can effectively identify optimal evaluation subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and evaluation directions $\rho_{\mathcal{I},\mathcal{D}}$ for complex contextual quality assessment tasks. This finding substantiates that automatic evaluation represents a reliable, scalable, and reproducible method for implementing advanced context quality evaluation, such as the mathematical term definition evaluation conducted in this study, without compromising the validity and consistency of the assessment outcomes.

2.4.4 Concrete Examples

Definition of Singular Value Decomposition in Linear Algebra with Cumulative Thinking Prompt

Recall your understanding about singular value decomposition in linear algebra from essential to abstract in sequential 5 times, then to generate definition of singular value decomposition in linear algebra with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format, based on all 5 previous sequential thinking before generating the mathematical definition.

first thinking: Recall that any real or complex matrix $\mathbf{A}$ of dimensions $m \times n$ can be decomposed into three simpler matrices: two orthogonal/unitary matrices and one diagonal matrix containing non-negative real numbers.
second thinking: The decomposition is $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$, where $\mathbf{U}$ is an $m \times m$ orthogonal matrix (if real) or unitary matrix (if complex), $\mathbf{V}$ is an $n \times n$ orthogonal/unitary matrix, and $\mathbf{\Sigma}$ is an $m \times n$ rectangular diagonal matrix with non-negative real entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ on its main diagonal, where $r = \text{rank}(\mathbf{A})$.
third thinking: The columns of $\mathbf{U}$ are called left-singular vectors and form an orthonormal basis for the column space of $\mathbf{A}$. The columns of $\mathbf{V}$ are called right-singular vectors and form an orthonormal basis for the row space of $\mathbf{A}$. The diagonal entries σ_i are the singular values, representing the scaling factors along orthogonal directions.
fourth thinking: SVD connects to fundamental matrix properties: the singular values are the square roots of the eigenvalues of $\mathbf{A}^*\mathbf{A}$ or $\mathbf{A}\mathbf{A}^*$, the columns of $\mathbf{V}$ are eigenvectors of $\mathbf{A}^*\mathbf{A}$, and the columns of $\mathbf{U}$ are eigenvectors of $\mathbf{A}\mathbf{A}^*$. This reveals the geometric action of $\mathbf{A}$: it rotates/reflects by $\mathbf{V}^*$, scales by $\mathbf{\Sigma}$, then rotates/reflects by $\mathbf{U}$.
fifth thinking: Abstractly, SVD provides optimal low-rank approximations via the Eckart–Young theorem: the best rank-k approximation $\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^*$, where $\mathbf{U}_k$ and $\mathbf{V}_k$ contain the first k columns, and $\mathbf{\Sigma}_k$ the top k singular values. This minimizes $\|\mathbf{A} - \mathbf{A}_k\|$ under spectral/Frobenius norms, making SVD fundamental for dimensionality reduction, data compression, and solving ill-conditioned systems.
definition: In linear algebra, the singular value decomposition (SVD) of a real or complex matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ is a factorization of the form $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$, where $\mathbf{U} \in \mathbb{C}^{m\times m}$ and $\mathbf{V} \in \mathbb{C}^{n\times n}$ are unitary matrices (orthogonal if $\mathbf{A}$ is real), and $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is a rectangular diagonal matrix with non-negative real entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ on its main diagonal, where $r = \text{rank}(\mathbf{A})$. The columns $\mathbf{u}_i$ of $\mathbf{U}$ are the left-singular vectors of $\mathbf{A}$, forming an orthonormal basis for the column space; the columns $\mathbf{v}_i$ of $\mathbf{V}$ are the right-singular vectors, forming an orthonormal basis for the row space. The singular values $\sigma_i = \sqrt{\lambda_i(\mathbf{A}^*\mathbf{A})}$ are the square roots of the eigenvalues of $\mathbf{A}^*\mathbf{A}$, revealing the geometric transformation: $\mathbf{A}$ first applies an orthogonal transformation $\mathbf{V}^*$, then scales each coordinate by σ_i, and finally applies another orthogonal transformation $\mathbf{U}$. This decomposition provides the optimal low-rank approximation via the Eckart–Young theorem: $\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^*$ minimizes $\|\mathbf{A} - \mathbf{B}\|$ for all rank-k matrices $\mathbf{B}$ under the spectral or Frobenius norm.

Definition of Stochastic Gradient Descent in Optimization with Cumulative Thinking Prompt

Recall your understanding about stochastic gradient descent in optimization from essential to abstract in sequential 5 times, then to generate definition of stochastic gradient descent in optimization with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format, based on all 5 previous sequential thinking before generating the mathematical definition.

first thinking: Stochastic Gradient Descent (SGD) is a fundamental optimization algorithm used to minimize a loss function, particularly in machine learning. At its core, it's an iterative method for finding the parameters (like weights in a neural network) that result in the lowest possible error. Unlike standard gradient descent, which computes the gradient using the entire dataset, SGD uses a single, randomly selected data point (or a small mini-batch) to estimate the gradient at each step. This makes each update much faster but also much noisier.
second thinking: The essential mechanism involves initializing parameters randomly, then repeatedly: 1) sampling a data instance, 2) computing the gradient of the loss for that single instance with respect to the parameters, and 3) updating the parameters by moving a small step in the opposite direction of that gradient. The step size is controlled by the learning rate. The 'stochastic' (random) nature of the sample selection introduces variance into the gradient estimates, which can help escape shallow local minima but requires careful tuning of the learning rate to ensure convergence.
third thinking: Abstracting further, SGD solves the empirical risk minimization problem: we aim to find parameters θ that minimize the average loss $L(\theta) = \frac{1}{N}\sum_{i=1}^{N} l_i(\theta)$, where l_i is the loss for the i-th data point. The true gradient is $\nabla L(\theta) = \frac{1}{N}\sum \nabla l_i(\theta)$. SGD approximates this full-batch gradient with an unbiased estimator $g_t = \nabla l_{i_t}(\theta_t)$, where i_t is randomly chosen. The update rule is $\theta_{t+1} = \theta_t - \eta_t g_t$, with η_t as the learning rate schedule. This trade-off between computational efficiency and noisy updates is central to its analysis.
fourth thinking: From a higher-level perspective, SGD is a stochastic approximation algorithm for finding the zeros of a function's gradient. It belongs to the family of first-order iterative optimization methods. The noise in the gradient estimate acts as a form of regularization, which can lead to better generalization in machine learning models compared to full-batch gradient descent. Its convergence properties depend critically on conditions like a decreasing learning rate (e.g., $\eta_t \propto \frac{1}{t}$) and the convexity (or smoothness) of the objective function.
fifth thinking: At the most abstract level, SGD is a manifestation of the Robbins-Monro stochastic approximation algorithm applied to optimization. It operates in a high-dimensional parameter space under uncertainty, using noisy, incomplete information (a single sample gradient) to navigate towards an optimum. This embodies a fundamental trade-off in large-scale learning: precise, expensive computations versus cheap, approximate ones. The algorithm's success hinges on the mathematical properties of the loss landscape and the annealing of the learning rate to average out the noise over time, ensuring convergence to a stationary point.
definition: Stochastic Gradient Descent (SGD) is a first-order iterative optimization algorithm designed to minimize an objective function $J(\theta)$ expressed as a sum of differentiable functions, $J(\theta) = \frac{1}{n}\sum_{i=1}^{n} L_i(\theta)$, where θ are the parameters and L_i is the loss for the i-th data point. Instead of computing the exact gradient $\nabla J(\theta) = \frac{1}{n}\sum_{i=1}^{n} \nabla L_i(\theta)$, which is costly for large n, SGD uses a stochastic approximation. At each iteration t, it uniformly samples an index i_t (or a mini-batch of indices) and computes the noisy gradient estimate $g_t = \nabla L_{i_t}(\theta_t)$. The parameters are then updated as $\theta_{t+1} = \theta_t - \eta_t g_t$, where $\eta_t > 0$ is the learning rate. This update moves θ in the direction opposite to the instantaneous gradient estimate. The unbiasedness $\mathbb{E}[g_t] = \nabla J(\theta_t)$ ensures that, on average, the process follows the true gradient descent direction. For convergence to a local minimum, the learning rate must satisfy the Robbins-Monro conditions $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$, allowing the noise to be averaged out over time while ensuring sufficient step sizes to reach the optimum.

2.5 Conclusion

2.5.1 Research Contributions

This study introduces the Cumulative Thinking Prompt p_{ct} as a novel contribution to prompt engineering, distinguished by its structured pre-generation knowledge accumulation mechanism. Unlike conventional prompt approaches that either generate outputs directly, decompose reasoning linearly, or rely on persona-based contextualization, the proposed method operationalizes a deliberate preparatory phase consisting of five sequential recalling

steps $\{r_1, r_2, \ldots, r_5\}$ that progress from essential to abstract understanding. This design reframes the relationship between reasoning navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ and solution generation by treating knowledge accumulation as an explicit cognitive scaffold rather than an implicit byproduct of generation. The empirical validation demonstrates that this approach consistently outperforms three widely used baseline methods across 18 mathematical subjects, achieving an average relative score of 104.11% against the lowest-performing baseline and securing the highest ranking in all of 18 math subjects. The theoretical grounding of this mechanism within the reasoning space framework, where the model progressively traverses from the holistic reasoning space $\mathfrak{R}$ toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and specific reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$, offers a conceptual lens for understanding why sequential knowledge accumulation yields superior performance compared to alternative prompt strategies that lack such preparatory structure.

A second contribution lies in the development and rigorous validation of a comprehensive dataset for mathematical definition generation, encompassing 439 terminologies T_{s_i} distributed across 18 subjects $S = \{s_1, s_2, \ldots, s_{18}\}$. The construction of this resource employed a multi-agent collaborative methodology wherein DeepSeek served as the suggesting agent $\mathcal{A}_{\text{suggest}}$ to propose initial math subject structures with corresponding math terms, while Qwen, ChatGPT, and Gemini acted as evaluator agents $\mathcal{A}_{\text{eval}} \subseteq \mathcal{A}$ and optimizer agents $\mathcal{A}_{\text{optim}} \subseteq \mathcal{A}$ to iteratively review and refine the dataset through sequential review until the final consensus has been reached. This agent-based collaboration strategy $\mathcal{S}_{\text{collab}}$ ensured that the final dataset achieved mutual agreement across all participating language models, establishing a foundation that is both representative of college STEM mathematical learning objectives and methodologically reproducible. The dataset's design intentionally captures the typical mathematical terminology encountered by students pursuing machine learning and neural network fundamentals, thereby aligning the evaluation framework with authentic educational contexts while enabling systematic comparison of reasoning behaviors across different prompt methods $\mathcal{P} = \{p_{\text{direct}}, p_{\text{e-pro}}, p_{\text{sbs}}, p_{\text{ct}}\}$.

The third contribution addresses the critical challenge of evaluation methodology through the construction and validation of an automatic evaluation framework comprising eleven metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$. Following the same multi-agent collaborative paradigm applied to dataset construction, DeepSeek initially proposed a series of evaluation metrics with detailed descriptions to enable the evaluator LLM $M_{\text{a-eval}}$ to clearly understand the scope and meaning of each criterion. Qwen, ChatGPT, and Gemini subsequently served as evaluator and optimizer agents to collaboratively review and refine the metrics from both holistic and specific perspectives, ensuring alignment with the fundamental knowledge and skills expected of college STEM students. The resulting evaluation framework encompasses dimensions ranging from Abstraction Level (AL) and Conceptual Completeness (CC) to Formal Mathematical Precision (FMP) and Suitability

for Textbook / Graduate Notes (STGN). The reliability of this automatic evaluation was rigorously validated through a comparative study wherein a randomly selected 25% sample of generated definitions was manually scored by a human assessor, yielding a 90% consistency rates with the LLM. This validation substantiates that large language model with extensive parametric knowledge can effectively navigate its reasoning space $\mathfrak{R}$ to identify optimal evaluation subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and evaluation directions $\rho_{\mathcal{I},\mathcal{D}}$, establishing automatic evaluation as a reliable, scalable, and reproducible methodology for complex contextual quality assessment.

In summary, this study advances the field of prompt engineering through three interconnected contributions: the introduction and empirical validation of the Cumulative Thinking Prompt as a structured pre-generation reasoning method; the development of a rigorously constructed and collaboratively optimized mathematical terminology dataset spanning 18 subjects; and the establishment of a validated automatic evaluation framework comprising eleven metrics with demonstrated reliability against human judgment. Together, these contributions provide both theoretical insights into how sequential knowledge accumulation facilitates optimal reasoning navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ and practical methodologies for prompt design, dataset construction, and evaluation that extend beyond the specific domain of mathematical definition generation to inform broader applications of large language models in educational and technical contexts.

2.5.2 Future Directions

The Cumulative Thinking Prompt's foundational principle, knowledge accumulation before generation, opens promising avenues for extending this paradigm beyond the five-step sequential recall structure employed in the present study. Future research could investigate adaptive knowledge accumulation mechanisms wherein the number and depth of recalling steps $\{r_1, r_2, \ldots, r_N\}$ are dynamically determined based on task complexity, subject domain characteristics, or real-time assessment of the language model's intermediate comprehension. Rather than fixing the accumulation phase at five steps, adaptive approaches could employ meta-reasoning frameworks where the model evaluates its own preparatory knowledge sufficiency before proceeding to final generation, effectively learning to calibrate the depth of pre-generation thinking. Additionally, the knowledge accumulation concept could be extended to multimodal contexts where mathematical definitions may benefit from integrating textual, symbolic, and visual reasoning streams, enabling the model to accumulate not only sequential conceptual recalls but also complementary representational formats that collectively enrich the final output. Exploring whether knowledge accumulation yields differential benefits across model architectures, sizes, and training paradigms would further elucidate the generalizability of this reasoning preparation mechanism.

The multi-agent collaborative methodology employed for dataset construction and

evaluation metric development represents a scalable paradigm for creating rigorously validated research artifacts, yet substantial opportunities remain for advancing this approach. Future work could formalize agent collaboration strategies $\mathcal{S}_{\text{collab}} = \{s_1, s_2, \ldots, s_L\}$ into reusable frameworks with defined roles, communication protocols, and consensus mechanisms that extend beyond sequential refinement to include parallel deliberation, adversarial validation, and hierarchical quality control. The integration of specialized agent roles, such as domain expert agents with fine-tuned knowledge in specific mathematical subdisciplines, diversity-promoting agents that ensure comprehensive subject coverage, and consistency-enforcing agents that maintain notational and structural coherence across the dataset, could yield artifacts with enhanced quality and reduced bias. Furthermore, the collaboration framework could be extended to incorporate iterative human-in-the-loop validation at critical junctures, creating hybrid systems where LLM agents propose and optimize while human experts provide targeted oversight, thereby combining the scalability of automated collaboration with the nuanced judgment of domain specialists. Such frameworks would be particularly valuable for constructing evaluation datasets in specialized technical domains where both breadth and depth of coverage are essential.

The explicit intermediate reasoning generated by the Cumulative Thinking Prompt, the five sequential recalling steps $\{r_1, r_2, \ldots, r_5\}$ that trace the model's cognitive trajectory from essential to abstract understanding, suggests significant potential for applications beyond quality enhancement. These intermediate reasoning traces could serve as valuable pedagogical artifacts, providing learners with transparent insight into how complex mathematical concepts are constructed from foundational principles. Future research could explore leveraging these explicit reasoning steps for personalized learning assistance, where the model's accumulated knowledge pathways are analyzed to identify gaps in conceptual understanding and generate targeted explanatory interventions. Additionally, the transparency afforded by explicit intermediate reasoning opens possibilities for improved interpretability and trustworthiness in high-stakes applications. By externalizing the knowledge accumulation process, these reasoning traces enable users to verify the logical progression underlying generated outputs, detect potential misconceptions or reasoning errors before they propagate to final solutions, and provide targeted feedback for refinement. Investigating how explicit intermediate reasoning can facilitate interactive refinement cycles, where users engage with and modify intermediate steps before final generation, could yield systems that function not merely as autonomous generators but as collaborative reasoning partners, aligning with broader movements toward explainable AI and human-AI collaborative intelligence.

Chapter 3

AUDIT Prompt with Iterative Reasoning: On-the-Fly Mathematical Definition Generation with Cyclic Quality Refinement

3.1 Introduction

Building upon the Cumulative Thinking Prompt research, which established the value of structured knowledge accumulation prior to generation for mathematical definition quality, this study investigates whether iterative reasoning, a fundamentally different navigation paradigm, can further enhance language model performance by incorporating real-time quality assessment and recursive refinement into the generation process. While linear reasoning methods $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ such as the Cumulative Thinking Prompt effectively guide the language model $M_{\text{a-gen}}$ through preparatory knowledge accumulation before producing a final output, they inherently lack mechanisms for intermediate quality checks and corrective refinement. Once the model commits to a reasoning direction $\rho_{\mathcal{I},\mathcal{D}}$ and generates a definition, there is no opportunity within the same prompt framework to evaluate that output, identify deficiencies, and refine it based on explicit feedback. This limitation is particularly consequential for mathematical definition generation, where subtle inaccuracies, incomplete formulations, or deviations from disciplinary conventions may persist undetected in a single-pass generation process, regardless of the quality of preparatory reasoning.

The AUDIT prompt addresses this limitation by introducing an iterative reasoning framework $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\to} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\to} \iota_3)^{\delta_2}\}$ that explicitly incorporates recursive quality audits and refinement cycles into the definition generation process. Rather than treating definition generation as a one-time navigation event, the AUDIT prompt structures the task as a

multi-stage process: an initial definition is generated, followed by a structured audit that identifies specific areas for improvement, then a refined definition is produced based on the audit feedback, and this cycle repeats to progressively enhance output quality. This design enables the language model to leverage previously generated outputs and audit insights as implicit context for subsequent refinements, effectively allowing the model to revisit and adjust its reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ based on self-assessed quality indicators. By incorporating explicit feedback loops into the reasoning navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I},\mathcal{D}) = \mathfrak{R}_{\mathcal{I},\mathcal{D}} \to \mathcal{R}_{\mathcal{I},\mathcal{D}} \to \rho_{\mathcal{I},\mathcal{D}}$, the AUDIT prompt aims to overcome the inherent constraints of linear reasoning approaches for tasks requiring high precision and adherence to formal standards.

To evaluate the effectiveness of iterative reasoning for mathematical definition generation, this study conducts a comprehensive comparison between the AUDIT prompt and established baseline methods, with particular emphasis on head-to-head analysis against the Cumulative Thinking Prompt, the linear reasoning method that demonstrated effectiveness in the first study. The evaluation employs an extended dataset that builds upon the 439 terminologies used in the Cumulative Thinking Prompt research, expanding to 921 mathematical terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ across 18 subjects $S = \{s_1, s_2, \ldots, s_{18}\}$, enabling more robust assessment of performance patterns across diverse mathematical domains. The same 11 evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$ are applied consistently across all prompt methods, ensuring that performance differences can be reliably attributed to the distinct reasoning characteristics of each approach rather than variations in evaluation criteria. Through this comparative framework, the study seeks to determine whether iterative reasoning with explicit quality audits and refinement cycles yields superior definition quality compared to linear reasoning strategies, and to provide insights into how recursive feedback mechanisms influence the language model's ability to navigate the holistic reasoning space $\mathfrak{R}$ toward optimal reasoning directions for complex mathematical definition tasks.

3.2 Related Works

The following section presents comprehensive examinations of representative prompt variations $\mathcal{P} = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$ that enable LLMs as $\mathcal{M} = \{M_{(\text{a-gen})}, M_{(\text{a-eval})}, M_{(\text{m-eval})}, \pi_\theta\}$ to engage in iterative traversal $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\to} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\to} \iota_3)^{\delta_2} \cdots (\overset{\circlearrowleft}{\to} \iota_T)^{\delta_{T-1}}\}$ of their expansive reasoning space $\mathfrak{R} = \bigcup_{u=1}^{U} \mathcal{R}_u$ across diverse task domains, systematically organized into two principal categories encompassing three subcategories. The first major category, "Prompt for Iterative Reasoning," comprises techniques that direct models through recursive cognitive processes via structured reasoning strategies (such as evidence triple verification, privacy-preserving prompt synthesis, teacher-inspired multi-stage retrieval, and abductive explanation tree construction) and multi-agent collaborative frameworks $\mathcal{A} = \{A_1, A_2, \ldots, A_K\}$ where $A_k = (\mathcal{M}_k, \mathcal{C}_k, \mathcal{R}_k)$ (including inter-model debate architec-

tures, planning-guided search with world model simulation, and principle distillation from diverse exemplar sets). The second major category, "Prompt Construction," addresses methodologies for crafting effective prompts through iterative demonstration generation $\mathcal{D} = \{d_1, d_2, \ldots, d_K\}$ (employing bidirectional synthesis cycles that create and subsequently select high-quality exemplars through complexity-based clustering mechanisms).

Table 3.1: AUDIT Prompt: Citations of Iterative Reasoning and Constructions

Main Category	Subcategory	Prompt Methods
Prompt for Iterative Reasoning	Prompt for Iterative Reasoning with Specific Reasoning Strategy	• Chain-of-Knowledge (CoK) [55] • Differentially-Private Offsite Prompt Tuning (DP-OPT) [56] • Teaching-Inspired Integrated Prompting [57] • Maieutic Prompting [58]
	Prompt for Iterative Reasoning with Agents in Collaboration	• Multiagent Debate [59] • Reasoning via Planning (RAP) [60] • Principle-Based Prompting [61]
Prompt Construction	Prompt Construction with Iterative Demonstration Generation	• Synthetic Prompting [62]

3.2.1 Iterative-Reasoning Prompt Applications

Prompt for Iterative Reasoning with Specific Reasoning Strategy

Chain-of-Knowledge (CoK) [55] prompting is a method designed to make the reasoning process of large language models more reliable by changing the format of their output in a constrained reasoning subspace. Instead of letting the model generate a free-form chain of thought, the CoK prompt instructs it to produce its reasoning as a series of explicit, structured facts, called evidence triples, formatted as (subject, relation, object), alongside a plain language explanation. This structured output directly tackles the common problem of hallucination, where models invent plausible-sounding but incorrect information. Because the reasoning is broken down into discrete, verifiable pieces of knowledge, it becomes possible to check each piece against a trusted external knowledge base. This is the role of the F^2-Verification step, which evaluates the generated triples for factuality (do they match known facts?) and faithfulness (is the explanation logically connected to both the triples and the final answer?). If the verification finds the reasoning to be unreliable, the framework triggers a rethinking process, feeding the correct information back into the model to generate a new, corrected answer. The principal evidence for the effectiveness of this entire framework can be seen in the results for the OpenBookQA [36] task using the GPT-3.5-Turbo [26] model. While standard Manual CoT [12] prompting achieved 82.6% accuracy, the CoK method alone improved this to 83.5%. When the full framework, including CoK prompting and the F^2-Verification with its rethinking process, was applied, the accuracy further increased to 85.0%, demonstrating the concrete benefit of making the reasoning transparent and verifiable.

Differentially-Private Offsite Prompt Tuning (DP-OPT) [56] is a method for creating a discrete text prompt using a local language model, such as Vicuna-7B [63], without exposing private data to the cloud. The process begins with an initial task instruction. A small batch of private training samples is used to evaluate this instruction, and the results, showing which samples were correctly or incorrectly predicted, are fed back to the local LLM as in-context examples. This "backward pass" prompts the local model to generate new candidate instructions. Crucially, to protect privacy, this generation step does not use the private data directly. Instead, the data is partitioned into disjoint subsets, and the local LLM is queried with each subset separately. A voting mechanism then counts the proposed next token from each query, and a differentially-private algorithm (called LimitedDomain) selects a token based on these votes while adding calibrated noise. This process repeats token-by-token to build a complete prompt. After multiple candidate prompts are generated, another private selection mechanism chooses the best one based on its accuracy on a validation set. The final output is a plain-text instruction that can be sent to a cloud-based LLM (like GPT-3.5 [26]) for inference. The principal evidence supporting its effectiveness comes from experiments on four classification tasks (SST-2, Trec, Mpqa, Disaster). When prompts generated by DP-OPT on Vicuna-7B were transferred to DaVinci-003, they achieved an average test accuracy of 81.4%. This performance closely approached the non-private in-context learning baseline (82.9%) and significantly outperformed a private soft-prompt baseline (PromptDPSGD [64]) which scored only 70.5%, all while providing a formal mathematical guarantee that the final prompt does not leak the private information used in its creation.

The **Teaching-Inspired Integrated Prompting** [57] Framework operates through a structured three-stage process designed to mimic how a teacher guides students. In the first stage, the prompt is constructed by retrieving two types of external information. Similar solved problems are selected from a database by tokenizing the test problem and using the BM25 [65] algorithm to find candidates, which are then ranked by their Longest Common Subsequence length with the test problem; the top similar problem is included with its analysis and answer. Simultaneously, relevant background knowledge is retrieved by extracting key knowledge points and uncertain theorems from the test problem (with help from an LLM) and using these as queries against a knowledge database of theorems and concepts; the top three candidates are selected based on their textual similarity to the problem. These elements are combined with two few-shot exemplars that include both Chain-of-Thought reasoning steps and Program-of-Thought Python code to demonstrate solution approaches. In the second stage, the model generates multiple solution paths using self-consistency [3] sampling, producing both step-by-step natural language answers and executable Python program outputs. In the third stage, an answer selection mechanism is applied. This includes a double-check verification that compares each path's Python program output with its corresponding step-by-step answer to check

for consistency. If all paths are consistent, majority voting determines the final answer. If inconsistencies arise, the model generates additional answers, and for Chinese problems, an English-Chinese ensemble strategy translates the problem and context into English before generating further solutions. A further selection step then compares the frequency of the most common answer from the Python programs against the most common answer from the step-by-step solutions, choosing the one with higher frequency as the final output. The principal evidence supporting this framework's effectiveness comes from experiments on nine benchmarks using GPT-3.5-Turbo [26] and GPT-4 [25]. The framework achieves new state-of-the-art performance on four math benchmarks with GPT-4 [25]: AddSub [66] at 98.2% (+3.3% improvement), SVAMP [67] at 93.9% (+0.2% increase), Math23K [68] at 94.3% (+7.2% increase), and AQuA [69] at 81.1% (1.2% increase). On GPT-3.5-Turbo [26], it improves accuracy on the research created MathMC dataset by 18.8% and on MathToF by 10.5%.

Maieutic Prompting [58] is a few-shot, text-based prompt method designed to improve the logical consistency of a language model's reasoning. The process begins by taking a question and prompting the model to generate two separate abductive explanations: one that supports a "True" answer and one that supports a "False" answer. For example, given a question, the model is asked to complete "True, because..." and "False, because...". This is the first level of the "maieutic tree." Recognizing that these first explanations might themselves be unreliable, the method then performs recursive validation. It takes each generated explanation and treats it as a new question, prompting the model again to justify both true and false answers for that explanation. This process continues to a specified depth, creating a tree of interconnected propositions. However, not all branches are kept; the method only continues to expand from a proposition if the model is inconsistent about its truth value (i.e., if it gives the same answer to the proposition and its negation). Once the tree is built, the method quantifies two types of relationships: the model's "belief" in each proposition (based on its probability assignments) and the "consistency" between a child explanation and its parent (based on generation likelihoods). To further strengthen the model, a separate Natural Language Inference (NLI) model is used to check for entailment or contradiction between all pairs of propositions in the tree, adding more logical constraints. Finally, all these beliefs and logical relationships are converted into weighted constraints and fed into a MAX-SAT solver [70], a symbolic algorithm that finds the single set of truth values (True/False) for every proposition in the tree that satisfies the maximum possible weight of constraints. The truth value assigned to the original question by this solver is the final, robust answer. The principled evidence from the paper shows this method's effectiveness: on the Com2Sense [71] benchmark, Maieutic Prompting achieved 72.5% accuracy, significantly outperforming other prompt methods like Chain of Thought [12] (61.6%) and Self-Consistency [3] (61.4%). Furthermore, its robustness was demonstrated with a pairwise accuracy of 75.0% on Com2Sense, a metric

testing questions with opposite meanings, which was substantially higher than the 68.7% achieved by the next best prompt method.

Algorithm 12 Pseudocode: Iterative Reasoning with Specific Reasoning Strategy

Require: Input x, Prompt instructions $\mathcal{I}$, Prompt demonstrations $\mathcal{D}$ (optional), Target LLM $M_{\text{a-gen}} \in \mathcal{M}$ for generation, Reasoning strategy configuration $\mathcal{C}_{\text{reason}}$, Maximum iterations T
Ensure: Final output with verified reasoning
Candidate Generation
Generate initial reasoning outputs according to $\mathcal{C}_{\text{reason}}$
Quality Verification
Evaluate generated outputs using verification mechanisms
Iterative Refinement
for iteration $t = 1$ **to** T **do**
    if verification identifies unreliable outputs **then**
        Incorporate verification feedback into reasoning context
        Generate refined outputs via $M_{\text{a-gen}}$
        Re-evaluate refined outputs
    else
        break
    end if
end for
Output Selection
Apply selection mechanism according to $\mathcal{C}_{\text{reason}}$
return Final verified output

Prompt for Iterative Reasoning with Agents in Collaboration

Multiagent Debate [59] is defined as a manually-crafted, zero-shot text prompting technique that orchestrates an iterative debate between multiple language model instances. The method operates by first using a "starting" prompt that instructs each independent model agent to generate an initial answer to a given query. Following this, a specialized "debate" or "consensus" prompt is employed in subsequent rounds. This prompt takes the responses from all agents and feeds them back to each individual agent, with instructions to critically examine the collective input and refine its own answer accordingly. This cycle of sharing answers and prompting for revision is repeated over multiple rounds, allowing the agents to engage in a structured, multi-turn interaction. The core mechanism leverages the language model's ability to process and critique text, compelling each agent to consider alternative perspectives and reasoning paths. This process is designed to enhance the final output by encouraging the models to move beyond their initial, potentially flawed, responses and converge on an answer that is more robust, factually valid, and logically sound. The prompt templates are kept consistent across all tasks, demonstrating the method's general applicability as a prompt strategy. The conducted experiments demonstrate that this debate process leads to significant performance gains across multiple benchmarks. For instance, on grade school math problems (GSM8K [22]), the Multi-Agent (Debate) approach achieved an accuracy of $85.0 \pm 3.5\%$, a clear improvement over the Single Agent baseline at $77.0 \pm 4.2\%$ and even over a Multi-Agent majority voting method at $81.0 \pm 3.9\%$ (Table 1). The paper also provides qualitative evidence, such as in Figure 4

and 5, showing cases where all agents initially provided incorrect answers but, through the iterative critique process of the debate, arrived at the correct final answer. This confirms that the method's strength lies its iterative and collaborative reasoning design that helps the language model to converge to correct solution.

The **Reasoning via Planning (RAP)** [60] framework operates by using distinct sets of manually-crafted, few-shot text prompts to repurpose a single large language model into two complementary roles that work together during the reasoning process. In its role as a reasoning agent, the model is given prompts containing task instructions and example demonstrations that ask it to propose the next logical step or action based on the current situation, for instance, in the Blocksworld task [72], the prompt shows several examples of initial block configurations and their corresponding action plans, then asks the model to generate the next action for a new configuration. In its parallel role as a world model, the same language model receives different prompts that instruct it to simulate the consequences of taking that proposed action, predicting how the situation will change, such as showing examples of how picking up a block alters which blocks are clear and what the hand is holding. A third set of prompts guides the model to evaluate its own reasoning steps, asking questions like "Is this reasoning step correct?" and using the model's confidence in its answer as a quality signal. These three prompt-driven functions, action proposal, outcome simulation, and self-evaluation, provide the necessary components for a Monte Carlo Tree Search algorithm [73], [74] to strategically explore multiple possible reasoning paths. The search process builds a tree of reasoning steps, uses the world model to look ahead at future consequences, employs the self-evaluation signals to assess the promise of different branches, and backtracks to try alternatives when a path appears unpromising. This structured, iterative approach enables the model to move beyond simply generating a single linear chain of thoughts, instead allowing it to deliberately plan by considering alternatives, anticipating future states, and refining its choices based on projected outcomes. On 6-step Blocksworld problems where simple Chain-of-Thought [12] prompting achieves 0% success, RAP achieves 42% success with LLaMA-33B [75] (Table 1); on mathematical reasoning (GSM8K [22]), RAP with aggregation achieves 51.8% accuracy compared to 48.1% for Chain-of-Thought [12] with self-consistency [3] (Table 2); and on logical reasoning ProntoQA [76], RAP achieves 78.8% proof accuracy versus 64.8% for standard Chain-of-Thought (Table 3). These consistent improvements across diverse tasks demonstrate that the prompt-based transformation of a single model into an agent with a world model, combined with strategic search, enables more effective reasoning than linear generation approaches.

Principle-Based Prompting [61] is a multi-agent prompt strategy designed to improve how Large Language Models (LLMs) perform text classification. The process begins in the "Principle Generation" step, where several different LLM agents are each

given a small set of demonstration examples (ranging from 4 to 16 samples) from the target classification task. These agents are instructed to analyze the examples and independently generate a set of candidate principles or guiding rules that distinguish between the different classes, such as rules for identifying anger versus joy in text. This creates a diverse pool of 36 candidate principles, generated by varying the number of examples, whether the examples include labels, and the type of LLM used. Next, in the "Principle Consolidation" step, a final LLM agent (the finalizer) receives all these candidate principles and synthesizes them into a single, comprehensive, and non-redundant set of principles. The core insight is that this final set of model-generated text principles captures the essential knowledge needed for the task. Finally, in the "Text Classification" step, these consolidated principles are inserted as the core instruction into a zero-shot prompt. The classifier LLM is then given this prompt, which contains the principles but no examples, and is asked to classify new, unseen data. The principal evidence supporting this method's efficiency and effectiveness is found in the paper's experimental results. On the Emotion20 dataset with the FLAN-UL2 model, the principle-based consolidation approach achieved a 15.11% improvement in macro F1 score over a standard zero-shot prompt. This significantly outperformed providing the model with eight examples (0.92% improvement) and even surpassed the performance of human-crafted principles on the PC2 dataset (19.37% vs. 13.26%), all while using an input token length equivalent to only a 2-shot example, proving it to be a more efficient and often more powerful alternative.

Algorithm 13 Pseudocode: Agent Collaboration with Iterative Reasoning

Require: Input x, Prompt instructions $\mathcal{I}$, Prompt demonstrations $\mathcal{D}$ (optional), Set of agents $\mathcal{A} = \{A_1, A_2, \ldots, A_K\}$ where $A_k = (\mathcal{M}_k, \mathcal{C}_k, \mathcal{R}_k)$, Collaboration strategy $\mathcal{S}_{\text{collab}}$, Maximum interaction rounds T

Ensure: Final output from coordinated agent reasoning

Initialization
Configure agents $\mathcal{A}$ according to $\mathcal{S}_{\text{collab}}$
Initialize coordination state $\mathcal{A}_{\text{coord}} \leftarrow \mathcal{A}$
Iterative Collaborative Reasoning
for round $t = 1$ **to** T **do**
    for each agent $A_k \in \mathcal{A}_{\text{coord}}$ **do**
        Generate output via A_k based on input x and shared context
    end for
    Aggregate agent outputs according to $\mathcal{S}_{\text{collab}}$
    Evaluate current state against collaboration objectives
    if termination condition met **then**
        break
    else
        Update coordination state $\mathcal{A}_{\text{coord}}$ based on aggregated outputs
        Distribute feedback to agents for next round
    end if
end for
Output Synthesis
Synthesize final output from aggregated agent responses
return Final output

3.2.2 Iterative-Reasoning Prompt Construction

Prompt Construction with Iterative Demonstration Generation

Synthetic Prompting [62] is a few-shot method where a large language model generates its own improved prompt demonstrations. It begins with a small number of manually-crafted seed examples, each consisting of a question and a reasoning chain written as Python code. The method then employs a unique two-step synthesis process to create new, high-quality examples. In the first step, called the backward process, the model is prompted to generate a reasoning chain of a specific length for a given topic, and then to create a question that logically matches that chain, ensuring the question is answerable. In the second step, the forward process, the same model is prompted again to generate a more precise and detailed reasoning chain specifically for the newly created question. This two-step cycle is repeated to build a large pool of synthetic question-answer pairs. To construct the final prompt, the method does not use all generated examples but selects the most effective ones. It employs an "in-cluster complexity" selection scheme, which groups the synthetic examples by their semantic meaning and chooses the most complex example (the one with the longest reasoning chain) from each group. This ensures the final set of demonstrations is both diverse and informative. The principal evidence supporting the effectiveness of this approach is presented in Table 2 of the paper. On the GSM8K [22] math reasoning dataset with only 2 seed examples, Synthetic Prompting achieved an accuracy of 74.7% , which is a meaningful improvement over standard PAL prompting, which scored 72.5% using the same seed examples. This demonstrates that the model-generated and carefully selected prompt demonstrations were significantly more effective at eliciting correct reasoning than the original manually-crafted ones.

Algorithm 14 Pseudocode: Iterative Prompt Demonstration Generation

Require: Seed content $\mathcal{C}_{\text{seed}} = \{c_1, c_2, \ldots, c_N\}$, Target LLM $M_{\text{a-gen}} \in \mathcal{M}$ for generation, Number of iterations T, Number of demonstrations K

Ensure: Optimized demonstration set $\mathcal{D}_{\text{selected}}$

Initialization
Initialize demonstration pool $\mathcal{D}_{\text{pool}} \leftarrow \emptyset$
Initialize current content with $\mathcal{C}_{\text{seed}}$
Iterative Generation
for iteration $t = 1$ **to** T **do**
    Generate synthetic outputs based on current content
    Refine generated outputs via subsequent generation steps
    Add refined outputs to $\mathcal{D}_{\text{pool}}$
end for
Demonstration Selection
Select K demonstrations from $\mathcal{D}_{\text{pool}}$ based on selection criteria:
    - Complexity or diversity metrics
    - Clustering or semantic grouping
    - Representativeness of reasoning patterns
Set $\mathcal{D}_{\text{selected}}$ as final demonstration set
return $\mathcal{D}_{\text{selected}}$

3.3 Methodology

3.3.1 Problem Definition

The AUDIT prompt addresses the fundamental limitation inherent in linear reasoning approaches $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ for mathematical definition generation: the absence of mechanisms for real-time quality assessment and iterative refinement during the generation process. While linear reasoning strategies such as the Cumulative Thinking Prompt effectively guide the language model $M_{\text{a-gen}}$ through structured pre-generation knowledge accumulation, they produce a final definition without incorporating intermediate quality checks or opportunities for corrective refinement based on identified deficiencies. This one-pass generation paradigm constrains the model's ability to detect and rectify subtle inaccuracies, incomplete formulations, or deviations from standard mathematical conventions that may arise during the initial generation.

Formally, given a mathematical term $t_{(i,j)}$ within subject s_i, the task of mathematical definition generation can be characterized as a reasoning navigation problem $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D}) = \mathfrak{R}_{\mathcal{I},\mathcal{D}} \to \mathcal{R}_{\mathcal{I},\mathcal{D}} \to \rho_{\mathcal{I},\mathcal{D}}$, where the language model must traverse from the holistic reasoning space $\mathfrak{R}$ toward an optimal reasoning direction $\rho_{\mathcal{I},\mathcal{D}}$ that yields a high-quality definition. Linear reasoning methods execute this navigation as a single forward trajectory without feedback loops, meaning that once a particular reasoning direction is selected, the model cannot revisit or refine its path based on intermediate output quality. This limitation is particularly consequential for mathematical definition generation, where the precision of formal statements, completeness of necessary conditions, and adherence to notational conventions require careful scrutiny that may benefit from iterative revision.

The AUDIT prompt addresses this problem by introducing an iterative reasoning framework $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\to} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\to} \iota_3)^{\delta_2}\}$ that explicitly incorporates quality audits and refinement cycles into the generation process. Rather than producing a single definition in one pass, the method generates an initial definition, evaluates it through a structured audit that identifies specific areas for improvement, refines the definition based on audit feedback, and repeats this cycle to progressively enhance output quality. This approach treats definition generation not as a one-time navigation event but as a recursive process where each iteration leverages previously generated outputs and audit insights to refine the model's traversal of the reasoning space. By incorporating explicit feedback mechanisms, the AUDIT prompt enables the language model to dynamically adjust its reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ based on self-assessed quality indicators, thereby overcoming the inherent constraints of linear reasoning for tasks requiring high precision and adherence to formal standards.

3.3.2 Dataset

To enable a more rigorous evaluation of the AUDIT prompt's iterative reasoning mechanism, this study employs an extended dataset that builds upon the dataset originally proposed and optimized in the Cumulative Thinking Prompt research. While maintaining the same 18 mathematical subjects $S = \{s_1, s_2, \ldots, s_{18}\}$, the dataset significantly expands the number of math terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ under each subject, increasing the total number of terms from 439 to 921, representing an addition of 482 terms and a relative increase of 109.8%. This substantial expansion enables more accurate evaluation of performance gains and losses across the proposed AUDIT prompt and baseline prompt methods, providing deeper insights into how iterative reasoning $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\rightarrow} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\rightarrow} \iota_3)^{\delta_2} \cdots (\overset{\circlearrowleft}{\rightarrow} \iota_T)^{\delta_{T-1}}\}$ elicited by the AUDIT prompt helps the language model $M_{\text{a-gen}}$ navigate the vast reasoning space $\mathfrak{R}$ across nearly 1,000 mathematical terms spanning 18 distinct domains.

Table 3.2: AUDIT Prompt Dataset: Extended Dataset

Subject	Original Dataset	Extended Dataset	Increased Number	Percentage Increase (%)
Abstract Algebra	26	52	+26	100.0%
Algebra	24	50	+26	108.3%
Calculus	24	51	+27	112.5%
Differential Geometry	25	50	+25	100.0%
Discrete Mathematics	25	53	+28	112.0%
Dynamical Systems	23	52	+29	126.1%
Geometry	24	50	+26	108.3%
Graph Theory	26	54	+28	107.7%
Information Theory	24	50	+26	108.3%
Linear Algebra	25	51	+26	104.0%
Logic	24	52	+28	116.7%
Measure Theory	23	50	+27	117.4%
Numerical Linear Algebra	24	50	+26	108.3%
Optimization	26	52	+26	100.0%
Probability	24	50	+26	108.3%
Statistics	24	53	+29	120.8%
Topology	24	51	+27	112.5%
Trigonometry	24	50	+26	108.3%
Total	439	921	+482	109.8%

The expansion of math terms followed the same multi-agent collaborative procedure that proved effective in constructing and optimizing the first version of the dataset and evaluation framework. DeepSeek served as the suggesting agent $\mathcal{A}_{\text{suggest}}$ to propose an expanded coverage scope with additional subject-relevant mathematical terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ for each subject $s_i \in S$. Subsequently, Qwen, ChatGPT, and Gemini were deployed as evaluator agents $\mathcal{A}_{\text{eval}} \subseteq \mathcal{A}$ and optimizer agents $\mathcal{A}_{\text{optim}} \subseteq \mathcal{A}$ to iteratively review the newly suggested terms, evaluate their relevance and appropriateness, and perform further optimizations by removing or adding terms until all participating language model agents reached a high degree of consensus. This collaborative and iterative procedure, which had been rigorously validated in the construction of the original dataset

and evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$, was reapplied in this dataset expansion task as an evidence-based methodology, ensuring the reliability and quality of the extended terminology collection.

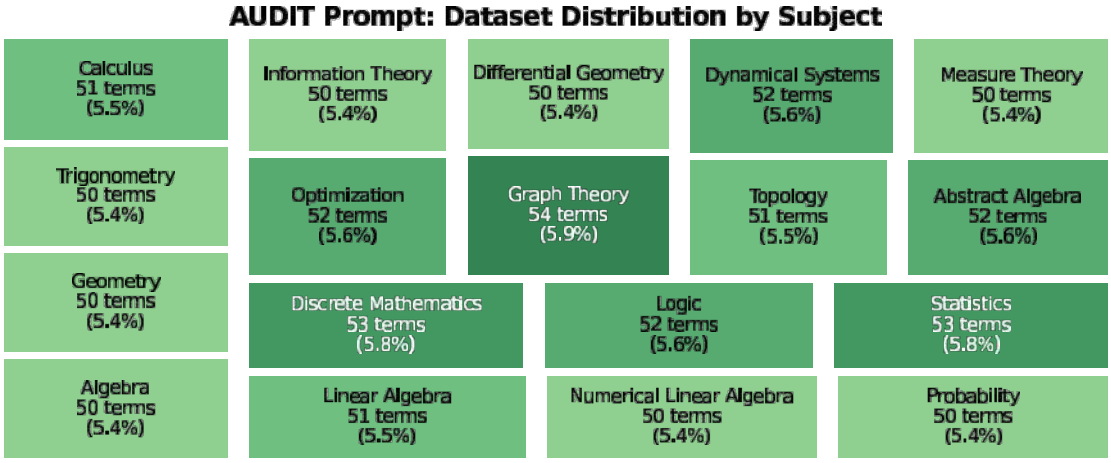


Figure 3.1: AUDIT Prompt: Dataset Distribution by Subject

3.3.3 Methods

Proposed Prompt Method

The AUDIT prompt introduces an iterative math definition generation and refinement cycle that leverages previously generated definitions and audit reports as implicit context to progressively enhance generation quality. The method operates through a structured five-step procedure: an initial definition is generated in the first paragraph as "definition_1", followed by a quality audit in the second paragraph as "quality_audit_1" that provides targeted refinement suggestions, then a first refined definition is produced in the third paragraph as "definition_2", subsequently a second audit in the fourth paragraph as "quality_audit_2" offers further optimization recommendations, and finally a second refined definition is delivered in the fifth paragraph as "definition_3". This iterative reasoning $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\rightarrow} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\rightarrow} \iota_3)^{\delta_2}\}$ framework enables the language model $M_{\text{a-gen}}$ to recursively refine its outputs by treating the previously generated definition and accompanying audit as contextual guidance, allowing for increasingly advanced refinement suggestions and improved generation quality at each cycle. The iterative nature of this process ensures that each iteration builds upon the strengths and addresses the limitations identified in prior stages, ultimately converging toward definitions that more strictly align with the standard mathematical definitions found in college-level mathematics lecture notes and textbooks.

Proposed Prompt

AUDIT Prompt:
definition of [term] in [subject] with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the first paragraph as "definition_1", then suggest math definition refinement suggestions in the second paragraph as "quality_audit_1", then generate the first refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the third paragraph as "definition_2", then suggest math definition refinement suggestions to further refine the "definition_2" in the fourth paragraph as "quality_audit_2", then generate the second refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the fifth paragraph as "definition_3:". Make sure each of the sequential definitions should be optimized to be more strictly aligned with the standard math definitions of college level mathematics lecture notes and textbooks.

Algorithm 15 Pseudocode: AUDIT Prompt

Require: Math subjects $S = \{s_1, s_2, \ldots, s_m\}$, Math terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ for each subject, AUDIT prompt p_{audit}, Number of audit rounds R, Generation LLM $M_{\text{a-gen}} \in \mathcal{M}$ accessed via API, Evaluation LLM $M_{\text{a-eval}} \in \mathcal{M}$ accessed via API, Eleven evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$
Ensure: Evaluation results $\mathcal{R}$
for each subject $s_i \in S$ **do**
  for each term $t_{(i,j)} \in T_{s_i}$ **do**
    Generate initial definition d_1 using p_{audit} via $M_{\text{a-gen}}$
    for audit round $r = 1$ **to** $R - 1$ **do**
      Generate quality audit a_r with refinement suggestions for definition d_r using p_{audit} via $M_{\text{a-gen}}$
      Generate refined definition d_{r+1} based on audit a_r using p_{audit} via $M_{\text{a-gen}}$
    end for
    Final definition $g_{(i,j)}^{(p_{\text{audit}})} = d_R$
    for each evaluation metric $e_k \in \mathcal{E}$ **do**
      Evaluate generated final definition $g_{(i,j)}^{(p_{\text{audit}})}$ using metric e_k via $M_{\text{a-eval}}$
      Store evaluation score $r_{(i,j,p_{\text{audit}},k)}$
    end for
  end for
end for
Compute aggregate scores for AUDIT prompt
return Evaluation results $\mathcal{R}$ for AUDIT method

Baseline Prompt Methods

To enable meaningful comparative analytics, this study evaluates the proposed AUDIT prompt against a set of established baseline prompt methods, with particular emphasis on the head-to-head comparison between the AUDIT prompt as an iterative reasoning strategy $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\rightarrow} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\rightarrow} \iota_3)^{\delta_2}\}$ and the Cumulative Thinking Prompt as a linear reasoning strategy $\mathcal{L} = \{\ell_1 \rightarrow \ell_2 \rightarrow \cdots \rightarrow \ell_R\}$. While the Cumulative Thinking Prompt demonstrated effectiveness through structured pre-generation knowledge accumulation in the first research study, its linear nature lacks on-the-fly quality audit, refinement suggestions, and dynamic iterative optimization during the generation process. In contrast, the AUDIT prompt introduces a recursive refinement cycle where each definition generation is immediately followed by a targeted audit and subsequent refinement, enabling the language model $M_{\text{a-gen}}$ to continuously improve output quality based on explicit feedback. This head-to-head comparison between linear and iterative reasoning approaches provides meaningful insights into the distinct benefits and drawbacks of each reasoning pattern,

with the AUDIT prompt offering the advantage of real-time quality assessment and progressive optimization that linear strategies cannot inherently provide. Additional baselines including the direct prompt p_{direct}, excellent-professor-aware prompt $p_{\text{e-pro}}$, and step-by-step prompt p_{sbs} further contextualize the performance of both reasoning paradigms within the broader landscape of commonly applied prompting techniques.

Baseline Prompt Methods

Direct Prompt:
definition of [term] in [subject] with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

Excellent-Professor-Aware Prompt:
you are an excellent math professor who are good at math teaching, generate definition of [term] in [subject] with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

Step-by-Step Prompt:
let's think step by step, definition of [term] in [subject] with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

Cumulative Thinking Prompt:
recall your understanding about [term] in [subject] from essential to abstract in sequential 5 times, then to generate definition of [term] in [subject] with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format, based on all 5 previous sequential thinking before generating the mathematical definition.

The pseudocode for each of these baseline methods is provided in algorithms as 9, 10, 11, and 8, respectively.

3.3.4 Experiment Setup

Hyperparameters

The hyperparameter configurations adopted in this study follow the same settings established in the Cumulative Thinking Prompt research, which were carefully selected to ensure stable inference and reproducible generation outcomes across all experimental runs. By maintaining consistency with the hyperparameters detailed in Table 2.4, this study preserves methodological alignment with the first investigation while enabling reliable evaluation of the AUDIT prompt's iterative refinement mechanism. This uniform experimental framework guarantees that any observed performance differences can be attributed to the prompt strategies themselves rather than variations in underlying model configurations.

3.3.5 Evaluation Metrics

To systematically evaluate how iterative reasoning $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\rightarrow} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\rightarrow} \iota_3)^{\delta_2} \cdots (\overset{\circlearrowleft}{\rightarrow} \iota_T)^{\delta_{T-1}}\}$ within the AUDIT prompt influences mathematical definition generation quality, this study employs the same evaluation framework comprising 11 distinct metrics that initially proposed and verified in the first research and presented in Table 2.5. This framework

enables direct comparison between the AUDIT prompt's iterative refinement mechanism and commonly applied prompt strategies, including a head-to-head comparison with the Cumulative Thinking Prompt, a linear reasoning method $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ that demonstrated effectiveness in the first study. By applying the same evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$ consistently across both iterative and linear reasoning approaches, this study ensures that performance differences can be reliably attributed to the distinct reasoning navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ characteristics of each prompt method rather than variations in evaluation criteria.

3.4 Results and Analytics

3.4.1 Holistic Performance Analytics

The following section presents the holistic relative and absolute performance analytics for the proposed AUDIT prompt alongside established baseline methods, including the Cumulative Thinking Prompt from the first study. This comparative analysis aims to elucidate how iterative reasoning $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\to}\ \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\to}\ \iota_3)^{\delta_2}\}$ and linear reasoning $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ differ in eliciting reliable generation from the language model $M_{\text{a-gen}}$ for mathematical term definition tasks. By examining both relative performance against a common baseline and absolute performance metrics, this analysis reveals the distinct advantages and stability characteristics of iterative refinement strategies compared to their linear counterparts.

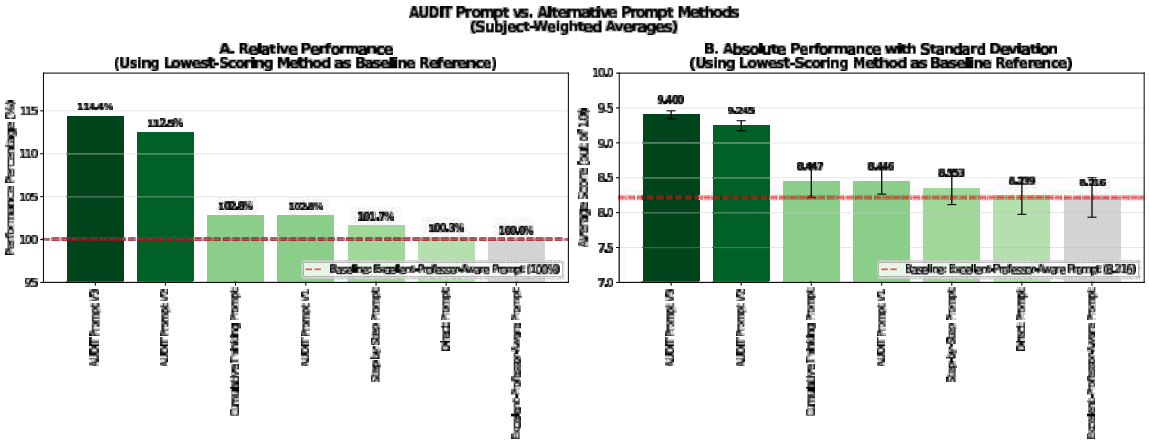


Figure 3.2: AUDIT Prompt: Holistic Relative and Absolute Performances (All Variations)

The holistic relative performance analysis, with the lowest-performing excellent-professor-aware prompt normalized to 100%, reveals a clear performance hierarchy with distinct clustering patterns. The top-performing cluster comprises AUDIT Prompt V3 (114.42%, +14.42%) and AUDIT Prompt V2 (112.52%, +12.52%), demonstrating substantial advantages over all other methods, with performance gains exceeding 12% above baseline. The middle cluster includes the Cumulative Thinking Prompt (102.81%, +2.81%) and AUDIT

Prompt V1 (102.80%, +2.80%), which achieve comparable improvements of approximately 2.8%, indicating that both linear reasoning with knowledge accumulation and single-iteration iterative reasoning yield similar moderate gains. The lower cluster encompasses step-by-step prompt (101.66%, +1.66%) and direct prompt (100.27%, +0.27%), which show marginal improvements over baseline. The numerical evidence strongly indicates that the multi-iteration refinement cycles employed in AUDIT V2 and V3, particularly the three-stage iterative process of generation, audit, and refinement, provide substantial advantages over the single-pass knowledge accumulation of the Cumulative Thinking Prompt, with AUDIT V3 outperforming the Cumulative Thinking Prompt by 11.61 percentage points.

Table 3.3: AUDIT Prompt: Holistic Relative Performance (All Variations)

Prompt Variation	**Holistic Relative Performance (%)**	**Performance Difference (%)**
AUDIT Prompt V3	114.42%	+14.42%
AUDIT Prompt V2	112.52%	+12.52%
Cumulative Thinking Prompt	102.81%	+2.81%
AUDIT Prompt V1	102.80%	+2.80%
Step-by-Step Prompt	101.66%	+1.66%
Direct Prompt	100.27%	+0.27%
Excellent-Professor-Aware Prompt	100.00%	0.00%

The holistic absolute performance analysis further substantiates the superiority of iterative reasoning, revealing that AUDIT V3 achieves the highest absolute performance (9.400) with the smallest standard deviation (0.058), indicating both superior quality and exceptional stability across all 18 mathematical subjects. AUDIT V2 follows with strong performance (9.245, std = 0.071), while the Cumulative Thinking Prompt (8.447, std = 0.236) and AUDIT V1 (8.446, std = 0.179) occupy a middle tier with notably higher variability. The baseline methods, step-by-step prompt (8.353, std = 0.239), direct prompt (8.239, std = 0.261), and excellent-professor-aware prompt (8.216, std = 0.283), exhibit the lowest absolute performance and highest variability, with standard deviations nearly five times larger than that of AUDIT V3. The min-max ranges further illustrate stability differences: AUDIT V3 spans only 0.190 points (9.315–9.505), while the excellent-professor-aware prompt spans 0.905 points (7.714–8.619), demonstrating that iterative reasoning not only yields higher average quality but also provides more consistent performance across diverse mathematical domains. These findings collectively indicate that the recursive refinement mechanism inherent to iterative reasoning enables the language model to navigate the reasoning space $\mathfrak{R}$ more reliably than linear reasoning strategies, converging toward optimal reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ with greater consistency and less susceptibility to subject-specific variations.

Table 3.4: AUDIT Prompt: Holistic Absolute Performance (All Variations)

Prompt Variation	Baseline	Holitic Absolute Performance	Performance Difference	Std	Min	Max
AUDIT Prompt V3	8.216	9.400	1.184	0.058	9.315	9.505
AUDIT Prompt V2	8.216	9.245	1.029	0.071	9.104	9.329
Cumulative Thinking Prompt	8.216	8.447	0.231	0.236	7.965	8.741
AUDIT Prompt V1	8.216	8.446	0.230	0.179	7.975	8.709
Step-by-Step Prompt	8.216	8.353	0.137	0.239	7.900	8.787
Direct Prompt	8.216	8.239	0.023	0.261	7.743	8.639
Excellent-Professor-Aware Prompt	8.216	8.216	0.000	0.283	7.714	8.619

3.4.2 Subject-Level Performance Analytics

The following section presents subject-level absolute and relative performance analytics for the proposed AUDIT prompt alongside baseline prompt methods, including the Cumulative Thinking Prompt from the first study. This granular analysis examines how iterative reasoning $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\rightarrow} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\rightarrow} \iota_3)^{\delta_2}\}$ performs across individual mathematical subjects, revealing domain-specific patterns that complement the holistic findings.

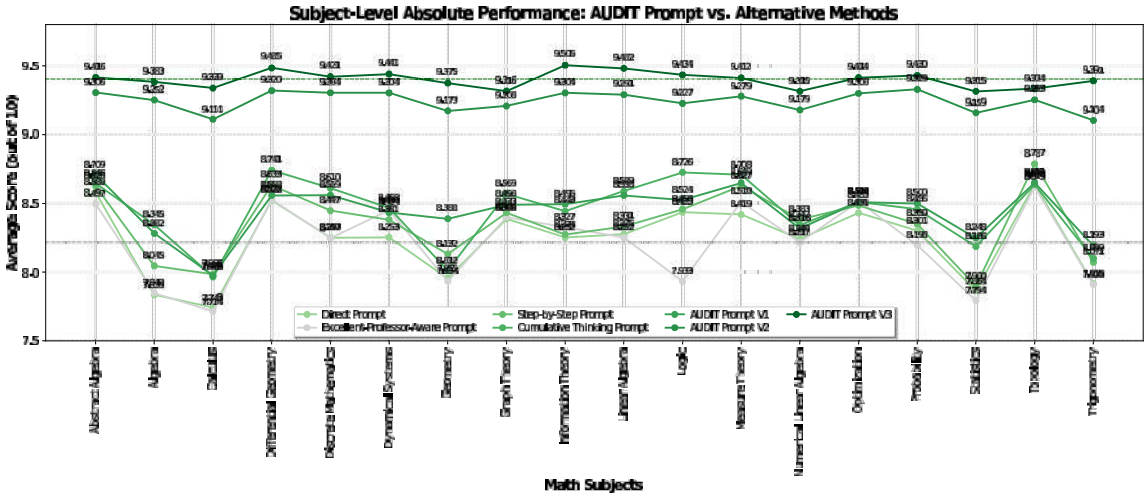


Figure 3.3: AUDIT Prompt: Subject-Level Absolute Performance (All Variations)

The subject-level absolute performance analytics demonstrate that AUDIT Prompt V3 and V2 achieve significantly higher performance across every mathematical subject compared to all other prompt variations. AUDIT V3 leads with scores ranging from 9.315 in Numerical Linear Algebra to 9.505 in Information Theory, while AUDIT V2 follows closely with scores between 9.104 and 9.329. The performance gap between these top-tier iterative methods and the remaining prompt variations is substantial and visually evident, Cumulative Thinking Prompt and AUDIT V1 cluster together with scores ranging from 7.965 to 8.741, residing noticeably lower on the performance scale. This distinct separation between the two-stage iterative methods (V2 and V3) and the single-pass

methods provides convincing evidence that iterative refinement across multiple cycles, particularly the three-stage process of generation, audit, refinement, second audit, and final refinement embedded in AUDIT V3, enables the language model $M_{\text{a-gen}}$ to produce substantially higher quality mathematical definitions. In contrast, AUDIT V1, which represents only the initial generation stage without iterative audit and refinement, shows performance nearly identical to the Cumulative Thinking Prompt (8.446 vs. 8.447), confirming that the iterative refinement mechanism, not merely the structural complexity of the prompt, is responsible for the observed performance gains.

Table 3.5: AUDIT Prompt: Subject-Level Absolute Performance (All Variations)

Subject	AUDIT Prompt V3	AUDIT Prompt V2	Cumulative Thinking Prompt	AUDIT Prompt V1	Step-by-Step Prompt	Direct Prompt	Excellent-Professor-Aware Prompt
Abstract Algebra	9.416	9.306	8.646	8.709	8.626	8.580	8.497
Algebra	9.383	9.252	8.345	8.282	8.045	7.835	7.848
Calculus	9.339	9.111	7.965	7.975	7.986	7.743	7.714
Differential Geometry	9.485	9.320	8.741	8.558	8.633	8.522	8.528
Discrete Mathematics	9.421	9.304	8.610	8.559	8.447	8.250	8.247
Dynamical Systems	9.441	9.304	8.468	8.434	8.381	8.253	8.450
Geometry	9.375	9.173	8.012	8.388	8.132	7.955	7.934
Graph Theory	9.316	9.208	8.569	8.490	8.433	8.388	8.400
Information Theory	9.505	9.304	8.443	8.495	8.274	8.252	8.327
Linear Algebra	9.482	9.291	8.589	8.558	8.331	8.275	8.252
Logic	9.434	9.227	8.726	8.524	8.458	8.435	7.933
Measure Theory	9.412	9.279	8.708	8.649	8.627	8.419	8.510
Numerical Linear Algebra	9.315	9.179	8.342	8.316	8.383	8.240	8.217
Optimization	9.414	9.300	8.510	8.508	8.485	8.431	8.514
Probability	9.430	9.329	8.456	8.500	8.350	8.301	8.196
Statistics	9.315	9.159	8.186	8.248	7.900	7.864	7.794
Topology	9.334	9.253	8.636	8.651	8.787	8.639	8.619
Trigonometry	9.391	9.104	8.099	8.193	8.071	7.915	7.908

The subject-level relative performance analysis, with the excellent-professor-aware prompt normalized to 100% as the lowest-performing baseline, further validates the effectiveness and reliability of iterative reasoning embedded in the AUDIT prompt. AUDIT V3 achieves consistent relative performance improvements across all 18 subjects, ranging from 108.3% in Topology to 121.1% in Calculus, with an average relative performance of approximately 114.4%. AUDIT V2 demonstrates similarly consistent gains, spanning from 107.4% in Topology to 118.1% in Calculus. Notably, the Cumulative Thinking Prompt and AUDIT V1 exhibit moderate improvements ranging from 99.9% to 110.0% and 99.8% to 107.4% respectively, with several subjects showing marginal gains or slight underperformance. This consistent pattern across all subjects demonstrates that the iterative reasoning, audit, and refinement behaviors elicited by the AUDIT prompt successfully guide the language model to navigate and dynamically select optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ with higher efficiency and reliability than linear reasoning approaches. The baseline excellent-professor-aware prompt, which

relies solely on the simple and general "You are an excellent professor..." message in its prompt instructions $\mathcal{I}$ without structured reasoning guidance, consistently ranks lowest, underscoring the critical importance of explicit reasoning scaffolding, particularly iterative refinement, for achieving superior definition generation quality across diverse mathematical domains.

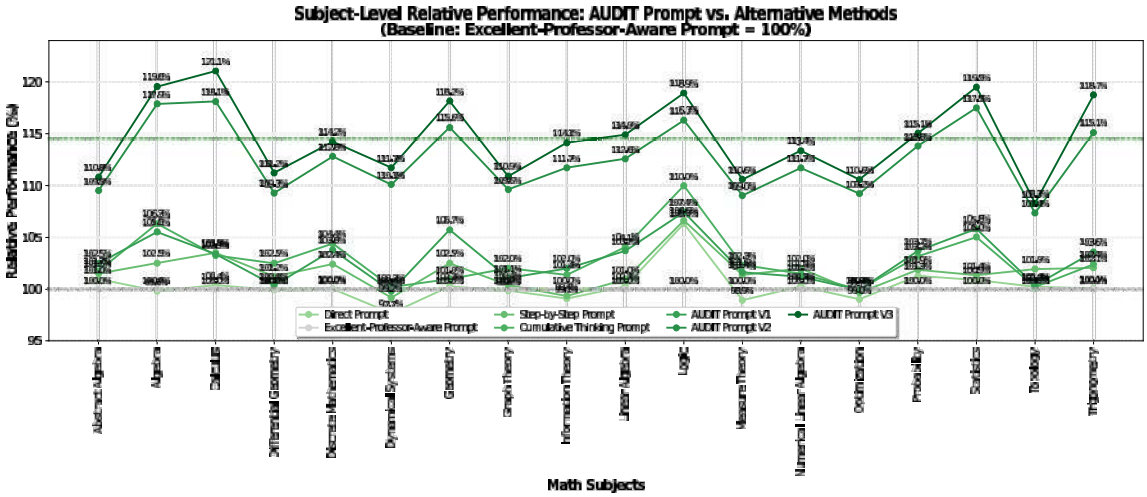


Figure 3.4: AUDIT Prompt: Subject-Level Relative Performance (All Vairations)

Table 3.6: AUDIT Prompt: Subject-Level Performance (All Vairations)

Subject	AUDIT Prompt V3	AUDIT Prompt V2	Cumulative Thinking Prompt	AUDIT Prompt V1	Step-by-Step Prompt	Direct Prompt	Excellent-Professor-Aware Prompt
Abstract Algebra	110.8%	109.5%	101.7%	102.5%	101.5%	101.0%	100.0%
Algebra	119.6%	117.9%	106.3%	105.5%	102.5%	99.8%	100.0%
Calculus	121.1%	118.1%	103.3%	103.4%	103.5%	100.4%	100.0%
Differential Geometry	111.2%	109.3%	102.5%	100.4%	101.2%	99.9%	100.0%
Discrete Mathematics	114.2%	112.8%	104.4%	103.8%	102.4%	100.0%	100.0%
Dynamical Systems	111.7%	110.1%	100.2%	99.8%	99.2%	97.7%	100.0%
Geometry	118.2%	115.6%	101.0%	105.7%	102.5%	100.3%	100.0%
Graph Theory	110.9%	109.6%	102.0%	101.1%	100.4%	99.9%	100.0%
Information Theory	114.1%	111.7%	101.4%	102.0%	99.4%	99.1%	100.0%
Linear Algebra	114.9%	112.6%	104.1%	103.7%	101.0%	100.3%	100.0%
Logic	118.9%	116.3%	110.0%	107.4%	106.6%	106.3%	100.0%
Measure Theory	110.6%	109.0%	102.3%	101.6%	101.4%	98.9%	100.0%
Numerical Linear Algebra	113.4%	111.7%	101.5%	101.2%	102.0%	100.3%	100.0%
Optimization	110.6%	109.2%	99.9%	99.9%	99.7%	99.0%	100.0%
Probability	115.1%	113.8%	103.2%	103.7%	101.9%	101.3%	100.0%
Statistics	119.5%	117.5%	105.0%	105.8%	101.4%	100.9%	100.0%
Topology	108.3%	107.4%	100.2%	100.4%	101.9%	100.2%	100.0%
Trigonometry	118.7%	115.1%	102.4%	103.6%	102.1%	100.1%	100.0%

3.4.3 AUDIT Prompt vs. Cumulative Thinking Prompt

The following section presents a focused subject-level performance comparison between the highest-performing AUDIT Prompt V3, the Cumulative Thinking Prompt from the first study, and the excellent-professor-aware prompt as the baseline normalized to 100%. This side-by-side analysis isolates the performance characteristics of iterative reasoning versus linear reasoning, providing clear evidence of how the multi-stage refinement mechanism of AUDIT Prompt translates into superior definition quality across individual mathematical subjects.

The subject-level absolute performance analytics demonstrate that AUDIT Prompt V3, which underwent two definition quality audits with improvement suggestions and two iterative refinements, is the unequivocal winner when compared against both the Cumulative Thinking Prompt and the excellent-professor-aware baseline. Across all 18 subjects, AUDIT V3 achieves absolute scores ranging from 9.315 to 9.505, while the Cumulative Thinking Prompt ranges from 7.965 to 8.741, and the baseline from 7.714 to 8.619. The pairwise differences reveal the magnitude of this performance lead: AUDIT V3 outperforms the Cumulative Thinking Prompt by margins ranging from 0.698 points in Topology to 1.374 points in Calculus, with an average advantage of 0.983 points across all subjects. Against the baseline excellent-professor prompt, the advantage is even more pronounced, with differences spanning from 0.715 points in Topology to 1.625 points in Calculus, averaging 1.179 points. Notably, the Cumulative Thinking Prompt itself shows modest advantages over the baseline, with differences ranging from -0.004 points in Optimization to 0.793 points in Logic, averaging 0.212 points. These patterns demonstrate that the iterative refinement mechanism embedded in AUDIT V3 yields substantially larger performance gains than linear reasoning alone, with the most pronounced advantages observed in subjects such as Calculus, Algebra, and Trigonometry, domains where formal precision and conceptual integration are particularly demanding.

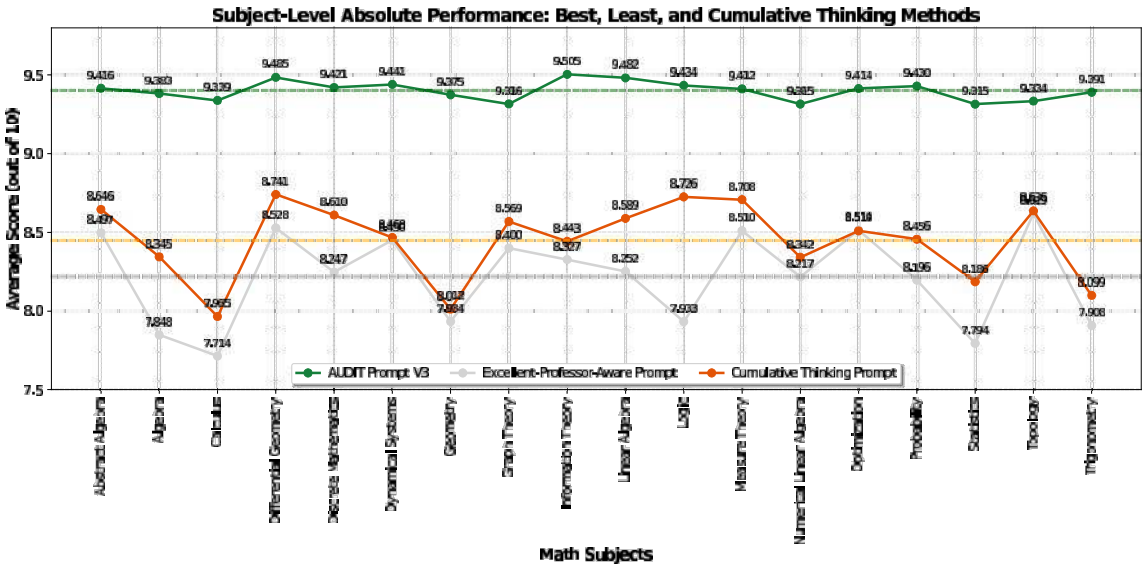


Figure 3.5: AUDIT Prompt: Subject-Level Absolute Performance (Selected Variations)

Table 3.7: AUDIT Prompt: Subject-Level Performance (Selected Variations)

Subject	AUDIT Prompt V3	Excellent-Professor-Aware Prompt	Cumulative Thinking Prompt
Abstract Algebra	9.416	8.497	8.646
Algebra	9.383	7.848	8.345
Calculus	9.339	7.714	7.965
Differential Geometry	9.485	8.528	8.741
Discrete Mathematics	9.421	8.247	8.610
Dynamical Systems	9.441	8.450	8.468
Geometry	9.375	7.934	8.012
Graph Theory	9.316	8.400	8.569
Information Theory	9.505	8.327	8.443
Linear Algebra	9.482	8.252	8.589
Logic	9.434	7.933	8.726
Measure Theory	9.412	8.510	8.708
Numerical Linear Algebra	9.315	8.217	8.342
Optimization	9.414	8.514	8.510
Probability	9.430	8.196	8.456
Statistics	9.315	7.794	8.186
Topology	9.334	8.619	8.636
Trigonometry	9.391	7.908	8.099

The subject-level relative performance comparison further reinforces the superiority of iterative reasoning, with AUDIT V3 achieving relative performance scores ranging from 108.3% in Topology to 121.1% in Calculus, all benchmarked against the excellent-professor-aware prompt at 100%. The performance differences between AUDIT V3 and the Cumulative Thinking Prompt reveal consistent double-digit gains across 11 out of 18 subjects, with advantages ranging from +8.1% in Topology to +17.8% in Calculus, and an average advantage of +11.7% across the 18 subjects. This pattern of consistent, substantial relative performance gains provides compelling evidence that the iterative

reasoning with dynamic generation, audit, and refinement mechanism of the AUDIT prompt enables the language model $M_{\text{a-gen}}$ to navigate the reasoning space $\mathfrak{R}$ toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ with far greater effectiveness than linear reasoning strategies. These findings serve as further evidence, extending the previously conducted performance comparison across all prompt methods, that iterative reasoning for mathematical term generation is indeed a more powerful and reliable method compared to linear reasoning approaches, with the capacity to consistently deliver superior definition quality across diverse mathematical domains.

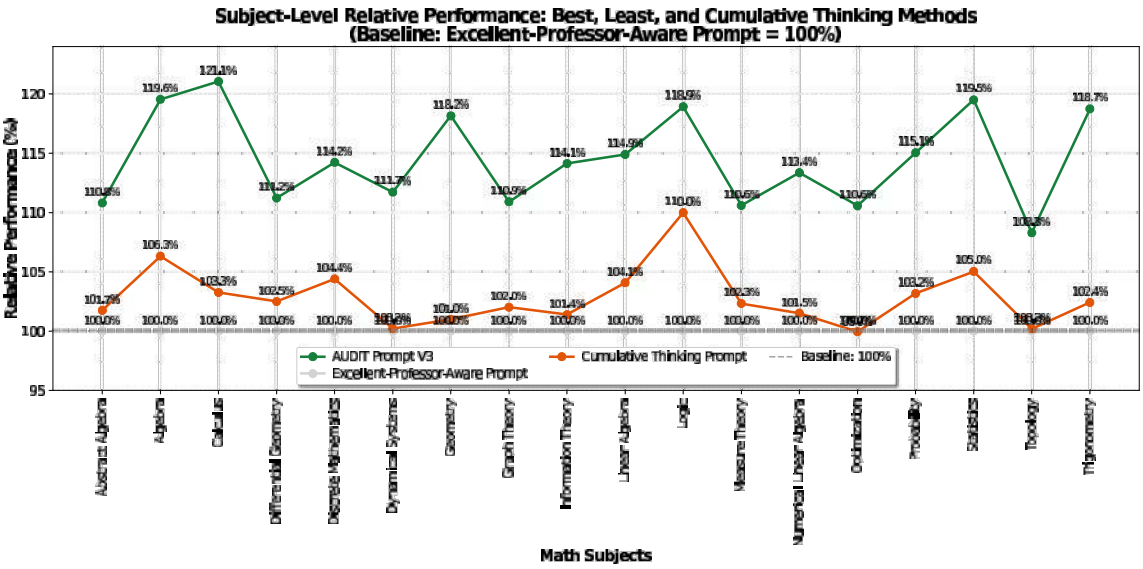


Figure 3.6: AUDIT Prompt: Subject-Level Relative Performance (Selected Variations)

Table 3.8: AUDIT Prompt: Subject-Level Relative Performance (Selected Variations)

Subject	AUDIT Prompt V3	Cumulative Thinking Prompt	Performance Difference	Excellent-Professor-Aware Prompt
Abstract Algebra	110.8%	101.7%	+9.1%	100.0%
Algebra	119.6%	106.3%	+13.3%	100.0%
Calculus	121.1%	103.3%	+17.8%	100.0%
Differential Geometry	111.2%	102.5%	+8.7%	100.0%
Discrete Mathematics	114.2%	104.4%	+9.8%	100.0%
Dynamical Systems	111.7%	100.2%	+11.5%	100.0%
Geometry	118.2%	101.0%	+17.2%	100.0%
Graph Theory	110.9%	102.0%	+8.9%	100.0%
Information Theory	114.1%	101.4%	+12.7%	100.0%
Linear Algebra	114.9%	104.1%	+10.8%	100.0%
Logic	118.9%	110.0%	+8.9%	100.0%
Measure Theory	110.6%	102.3%	+8.3%	100.0%
Numerical Linear Algebra	113.4%	101.5%	+11.9%	100.0%
Optimization	110.6%	99.9%	+10.7%	100.0%
Probability	115.1%	103.2%	+11.9%	100.0%
Statistics	119.5%	105.0%	+14.5%	100.0%
Topology	108.3%	100.2%	+8.1%	100.0%
Trigonometry	118.7%	102.4%	+16.3%	100.0%

Notes: Performance difference refers to the relative performance difference between the best-performing prompt variation, AUDIT Prompt V3, as an iterative reasoning prompt method, and the previously proposed linear reasoning prompt method, Cumulative Thinking Prompt. A higher positive difference indicates a greater advantage for the AUDIT Prompt.

3.4.4 Concrete Examples

Definition of Eigenvalue in Linear Algebra with AUDIT Prompt

Definition of eigenvalue in linear algebra with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the first paragraph as "definition_1", then suggest math definition refinement suggestions in the second paragraph as "quality_audit_1", then generate the first refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the third paragraph as "definition_2", then suggest math definition refinement suggestions to further refine the "definition_2" in the fourth paragraph as "quality_audit_2", then generate the second refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the fifth paragraph as "definition_3:". Make sure each of the sequential definitions should be optimized to be more strictly aligned with the standard math definitions of college level mathematics lecture notes and textbooks.

definition_1: In linear algebra, an eigenvalue of a square matrix $A \in \mathbb{C}^{n\times n}$ is a scalar $\lambda \in \mathbb{C}$ for which there exists a non-zero vector $\mathbf{v} \in \mathbb{C}^n$, called an eigenvector, such that the linear transformation defined by A acts on $\mathbf{v}$ by simple scalar multiplication. This relationship is expressed by the eigenvalue equation $A\mathbf{v} = \lambda\mathbf{v}$. Equivalently, λ is an eigenvalue if and only if the matrix $A - \lambda I_n$ is singular, meaning its determinant is zero: $\det(A - \lambda I_n) = 0$. This determinant condition defines the characteristic polynomial $p_A(\lambda) = \det(\lambda I_n - A)$, a monic polynomial of degree n whose roots are precisely the eigenvalues of A. The set of all eigenvalues is the spectrum $\sigma(A)$. Geometrically, eigenvectors indicate directions invariant under the transformation, stretched or compressed by the factor λ, which can be a real or complex number.

quality_audit_1: The definition correctly identifies the core equation and the determinant condition. However, it could be refined for stricter alignment with standard textbooks by: 1) Explicitly stating the domain (complex numbers) for both the matrix entries and eigenvalues from the outset to cover the general case. 2) More formally defining the eigenvector as belonging to the domain space $\mathbb{C}^n$ and being non-zero. 3) Clarifying that the characteristic polynomial is typically defined as $\det(\lambda I - A)$ to ensure it is monic. 4) Emphasizing that the eigenvalue equation implies a non-trivial solution to $(A - \lambda I)\mathbf{v} = \mathbf{0}$, which is the key to the singularity condition. 5) Avoiding the less common notation I_n in favor of I when dimension is clear.

definition_2: Let A be an $n \times n$ matrix with entries in $\mathbb{C}$. A complex number λ is called an eigenvalue of A if there exists a non-zero vector $\mathbf{v} \in \mathbb{C}^n$ such that $A\mathbf{v} = \lambda\mathbf{v}$. The vector $\mathbf{v}$ is an eigenvector corresponding to λ. This condition is equivalent to stating that the homogeneous linear system $(A - \lambda I)\mathbf{v} = \mathbf{0}$ has a non-trivial solution, which occurs if and only if the matrix $A - \lambda I$ is singular. Therefore, λ is an eigenvalue precisely when $\det(A - \lambda I) = 0$. The function $p(\lambda) = \det(\lambda I - A)$ is the characteristic polynomial of A, a monic polynomial of degree n. The fundamental theorem of algebra guarantees that $p(\lambda)$ has n roots in $\mathbb{C}$ counted with multiplicity, which constitute the spectrum of A. Eigenvalues reveal the scaling factors along invariant directions under the linear transformation $T(\mathbf{x}) = A\mathbf{x}$.

quality_audit_2: This definition is more precise. To further refine it to college-level textbook standards: 1) Specify that the eigenvector $\mathbf{v}$ must be in the vector space on which A acts (typically $\mathbb{C}^n$ or $\mathbb{R}^n$). 2) Explicitly connect the eigenvalue to the linear operator $T : V \to V$ represented by A, as eigenvalues are intrinsic to the operator, not just the matrix. 3) Clarify that the algebraic multiplicity of an eigenvalue is its multiplicity as a root of the characteristic polynomial. 4) Note that for real matrices, non-real eigenvalues occur in complex conjugate pairs. 5) Strengthen the final sentence by explicitly stating that an eigenvalue λ scales its eigenvector without changing its direction in the underlying vector space.

definition_3: Let V be a finite-dimensional vector space over $\mathbb{C}$ and let $T : V \to V$ be a linear operator. A scalar $\lambda \in \mathbb{C}$ is an eigenvalue of T if there exists a non-zero vector $\mathbf{v} \in V$, called an eigenvector of T, such that $T(\mathbf{v}) = \lambda\mathbf{v}$. If we choose a basis for V and represent T by an $n \times n$ matrix A, then λ is an eigenvalue of A if and only if $\det(\lambda I - A) = 0$. The polynomial $p_T(\lambda) = \det(\lambda I - A)$ is the characteristic polynomial of T; it is independent of the choice of basis. The eigenvalues are the roots of this monic degree-n polynomial. Each eigenvalue λ has an algebraic multiplicity, which is its multiplicity as a root of $p_T(\lambda)$, and a geometric multiplicity, which is the dimension of the eigenspace $E_\lambda = \{\mathbf{v} \in V : T(\mathbf{v}) = \lambda\mathbf{v}\}$. Geometrically, an eigenvector defines a one-dimensional invariant subspace of V on which T acts by scalar multiplication by λ.

Definition of Integer in Algebra with AUDIT Prompt

Definition of integer in algebra with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the first paragraph as "definition_1", then suggest math definition refinement suggestions in the second paragraph as "quality_audit_1", then generate the first refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the third paragraph as "definition_2", then suggest math definition refinement suggestions to further refine the "definition_2" in the fourth paragraph as "quality_audit_2", then generate the second refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the fifth paragraph as "definition_3:". Make sure each of the sequential definitions should be optimized to be more strictly aligned with the standard math definitions of college level mathematics lecture notes and textbooks.

definition_1: In algebra, an integer is an element of the set $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$, which extends the natural numbers $\mathbb{N}$ to include additive inverses. Formally, $\mathbb{Z}$ is constructed as the set of equivalence classes of ordered pairs of natural numbers (a, b) under the equivalence relation $(a, b) \sim (c, d)$ if and only if $a + d = b + c$, intuitively representing the solution to $a - b = c - d$. This construction yields the integers as the Grothendieck group of the natural numbers, ensuring every integer n has an inverse $-n$ such that $n + (-n) = 0$, where 0 is the additive identity. The set $\mathbb{Z}$ forms a commutative ring under standard addition and multiplication, denoted $(\mathbb{Z}, +, \cdot)$, which is an integral domain as it contains no zero divisors: if $ab = 0$ for $a, b \in \mathbb{Z}$, then $a = 0$ or $b = 0$. The integers are countably infinite and are ordered by the relation $\leq$, which is a total order compatible with the ring operations.
quality_audit_1: The definition correctly identifies integers as elements of $\mathbb{Z}$ and mentions their ring structure, but it is overly focused on the constructive equivalence class method, which, while foundational, is not the primary definition in introductory college algebra. It should more directly state the standard algebraic definition: the integers are the unique (up to isomorphism) ordered integral domain whose positive elements are well-ordered. The explanation of the Grothendieck group is too advanced for a first definition. The refinement should emphasize the axioms of an ordered ring: closure, associativity, commutativity, identities, inverses for addition, distributivity, and the total order properties. The link to natural numbers should be clarified as $\mathbb{Z}^+ \cup \{0\}$ being isomorphic to $\mathbb{N}$.
definition_2: In college-level algebra, the integers are defined axiomatically as the set $\mathbb{Z}$ that forms an ordered integral domain with unity, where the positive elements are well-ordered. Specifically, $(\mathbb{Z}, +, \cdot, \leq)$ satisfies: (1) $(\mathbb{Z}, +, \cdot)$ is a commutative ring with multiplicative identity 1, (2) it is an integral domain (i.e., if $ab = 0$ for $a, b \in \mathbb{Z}$, then $a = 0$ or $b = 0$), and (3) $\leq$ is a total order on $\mathbb{Z}$ compatible with the ring operations: if $a \leq b$, then $a + c \leq b + c$ for all $c \in \mathbb{Z}$, and if $a \leq b$ and $0 \leq c$, then $ac \leq bc$. The set of positive integers $\mathbb{Z}^+ = \{n \in \mathbb{Z} : n > 0\}$ is order-isomorphic to the natural numbers $\mathbb{N}$ and satisfies the well-ordering principle: every non-empty subset of $\mathbb{Z}^+$ has a least element. This structure is unique up to isomorphism and is the minimal ring extension of $\mathbb{N}$ that includes additive inverses, enabling solutions to equations of the form $a + x = b$ for all $a, b \in \mathbb{Z}$.
quality_audit_2: This definition is more aligned with standard algebraic presentations. However, it can be refined for greater precision and clarity. The phrase 'ordered integral domain with unity, where the positive elements are well-ordered' is correct but could be explicitly connected to the Peano axioms via the well-ordering principle. The uniqueness claim should be tempered: $\mathbb{Z}$ is the unique (up to isomorphism) ordered integral domain whose positive elements are well-ordered. The explanation of isomorphism to $\mathbb{N}$ should specify it as a semiring isomorphism. The final sentence about solving $a + x = b$ is important but should be framed as $\mathbb{Z}$ being the group of differences of $\mathbb{N}$. The definition should also explicitly state that $\mathbb{Z}$ contains a copy of $\mathbb{N}$ as $\mathbb{Z}^+ \cup \{0\}$.
definition_3: The integers are the unique (up to isomorphism) ordered integral domain $(\mathbb{Z}, +, \cdot, \leq)$ in which the set of positive elements $\mathbb{P} = \{x \in \mathbb{Z} : x > 0\}$ is well-ordered. This means $(\mathbb{Z}, +, \cdot)$ is a commutative ring with multiplicative identity $1 \neq 0$, has no zero divisors ($ab = 0 \implies a = 0$ or $b = 0$), and the total order $\leq$ satisfies: (i) if $a \leq b$ then $a + c \leq b + c$ for all $c \in \mathbb{Z}$, and (ii) if $a \leq b$ and $0 \leq c$ then $ac \leq bc$. The well-ordering of $\mathbb{P}$ implies every non-empty subset $S \subseteq \mathbb{P}$ has a least element under $\leq$, which is equivalent to the induction principle on $\mathbb{P}$. The structure embeds the natural numbers $\mathbb{N}$ via the monomorphism $\varphi : \mathbb{N} \to \mathbb{Z}$ with $\varphi(n) = \underbrace{1 + 1 + \cdots + 1}_{n \text{ times}}$, identifying $\mathbb{N}$ with $\mathbb{P} \cup \{0\}$. Algebraically, $\mathbb{Z}$ is the Grothendieck group (group of differences) of $\mathbb{N}$, ensuring every integer can be expressed as $a - b$ for $a, b \in \mathbb{N}$, and it is the initial object in the category of rings with unity.

3.5 Conclusion

3.5.1 Research Contributions

This study makes several significant contributions to the understanding and application of iterative reasoning for mathematical definition generation. First, it introduces the AUDIT prompt, a novel iterative reasoning framework $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\to} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\to} \iota_3)^{\delta_2}\}$ that operationalizes recursive quality assessment and refinement as integral components of the generation process. Unlike linear reasoning methods that produce a single output without intermediate feedback, the AUDIT prompt structures the task as a multi-stage cycle: initial definition generation, structured quality audit with refinement suggestions, refined definition generation, second quality audit, and final refined definition. This design represents a fundamental departure from conventional prompt strategies, treating definition generation not as a one-time navigation event but as a dynamic process where the language model $M_{\text{a-gen}}$ can revisit and refine its reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ based on explicit

self-assessment. The empirical validation demonstrates that this iterative mechanism yields substantial performance improvements, with AUDIT V3 achieving a holistic relative performance of 114.42% against the baseline, a gain of 14.42 percentage points, and outperforming the Cumulative Thinking Prompt by 11.61 percentage points, confirming that recursive refinement with explicit feedback loops enables superior navigation of the reasoning space $\mathfrak{R}$ compared to linear reasoning approaches.

Second, this research contributes a comprehensive comparative framework that isolates the distinct effects of iterative refinement stages on generation quality. By evaluating three AUDIT variants, V1 (single generation without refinement), V2 (one audit-refinement cycle), and V3 (two audit-refinement cycles), the study reveals a clear progression where additional refinement cycles yield diminishing but still meaningful improvements. AUDIT V3 achieves the highest performance (9.400) with the lowest standard deviation (0.058), indicating that two refinement cycles not only enhance quality but also improve consistency across diverse mathematical subjects. The finding that AUDIT V1 performs nearly identically to the Cumulative Thinking Prompt (8.446 vs. 8.447) confirms that the performance advantage of AUDIT V2 and V3 stems specifically from the iterative audit-refinement mechanism rather than from the structural complexity of the prompt itself. This granular analysis provides actionable insights for prompt design: the optimal number of refinement cycles depends on the balance between quality requirements and computational efficiency, with two cycles offering substantial gains at acceptable cost.

Third, this study establishes the reliability and generalizability of iterative reasoning for mathematical definition generation across 18 diverse mathematical subjects $S = \{s_1, s_2, \ldots, s_{18}\}$ with 921 terminologies T_{s_i}. The consistent performance advantages of AUDIT V3 across all subjects, from Algebra (119.6% relative performance) to Topology (108.3%), demonstrate that iterative refinement benefits are not confined to specific domains but represent a robust phenomenon across the mathematical spectrum. Moreover, the exceptionally low variability of AUDIT V3 (std = 0.058) compared to linear methods (std ranging from 0.179 to 0.283) indicates that iterative reasoning provides not only higher average quality but also greater stability, making it particularly valuable for applications requiring consistent, high-quality outputs across diverse inputs. Together, these contributions advance both the theoretical understanding of how iterative reasoning facilitates navigation of the reasoning space $\mathfrak{R}$ and the practical methodology for designing effective prompts for complex generation tasks.

3.5.2 Future Directions

Building upon the findings of this AUDIT prompt research, several promising future directions emerge for extending the iterative reasoning paradigm. The first direction concerns the optimization of refinement cycles through adaptive mechanisms. In the

current study, the number of audit-refinement cycles was fixed at two based on empirical observation. Future research could investigate dynamic termination criteria, where the language model $M_{\text{a-gen}}$ determines when to conclude the refinement process based on the quality of the current output or the marginal improvement observed between successive iterations. Such adaptive approaches could employ meta-evaluation mechanisms where the model assesses its own output against the evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$ and terminates refinement when quality thresholds are met or when diminishing returns indicate further iterations would yield minimal improvement. This would enable more efficient use of computational resources while maintaining high output quality, particularly for tasks where the optimal number of refinement cycles may vary depending on input complexity or subject domain.

A second promising direction involves expanding the iterative reasoning framework to incorporate multi-agent collaboration and diverse feedback sources. While the current AUDIT prompt relies on a single language model instance to both generate definitions and perform self-audits, future research could implement multi-agent architectures where distinct agents $\mathcal{A} = \{A_1, A_2, \ldots, A_K\}$ take on specialized roles. For example, a generator agent could produce initial definitions, a critic agent could perform structured audits, a refinement agent could incorporate feedback, and a consensus agent could adjudicate between competing refinements. Such distributed approaches could leverage the strengths of different model configurations or personas, such as research-focused agents for formal precision and teaching-focused agents for pedagogical clarity, to produce definitions that balance multiple quality dimensions simultaneously. Additionally, incorporating human feedback into the iterative cycle could create hybrid systems where domain experts provide targeted refinement suggestions that the language model then incorporates, combining the scalability of automated generation with the nuanced judgment of human specialists.

A third direction addresses the extension of iterative reasoning beyond definition generation to broader educational and technical applications. The AUDIT prompt's core mechanism, recursive generation, audit, and refinement, could be adapted for tasks such as mathematical proof construction, algorithm explanation, technical writing, and educational content development. For instance, in mathematical proof generation, iterative refinement could help models identify logical gaps or missing steps through self-audit, progressively building toward complete and rigorous proofs. Similarly, in educational settings, the audit component could be reframed as a pedagogical tool that generates not only refined content but also explanations of why refinements were necessary, providing learners with insights into the revision process itself. Future research could also explore domain-specific audit criteria tailored to different tasks, moving beyond the 11 evaluation metrics used in this study to incorporate discipline-specific quality standards. By extending the iterative reasoning framework to new domains and integrating it with emerging capabilities such

as multi-agent collaboration and adaptive refinement control, future work can further unlock the potential of large language models to generate high-quality, precisely calibrated outputs across the full spectrum of knowledge-intensive tasks.

Chapter 4

Reasoning Assumptions: What We Expect vs. What We Trust in Persona-Aware Prompt Engineering

4.1 Introduction

The Cumulative Thinking Prompt and AUDIT prompt studies established that both structured knowledge accumulation and iterative refinement can significantly enhance mathematical definition generation quality by guiding the language model $M_{\text{a-gen}}$ through deliberate reasoning navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D}) = \mathfrak{R}_{\mathcal{I},\mathcal{D}} \rightarrow \mathcal{R}_{\mathcal{I},\mathcal{D}} \rightarrow \rho_{\mathcal{I},\mathcal{D}}$ within the vast reasoning space $\mathfrak{R}$. However, both studies employed persona-based prompting, specifically the excellent-professor-aware prompt $p_{\text{e-pro}}$, as one of their baseline methods, yet the role of persona specificity itself remained underexplored. The excellent-professor prompt, with its coarse-grained "excellent professor" specification, represents only one point in a potentially rich spectrum of persona configurations. This raises a fundamental question: does the precision of persona specification matter, and if so, how do fine-grained distinctions in role description influence the language model's ability to navigate toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ for mathematical definition generation?

This study introduces the Professor-Rank-Aware Prompt Framework, which systematically investigates how fine-grained variations in professor rank specifications affect mathematical definition generation quality. Rather than relying on a single, general "excellent professor" persona, this research proposes eight distinct Professor-Rank-Aware Prompt variations $\mathcal{P}_{\text{profs}} = \{p_{(\text{profs},1)}, p_{(\text{profs},2)}, \ldots, p_{(\text{profs},8)}\}$, each specifying a distinct academic rank: distinguished professor, emeritus professor, full professor, research professor, associate professor, teaching professor, assistant professor, and lecturer. These variations are designed to test whether the explicit connotations embedded within each rank, such as academic seniority, research emphasis, teaching focus, or career stage, serve as effective

navigation signals that guide the language model toward reasoning subspaces containing more valuable mathematical information. The baseline excellent-professor-aware prompt $p_{\text{e-pro}} \in \mathcal{P}_{\text{base}}$ serves as the reference point for comparative analysis, allowing the study to determine whether fine-grained rank specifications yield measurable performance improvements or degradations relative to the commonly used but coarse-grained persona setting.

The evaluation employs the same extended dataset utilized in the AUDIT prompt research, comprising 921 mathematical terminologies $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ distributed across 18 mathematical subjects $S = \{s_1, s_2, \ldots, s_{18}\}$, with definitions assessed using the same 11 evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$ established in the first two studies. This consistency in experimental design ensures that performance comparisons across all three studies are methodologically aligned and reproducible. Through systematic comparative analysis of absolute and relative performance across the eight proposed variations and the baseline, this study seeks to answer three core questions: first, whether fine-grained professor-rank specifications yield significantly different definition quality compared to the coarse-grained excellent-professor baseline; second, whether a clear performance hierarchy emerges that correlates with academic seniority or role specialization; and third, whether the precision of persona context settings can serve as a reliable, training-free mechanism for optimizing language model outputs. By addressing these questions, the Professor-Rank-Aware Prompt research contributes to the broader understanding of how contextual cues within prompt instructions $\mathcal{I}$ shape reasoning navigation and offers practical guidance for selecting optimal persona configurations for educational and technical applications.

4.2 Related Works

The following section presents a systematic examination of representative prompt methodologies $\mathcal{P} = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$ that utilize language models with learned and reinforced cognitive capabilities as $\mathcal{M} = \{M_{(\text{a-gen})}, M_{(\text{a-eval})}, M_{(\text{m-eval})}, \pi_\theta\}$ to engage in persona-aware reasoning across diverse task domains, organized into two principal categories reflecting fundamental distinctions in how persona characteristics are operationalized within prompt structures. The nine prompting approaches $\mathcal{P}_{\text{cand}} \subseteq \mathcal{P}$ reviewed herein collectively demonstrate how the strategic assignment of identities, perspectives, and expertise profiles to language models fundamentally shapes their reasoning processes $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D}) = \mathfrak{R}_{\mathcal{I},\mathcal{D}} \to \mathcal{R}_{\mathcal{I},\mathcal{D}} \to \rho_{\mathcal{I},\mathcal{D}}$, knowledge activation patterns, and output characteristics. These methodologies span a spectrum from simple persona injections within single prompts to complex multi-agent frameworks $\mathcal{A} = \{A_1, A_2, \ldots, A_K\}$ where $A_k = (\mathcal{M}_k, \mathcal{C}_k, \mathcal{R}_k)$ representing multiple persona-bearing model instances that collaborate through structured interactions $\mathcal{S}_{\text{collab}} = \{s_1, s_2, \ldots, s_L\}$. Across both categories, the core insight unifying these approaches

is that persona-aware prompting functions as a cognitive steering mechanism, guiding models toward particular interpretive frameworks and reasoning subspaces $\mathcal{R}_u \subset \mathfrak{R}$ that would characterize a human possessing the specified persona $\mathcal{C}_k$. The first major category, "Persona-Aware Prompt with Role Settings," encompasses techniques that assign explicit social, professional, or demographic identities to language models, thereby framing task execution through the lens of particular role-based perspectives such as domain experts, imaginary readers with specific interests, or evaluative personas with distinct analytical orientations. The second major category, "Persona-Aware Prompt with Competency Settings," addresses methodologies that emphasize capability-based characterizations, directing models to adopt personas defined by particular skill sets, knowledge domains, or reasoning competencies rather than social identities per se, thereby framing task performance through the lens of specialized expertise and cognitive proficiencies.

Table 4.1: Persona-Rank-Aware Prompt: Citations of Persona-Aware Prompt Variations

Main Category	**Prompt Methods**
Persona-Aware Prompt with Role Settings	• Diverse Role-Player for Text Generation Evaluation (DRPE) [77] • Two-Stage Role-Play Prompting [78] • CoMM (Collaborative Multi-Agent, Multi-Reasoning-Path) [79] • Controlled Natural Language for Prompts (CNL-P) [80] • ChatEval [81] • Persona-aware Prompt Variation Evaluation [82]
Persona-Aware Prompt with Competency Settings	• Persona In-Context Learning (PICLe) [83] • Multi-Agent Debate (MAD) [84] • Solo Performance Prompting (SPP) [85]

4.2.1 Persona-Aware Prompt with Role Settings

The **Diverse Role-Player for Text Generation Evaluation (DRPE)** [77] is a zero-shot prompting framework designed to align automatic text evaluation more closely with human judgment. It operates by instructing a large language model through two distinct types of prompts. First, it uses a set of manually-crafted, fixed prompts that direct the model to act as an objective judge, focusing on specific, universally agreed-upon quality dimensions of a summary such as its grammar, fluency, and coherence with the original article. Second, and more distinctively, it employs model-generated prompts: for each new piece of text, the language model is asked to create a set of dynamic user profiles based on the content. This results in prompts that instruct the model to adopt the perspective of diverse, imaginary readers, such as a sports enthusiast, a legal expert, or a casual news consumer. These multiple role-player prompts then work in concert, each instructing the model to compare a pair of candidate summaries from its assigned perspective. The final evaluation aggregates these diverse votes, and the paper's experiments demonstrate that this multi-perspective prompting approach yields results that correlate with human evaluations (up to 90% on the CNN/DM dataset) far more effectively than traditional

metrics like BERTScore (55%) or even simple, direct prompts to the language model (58%).

The **Two-Stage Role-Play Prompting** [78] technique is a two-stage, zero-shot, text-based prompt method. In the first stage, the user creates a manually-crafted "role-setting prompt" to assign a specific persona to the language model, such as "you are an excellent math teacher." The model then generates multiple responses to this prompt, from which the researchers select the most immersive one to serve as the "role-feedback prompt." In the second stage, both the original role-setting prompt and the selected role-feedback prompt are combined with the actual reasoning question to form a single API call to the model, which then generates the final answer. The principal evidence supporting the effectiveness of this method is the method's consistent higher performance of the standard zero-shot baseline across a wide range of reasoning benchmarks. For instance, on the Last Letter Concatenation task, the role-play prompt method achieved an accuracy of 84.2%, which is a dramatic improvement over the 23.8% accuracy of the standard zero-shot approach, demonstrating the powerful effect of this two-stage character immersion on the model's ability to perform the required reasoning.

The **Collaborative Multi-Agent, Multi-Reasoning-Path (CoMM)** [79] prompt is a manually-crafted, few-shot text prompting framework that operates by assigning distinct expert roles to multiple instances of a large language model. In this design, the researchers manually write specific prompts to define each agent's identity, such as a "physicist" responsible for providing domain knowledge or a "mathematician" responsible for accurate calculations, along with their duties and guiding principles. Each agent is then given the same set of few-shot demonstration examples, but these examples are presented following different reasoning paths tailored to their specific expertise, for instance, the physicist receives examples solved through physics reasoning, while the mathematician receives the same problems solved through mathematical logic. The agents work collaboratively: the first expert provides its analysis, the second expert builds upon or corrects that analysis, and a third summarizer agent synthesizes both perspectives to produce a final answer. The core idea behind this design is to combine the complementary strengths of different expert perspectives, allowing one agent's deep knowledge to inform another agent's precise calculations, thereby overcoming individual agent limitations and achieving better results on complex reasoning tasks. The principal evidence supporting this method's effectiveness comes from the College Physics benchmark [23], where CoMM achieved 64.71% accuracy in the few-shot setting, significantly outperforming the standard prompting baseline [26] at 48.04% and the Chain-of-Thought at 56.86%.

Controlled Natural Language for Prompts (CNL-P) [80] is a way of writing prompt instructions for language models that combines the ease of everyday language

with the clarity of a computer program. It uses simple labels, like '[DEFINE_PERSONA:]' to set the AI's role, making it easy for anyone to write organized instructions while giving the AI clear markers to follow exactly. For example, instead of just saying "act like a travel guide," a CNL-P prompt would structure it as:

Persona Settings in CNL-P

```
[DEFINE_PERSONA:]
ROLE: You are a knowledgeable local tour guide specializing in hidden gems and cultural experiences.
Expertise: Historical sites, local cuisine recommendations, off-the-beaten-path attractions
Communication Style: Friendly, enthusiastic, and concise with practical tips
[END_PERSONA]
```

This structured approach has proven highly effective in practice. When evaluated across 93 prompts from the "Awesome ChatGPT Prompts" repository, CNL-P scored significantly higher than other template methods like RISEN [86] and RODES [87] in key areas like modularity and ease of modification. Testing across five different AI models as GPT-4o [25], Gemini-1.5-Pro-002 [88], GPT-4o-Mini [89], Llama3-70B-8192 [90], and Claude-3-Haiku [91], on six diverse classification tasks from the Natural Instructions dataset (including tasks on probability, mathematical reasoning, and title appropriateness) showed that CNL-P prompts were understood and executed just as well as regular natural language, with no loss in performance. Most notably, a specialized checking tool built for CNL-P successfully identified 100% of errors planted in a test dataset of 47 task instances, while standard GPT-4o [25] verification achieved only 76.60% accuracy, demonstrating that this structured format makes AI instructions not only clearer but also far more reliable and easier to debug.

ChatEval [81] is a manually-crafted, zero-shot text prompting framework that functions by orchestrating a structured debate among multiple large language model agents to evaluate text quality. Unlike a single model providing an immediate score, ChatEval first assigns each agent a unique persona, such as a "Critic" or a "Psychologist", through specific instructions in their initial prompt. This design is intended to force the agents to analyze the text from different angles, mirroring how a diverse group of human evaluators might assess something. The agents then engage in a discussion over several rounds, using one of several predefined "communication strategies" that dictate how they share and respond to each other's arguments. For example, in the "One-By-One" strategy, agents speak in a fixed order, building on the previous comments. The goal of this entire debate process is for the agents to reach a consensus or provide a final judgment that is more nuanced, balanced, and reliable than a single, isolated opinion. The paper's experiments provide principled evidence for this claim, showing that on the FairEval [92] benchmark, the ChatEval multi-agent variation with GPT-4 [25] achieved a 63.8% accuracy in aligning with human preferences. This was a notable improvement over

the 61.3% accuracy achieved by a single GPT-4 [25] agent using a standard, uniform prompt. Furthermore, the paper explicitly demonstrates the necessity of its core principle by showing that when all agents were given the same generic role prompt, the multi-agent system's performance dropped to the level of a single agent (53.8%), confirming that diverse role-playing instructions are essential for the framework's success.

Persona-Aware Prompt Variation Evaluation [82] is a systematic investigation into the specific prompt engineering technique known as "persona prompting," which involves adding a sentence to a prompt that assigns a particular role or identity to a large language model, such as "You are an expert in mathematics." The study rigorously examines how these seemingly simple additions to the prompt's wording influence the accuracy of the model's answers across a wide range of 27 different tasks, including factual question answering, reasoning, and mathematics. Rather than just observing whether personas have any effect, the researchers establish three specific criteria to evaluate the desirability of these effects: whether expert personas improve performance, whether irrelevant details like names cause any changes, and whether the model's performance aligns with logical expectations based on attributes like education level. They then test nine open source language models of varying sizes and families against these criteria, controlling for other factors. The principal evidence from this systematic evaluation reveals that while expert personas often help or at least do not harm performance, the models are surprisingly and undesirably sensitive to irrelevant details. For instance, the worst-case performance drop when adding an irrelevant name to the prompt was nearly 30% for some models, demonstrating that the specific wording of a prompt, even with task-irrelevant information, has a significant and often negative impact on language model behavior.

Algorithm 16 Pseudocode: Persona-Aware Prompt with Role Settings

Require: Input x, Prompt instructions $\mathcal{I}$, Set of agents $\mathcal{A} = \{A_1, A_2, \ldots, A_K\}$ where $A_k = (\mathcal{M}_k, \mathcal{C}_k, \mathcal{R}_k)$, Persona specifications $\mathcal{C}_{\text{persona}} \subseteq \mathcal{C}_k$ for each agent, Collaboration strategy $\mathcal{S}_{\text{collab}}$ (optional)
Ensure: Final output from persona-aware reasoning
Persona Assignment
Assign distinct persona $\mathcal{C}_{\text{persona}}$ to each agent A_k according to task requirements:
    - Expert roles (e.g., domain specialists)
    - Evaluator personas (e.g., critic, analyst)
    - User profiles (e.g., diverse reader perspectives)
Persona-Based Generation
for each agent $A_k \in \mathcal{A}$ **do**
    Configure A_k with assigned persona $\mathcal{C}_{\text{persona}}$ and instructions $\mathcal{I}$
    Generate output via A_k based on input x
end for
Multi-Persona Collaboration (if applicable)
if $\mathcal{S}_{\text{collab}}$ is defined **then**
    Enable agents to interact according to $\mathcal{S}_{\text{collab}}$:
        - Sequential discussion with feedback
        - Parallel deliberation and voting
        - Hierarchical synthesis of perspectives
    Aggregate outputs across collaboration rounds
end if
Output Synthesis
Synthesize final output from agent responses:
    - Majority voting across personas
    - Weighted aggregation based on role relevance
    - Consensus-based selection
return Final output

4.2.2 Persona-Aware Prompt with Competency Settings

Persona In-Context Learning (PICLe) [83] is defined as a few-shot text prompt method designed to elicit specific personal characteristics from a large language model. Unlike standard prompting that simply asks a question, PICLe constructs its prompt by first prepending a small set of carefully chosen example demonstrations that are highly indicative of the target competency, such as "narcissism" or "interest-in-literature." The method's core innovation is its scoring mechanism for selecting these examples. It computes a score for each candidate statement by measuring the difference between its likelihood under a model fine-tuned on the target competency and its likelihood under the original, unmodified language model. This score identifies statements that are very characteristic of the target competency but are not already well-predicted by the base model, deeming them the most "informative" to include. By providing these specific examples in the prompt, PICLe guides the model to infer the desired competency and align its subsequent answers accordingly. The effectiveness of this design is rigorously evaluated using a metric called action consistency, which measures the percentage of times the model's yes/no answers correctly match the ground truth for a given competency. The principal evidence from the paper shows that this approach yields substantial and consistent improvements over baseline prompt strategies. For instance, on the Llama-2 [47] model, PICLe achieves an average action consistency of 88.1%, which is a significant jump

from the base prompt method's score of 65.5% and also outperforms the best comparable few-shot baseline, which scored 84.6%.

The **Multi-Agent Debate (MAD)** [84] framework is a prompt-based method that uses manually-crafted, zero-shot text prompts to assign three distinct roles to large language model instances: affirmative debater, negative debater, and judge. The debater prompts include explicit "tit for tat" instructions that create a structured argumentation environment, such as "You are negative side. You disagree with the affirmative side's points. Provide your reasons and answer." This design forces the models to generate opposing viewpoints and counter-arguments rather than simply refining their own initial responses. The judge role is implemented through two types of prompts: a discriminative mode where the judge evaluates whether the correct solution has been reached after each debate round ("You are a moderator... At the end of each round, you will evaluate both sides' answers and decide which one is correct"), and an extractive mode where the judge extracts the final answer from the complete debate history if no consensus is reached within the iteration limit. This adaptive mechanism allows the framework to stop the debate at the optimal moment rather than continuing for a fixed number of rounds. The framework directly addresses the Degeneration-of-Thought problem by providing external feedback to each agent through the opposing debater's arguments, challenging rigid thinking patterns, and potentially correcting biased or distorted perceptions through exposure to alternative perspectives. The principal evidence supporting this framework comes from the commonsense machine translation task, where GPT-3.5-Turbo [26] with MAD achieved a human manual evaluation score of 3.78 on lexical ambiguity translations, significantly outperforming the baseline GPT-3.5-Turbo (3.43), Self-Reflect (3.43) and GPT-4 [25] (3.41). On the research reformulated Counter-Intuitive Arithmetic Reasoning task, MAD improved accuracy from 26.0% (baseline) to 37.0%, outperforming CoT [12] (28.0%), Self-Consistency [3] (29.5%), and Self-Reflect [93], [94] (27.5%), demonstrating that the structured debate prompts effectively enable models to overcome intuitive but incorrect responses through divergent thinking.

Solo Performance Prompting (SPP) [85] is a manually-crafted text prompt designed to transform a single large language model into a "cognitive synergist." The prompt works by instructing the model to follow a specific three-step procedure for any given task. First, in the "Persona Identification" phase, the model must generate a list of relevant expert participants, such as a "Film Expert" or "Sports Enthusiast," based on the task input. Second, during the "Brainstorming" phase, these identified personas share their knowledge and suggestions on how to approach the task from their unique perspectives. Third, in the "Multi-Persona Iterative Collaboration" phase, a leader persona (the AI Assistant) proposes an initial solution, then consults each of the other personas for feedback and revision suggestions. This process can repeat multiple times

until all participants are satisfied, at which point a final answer is produced. SPP is applied as a few-shot prompt with two demonstration examples that show the expected collaborative procedure but without solved examples of the specific evaluation tasks. The principal evidence from the paper supporting this definition comes from the Trivia Creative Writing task. On the more challenging version requiring ten facts, SPP achieved an average score of 84.7%, significantly outperforming both the standard prompt at 77.0% and the Chain-of-Thought [12] prompt at 68.5%. This demonstrates that the multi-persona collaboration prompted by SPP effectively helps the model produce more factually correct answers, directly addressing the problem of hallucination without relying on any external tools or running multiple separate model instances.

Algorithm 17 Pseudocode: Persona-Aware Prompt with Competency Settings

Require: Input x, Prompt instructions $\mathcal{I}$, Target LLM $M_{\text{a-gen}} \in \mathcal{M}$ for generation, Persona configuration $\mathcal{C}_{\text{persona}} = (\mathcal{P}_{\text{role}}, \mathcal{P}_{\text{comp}})$, Competency settings $\mathcal{R}_{\text{comp}}$, Collaboration mode $\mathcal{S}_{\text{collab}}$ (optional)
Ensure: Output aligned with target persona and competency
Persona Identification
Identify relevant personas according to $\mathcal{C}_{\text{persona}}$:
    - Expert roles based on task domain
    - Competency characteristics
    - Behavioral attributes
Competency Configuration
Configure competency settings $\mathcal{R}_{\text{comp}}$:
    - Select informative demonstrations indicative of target competency
    - Define role-specific instructions and constraints
    - Set behavioral objectives for each persona
Persona-Driven Generation
if $\mathcal{S}_{\text{collab}}$ = Multi-Persona **then**
    Generate outputs from multiple personas in parallel or sequentially
    Facilitate interaction between personas (e.g., debate, feedback, refinement)
    Aggregate and synthesize responses across personas
else
    Adopt target persona with competency settings
    Generate output via $M_{\text{a-gen}}$ with persona-aware prompt
end if
Output Validation
Evaluate output against competency requirements:
    - Consistency with target persona
    - Alignment with competency objectives
    - Quality of generated content
return Final persona-aligned output

4.3 Persona-Aware Reasoning Assumptions

The preceding review of persona-aware prompt methods has shown how researchers assign different roles and identities to language models $\mathcal{M} = \{M_{(\text{a-gen})}, M_{(\text{a-eval})}, M_{(\text{m-eval})}, \pi_\theta\}$, asking them to act as experts, judges, or specific types of roles $\mathcal{C}_k$ within agent configurations $A_k = (\mathcal{M}_k, \mathcal{C}_k, \mathcal{R}_k)$ to improve the quality of generated responses. While these approaches have demonstrated promising results across various tasks, they all rest on a basic assumption that deserves closer examination: the idea that simply telling a model to

adopt a persona $\mathcal{C}_k$ without fine-grained context, such as "you are an excellent professor," will actually make it think and respond like that persona would. This assumption is often taken for granted in research papers, sometimes stated directly but more often implied as a given. However, there is no automatic guarantee that adding a persona description $\mathcal{C}_k$ to a prompt $p \in \mathcal{P}$ truly changes how the model processes information or that the outputs genuinely reflect the expertise and perspective of the assigned role. Before we can confidently rely on these methods, we need to check whether the performance results reported in studies actually prove that the persona prompts $p_{\text{e-pro}} \in \mathcal{P}$ are working as intended, or whether we are simply assuming they work without solid evidence.

Examining these underlying assumptions serves two clear purposes. First, it helps us understand what each research paper actually claims about how persona prompts influence model behavior $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D}) = \mathfrak{R}_{\mathcal{I},\mathcal{D}} \to \mathcal{R}_{\mathcal{I},\mathcal{D}} \to \rho_{\mathcal{I},\mathcal{D}}$, whether those claims are stated openly or just hinted at, and most importantly, whether the authors provide real evidence to back up those claims. Some studies may carefully test whether their persona assignments truly change how the model reasons within reasoning subspaces $\mathcal{R}_u \subset \mathfrak{R}$ and reasoning directions $\rho_{(u,w)} \in \mathcal{R}_u$, while others may simply assume the persona is doing its job. Second, this analysis creates a foundation for designing better, more specific persona prompts $p_{(\text{profs},k)} \subseteq \mathcal{P}_{\text{profs}}$ for particular tasks. The common practice of using simple, broad persona descriptions $\mathcal{C}_k$ like "you are an excellent math professor" may not be the most effective approach, and understanding what assumptions have been proven true versus what remains unverified can guide us toward creating more detailed and task-appropriate persona settings. This matters because if we want to generate high-quality mathematical definitions $g^{(p)}_{(i,j)}$ for terms $t_{(i,j)} \in T_{s_i}$ in subjects $s_i \in S$, we need to know what kind of professor persona $\mathcal{C}_k$ actually helps the model $M_{\text{a-gen}}$ perform best, rather than just defaulting to whatever role description happens to be most common in the research literature.

Table 4.2: Persona-Rank-Aware Prompt: Citations of Persona-Aware Prompt Reasoning Assumptions

Assumption Category	Explanation	Supporting Citations and Prompt Titles
Prompt structure shapes model reasoning pathways	The organization and format of a prompt influences how a language model processes information and arrives at responses. Well-structured prompts with clear components, logical flow, and explicit guidance can steer model reasoning more effectively than unstructured instructions. Models are capable of following structured formats that resemble systematic task decomposition.	• Two-Stage Role-Play Prompting [78] • Controlled Natural Language for Prompt (CNL-P) [80] • Cue-Based Chain-of-Thoughts (Cue-CoT) [8] • Solo Performance Prompting (SPP) [85]
Persona assignments influence how models access and apply knowledge	Assigning specific roles or identities to language models directs their attention toward particular knowledge domains and reasoning perspectives. Different personas activate different subsets of the model's learned information, shaping how it interprets tasks and what knowledge it draws upon. More detailed and contextually relevant role descriptions produce stronger effects on model behavior.	• ChatEval [81] • CoMM [79] • Two-Stage Role-Play Prompting [78] • Solo Performance Prompting (SPP) [85] • Persona In-Context Learning (PICLe) [83] • Principled Persona Prompt Evaluation [82]
Reasoning behaviors emerge from how models navigate their internal knowledge space	Language models explore their internal knowledge along pathways that can be guided by prompt design. Without guidance, models tend to follow narrow or biased reasoning trajectories. Effective prompts can direct models toward more thorough exploration of relevant knowledge, structured thinking steps, and consideration of multiple angles before arriving at conclusions.	• Cue-Based Chain-of-Thoughts (Cue-CoT) [8] • Two-Stage Role-Play Prompting [78] • Principled Persona Prompt Evaluation [82] • Multi-Agent Debate (MAD) [84] • Multiagent Debate [59]
Multi-agent collaboration enables collective reasoning beyond individual capabilities	Deploying multiple model instances with diverse perspectives allows for exploration of complementary reasoning paths. Through structured interaction involving critique, debate, and consensus-building, these agents can identify and correct each other's errors, overcome individual blind spots, and converge on outcomes that surpass what any single agent could produce. The collective process leverages diversity of thought to achieve more robust reasoning.	• ChatEval [81] • Diverse Role-Player for Text Generation Evaluation (DRPE) [77] • CoMM [79] • Multi-Agent Debate (MAD) [84] • Multiagent Debate [59] • Solo Performance Prompting (SPP) [85]

The **ChatEval** [81] framework is predicated on a set of interconnected reasoning assumptions about how large language models navigate their internal reasoning space, which together form the theoretical foundation for employing multi-agent debate in text evaluation tasks. These assumptions range from identifying the inherent limitations of single-agent reasoning to proposing mechanisms by which multiple agents can synergistically overcome those limitations to achieve superior outcomes. The first assumption, that an LLM, despite its vast parametric knowledge, tends to explore a limited and potentially biased region of its overall reasoning space when activated by a simple, single-perspective prompt, is implicitly proposed in the paper's critique of existing single-agent evaluation methods. The authors explicitly identify a performance gap between such methods and

human-level evaluation, and reference the "Degeneration-of-Thought" problem, thereby validating the necessity of moving beyond a singular, constrained reasoning pathway. The second assumption, that an LLM can adopt and maintain different roles when assigned specifically designed persona-aware prompts across multiple instances of the same underlying model, is both explicitly proposed and empirically verified. The paper details its "Diverse Role Specification" methodology, where each agent receives a unique 'role_description' such as "You are now Educator, one of the referees in this task. You will assess how well the responses explain concepts and whether they would be understandable to someone learning the material for the first time" or "You are now Domain Expert, one of the referees in this task. You possess specialized knowledge in the relevant field and will evaluate the technical accuracy, depth, and correctness of the information presented". The assumption is crucially verified through experimentation showing that using identical role prompts for all agents leads to a significant drop in accuracy from 60.0% to 53.8% and Kappa from 0.33 to 0.25 (Table 3), proving that the models are indeed capable of internalizing and sustaining distinct personas that are essential for performance gains. Building upon this, the third assumption, that a persona-assigned LLM instance can effectively explore a unique reasoning subspace and direction that differs from those explored by other persona-assigned instances, is implicitly proposed as the mechanism through which diversity enriches the collective evaluation process. This is qualitatively demonstrated in the paper's debate example, where agents focus on different evaluative dimensions such as detail, conciseness, and contextual applicability, thereby navigating distinct subspaces of the overall reasoning space. Finally, the fourth assumption, that collaboration among these diversely-oriented LLM instances can steer the collective reasoning toward a more optimal and human-aligned solution direction, is explicitly proposed as the core hypothesis of "Collective Intelligence" and "Cognitive Synergy." This assumption is rigorously verified through quantitative results, with the multi-agent ChatEval achieving an accuracy of 60.0% (Kap. 0.33) compared to the single-agent ChatGPT baseline of 53.8% (Kap. 0.27) on the FairEval [92] benchmark, and for GPT-4, improving accuracy from 61.3% to 63.8% (Table 1). On the Topical-Chat [95] benchmark, the multi-agent approach improved GPT-4's average Spearman correlation from 0.658 to 0.684 and Kendall-Tau correlation from 0.611 to 0.632 (Table 2), confirming that structured debate guides the system toward final judgments that better align with human preferences.

The **Diverse Role-Player for Text Generation Evaluation (DRPE)** [77] framework is defined as a comparison-based metric that leverages Large Language Models (LLMs) to evaluate generated text by having them adopt multiple, distinct roles, both static objective judges (e.g., for fluency, coherence) and dynamically generated subjective personas (e.g., specific types of readers), to vote on which of two summaries is superior, with the results then aggregated into a final score. This framework operates on the premise that an LLM's parametric knowledge inherently contains diverse human perspectives

and evaluative competencies, which can be reliably accessed through carefully structured prompts. By designing persona-aware prompts, the framework guides the LLM into specific reasoning subspaces focused on particular quality dimensions relevant to each role, while the prompt structure itself steers the model's reasoning direction toward a comparative judgment rather than an absolute score. The effectiveness of these underlying principles is strongly supported by empirical evidence: the DRPEScore achieved a Pearson correlation with human annotations of 0.90 on CNN/DailyMail and 0.87 on XSum, substantially outperforming both traditional metrics like BERTScore (0.55, 0.57) and single-perspective LLMScore (0.58, 0.62). This numerical validation demonstrates that accessing diverse perspectives through structured role-playing and aggregating multiple comparative judgments produces an evaluation that closely mirrors the complexity and nuance of human assessment.

Solo Performance Prompting (SPP) [85] are built upon how large language models store and access knowledge and how their reasoning processes can be guided through prompt structure. First, the research assumes that a sufficiently capable model like GPT-4 possesses vast parametric knowledge but that a standard, single-perspective prompt fails to reliably access this knowledge, leading to factual hallucinations as observed in tasks requiring the integration of information from diverse domains (p. 5, Table 2). Second, it assumes that by instructing the model to adopt multiple fine-grained expert personas, such as "Film Expert," "Sports Enthusiast," or "Music Enthusiast", the prompt effectively partitions the model's reasoning process into distinct perspectives, each specialized for a particular domain of knowledge (p. 9, Figure 13). This structure allows the model to navigate to the correct factual information more reliably, a mechanism implicitly verified by the qualitative observation that dynamic, fine-grained personas outperform fixed, general personas like "Expert" (p. 8, Figure 8). Third, the iterative collaboration mechanism assumes that a reasoning path involving critique and revision from multiple perspectives, formalized through brainstorming, initial solution generation, and multi-turn feedback loops, leads to a more accurate final output than a single, linear reasoning path such as that generated by Chain-of-Thought prompting. This also assumes that a model can effectively simulate multiple agents through a structured, multi-turn dialogue process wherein the initial reasoning is implemented by a lead "AI Assistant" persona, critique is generated by different dynamically identified expert personas (e.g., "Computer Scientist," "Ten year old child"), and the lead persona then revises its own work based on that feedback, meaning feedback and revision are performed by different simulated roles with distinct perspectives, as operationalized in the SPP prompt's three-phase procedure of Persona Identification, Brainstorming, and Multi-Persona Iterative Collaboration (p. 3-4). This role differentiation is empirically verified by comparing SPP against Self-Refine, a single-agent iterative refinement method where the same model both proposes and critiques without distinct personas; SPP's superior performance (e.g., 79.9% vs. 73.9% accuracy on

research proposed Trivia Creative Writing, Table 2) demonstrates that the value derives not merely from iteration, but from the structured integration of multiple persona-driven perspectives with clearly differentiated roles, where feedback and revision emerge from different simulated perspectives, within a single model. However, it failed to do so on less capable models like GPT-3.5-Turbo [26] and Llama2-13B-Chat [47], where performance either showed no improvement or degraded due to issues like early-termination (p. 7, Figure 6; p. 14, Table 4). This suggests that the underlying assumptions about accessing parametric knowledge and simulating multi-perspective collaboration with differentiated roles are valid primarily when a language model reaches a certain threshold of capability, drawing an analogy to human cognitive development where complex role-playing emerges only at a certain stage.

The principal assumption underlying **Persona In-Context Learning (PICLe)** [83] is that a large language model's behavior can be understood as drawing from a mixture of different "persona subspaces" within its overall reasoning space. The paper explicitly assumes that the model's output probability for any given prompt is the result of a weighted combination of these latent personas. Based on this, the core mechanism assumes that by carefully designing the prompt, one can steer the model's "reasoning direction" toward a specific target persona. This is achieved by selecting demonstrations that maximize the posterior probability of that target persona, effectively guiding the model's "reasoning path search" toward a particular subspace. This selection process relies on a comparison between the model's existing "parametric knowledge" (represented by the original model's likelihood of a statement) and the knowledge signature of the target persona (represented by a fine-tuned model's likelihood). Therefore, the key assumptions are that persona-specific subspaces exist within the model, that prompts can shift the model's inference toward a desired subspace, and that the most effective prompts for this shift are those containing examples where the target persona's influence represents a significant change from the model's default behavior. The conclusive evidence supporting these assumptions is the method's consistent success across three different language models, Llama-2 [47], Vicuna, and GPT-J, with PICLe achieving the highest action consistency scores on all of them, including a significant improvement from 65.5% to 88.1% on Llama-2 [47], demonstrating that the prompt design based on these assumptions effectively alters model behavior in the intended direction.

The **CoMM** [79] framework is built on several principal assumptions about language model behaviors and prompt design. First, it assumes that a single instance of a language model has a limited reasoning subspace, meaning its thinking is confined to a single path and is prone to specific types of errors, such as mistakes in domain knowledge or in calculation. Second, it assumes that different expert roles can elicit different subspaces of the model's parametric knowledge; prompting a model to act as a physicist draws

out its knowledge of physical laws, while prompting it as a mathematician activates its computational abilities. Third, it assumes that these different reasoning paths can be combined through collaboration, where the output of one agent influences the reasoning direction of another, allowing the team to search for a correct solution more effectively than an individual agent. A further critical assumption is that providing few-shot demonstrations with distinct reasoning paths for each role is essential to guide the agents toward their specialized ways of thinking. The paper implicitly assumes that this structured, multi-agent prompt design can overcome the self-consistency bias of a single model, which tends to stay on its initial reasoning path even if it is wrong. These assumptions are proved by the conclusive evidence that CoMM significantly outperforms single-agent methods. As shown in Table 2, on the College Physics [23] benchmark, a single agent approach scored only 42.16% in zero-shot and 56.86% in few-shot settings, while the multi-agent CoMM achieved 54.90% and 64.71% respectively, demonstrating that multiple independent agents collaborating are necessary to access complementary reasoning subspaces and correct individual errors.

The **Two-Stage Role-Play Prompting** [78] technique operates on several principal assumptions about how language models behave and how prompts can influence them. First, it assumes that language model behaviors can be deliberately shaped through prompt design and prompt structure, specifically that assigning a character creates a mental framework or "subspace" within the model's reasoning space that guides its response generation. The authors assume that a more deeply immersed role leads the model to access and utilize its parametric knowledge, the information stored during training, in a way that is more aligned with the task, effectively steering the reasoning direction toward a more productive path. Second, it assumes that this role-playing effect can be amplified by the structure of the interaction; a two-stage dialogue, where the model first acknowledges and elaborates on its role, provides a stronger contextual anchor than a single-turn prompt. This deeper immersion is believed to make the subsequent reasoning path search more deliberate and structured, often resulting in a clearer step-by-step explanation, even though the prompt does not explicitly command the model to "think step by step." The principal evidence for these assumptions is found in the ablation study on prompt structure. When the model was cast as a math teacher using a single-round prompt (just the role assignment), it achieved 57.1% accuracy on AQuA. Adding more descriptive details to the single-round prompt increased accuracy to 60.2%. However, when the two-stage structure with a role-feedback prompt was introduced to deepen immersion, accuracy further rose to 63.8%. This progressive improvement, directly tied to the increasing depth of role immersion through prompt design, validates the core assumption that carefully crafted character assignments and multi-turn prompt structures can steer the model's reasoning capabilities more effectively.

The principal assumption underlying the **Controlled Natural Language for Prompt (CNL-P)** [80] research is that the inherent ambiguity of natural language is the primary obstacle to reliable and predictable behavior from large language models, and that this can be overcome by applying software engineering principles to prompt design and prompt structure. The researchers assumed that by imposing a precise, modular structure on a prompt, with clear sections for persona, constraints, variables, and a defined workflow, they could guide the model's reasoning path more effectively, essentially creating a constrained reasoning subspace that leads to more consistent outputs. They further assumed that this structured format would not hinder, but rather be intuitively understood by, the models' underlying parametric knowledge and learned language model behaviors, allowing them to execute complex tasks in a zero-shot setting without needing to be retrained or given examples. This is fundamentally an assumption about how LLMs process instructions: that they can benefit from a clear, almost programmatic, decomposition of a task. The conclusive evidence supporting this assumption is provided in the experiment for Research Question 2, where across five different language models (GPT-4o [25], Gemini-1.5-Pro, GPT-4o-Mini [25], Llama3-70B, and Claude-3-Haiku) and six varied tasks, CNL-P prompts achieved performance comparable to or better than standard natural language prompts. This demonstrates that the structured design effectively communicates the user's intent within the model's existing reasoning capabilities.

The **Principled Persona Prompt Evaluation** [82] explicitly proposes three normative reasoning assumptions, termed "desiderata", about how personas should guide a language model's traversal of its holistic reasoning space. The first explicit assumption, "Expertise Advantage," posits that a persona with task-aligned expertise should direct the model into a reasoning subspace that yields a reasoning direction with performance at least on par with the non-persona baseline. This is supported by numerical evidence in Figure 2, where expert personas lead to non-negative performance changes in 78% to 100% of tasks across models. The second explicit assumption, "Robustness," holds that task-irrelevant attributes (like names or colors) should not influence the model's chosen reasoning direction; the model should remain within the appropriate reasoning subspace regardless of such superficial variations. However, Figure 3 shows this assumption is violated, with irrelevant personas causing significant performance drops in up to 59% of tasks. The third explicit assumption, "Fidelity," asserts that personas specifying relevant attributes (e.g., education level) should shape the reasoning direction in ways consistent with those attributes, effectively ordering performance along an expected hierarchy within that reasoning subspace. This is partially confirmed by the mixed-effects model in Section 5.4, which shows significant positive correlations (0.2 to 0.8 percentage points) between attribute levels and task accuracy, though Figure 4 reveals that such fidelity is not consistently observed across all tasks or model families.

The **Cue-Based Chain-of-Thoughts (Cue-CoT)** [8] method is built upon three foundational reasoning assumptions about how large language models process dialogue for response generation. The first assumption posits that LLMs inherently possess sufficient comprehension of human behavioral patterns to accurately infer implicit or explicit user states, such as personality traits, emotional conditions, and psychological characteristics, directly from conversational text. This is evidenced by the model's ability to generate meaningful intermediate status descriptions across diverse linguistic contexts. The second assumption holds that carefully crafted prompts can steer the model's cognitive processing toward more productive analytical pathways, effectively constraining its generative focus to relevant user attributes rather than producing generic responses. The third assumption suggests that decomposing the reasoning process into sequential steps enables clearer cognitive focus, allowing the model to dedicate attention to each component task individually and consequently produce more nuanced intermediate outputs that enhance final response quality. Supporting numerical evidence from the study demonstrates that O-Cue CoT achieves above 50% win rates across most datasets and models compared to standard prompting, validating the prompt design assumption. Furthermore, M-Cue CoT consistently outperforms both standard prompting and O-Cue CoT across all six datasets and five LLMs under both zero-shot and one-shot settings, with win rates frequently exceeding 80% on datasets like Zhihu and ED, thereby confirming the effectiveness of multi-step reasoning. However, non-supporting evidence emerges in specific cases where certain models struggle with complex instructions, such as ChatGPT showing poorer performance with O-Cue CoT on Chinese datasets due to instruction-following limitations, and Alpaca exhibiting ambiguous outputs under O-Cue CoT in English datasets, indicating that the effectiveness of these assumptions is partially model-dependent and sensitive to instruction complexity.

The **Multi-Agent Debate (MAD)** [84] framework operates on several principal assumptions about language model behaviors and how prompt design can influence them. First, it assumes that a single language model, when operating alone, has a limited reasoning subspace it can explore due to its parametric knowledge being filtered through initial biases, making it prone to getting stuck in a local, incorrect reasoning path (the Degeneration-of-Thought problem). Second, it assumes that by using a specific prompt structure that assigns adversarial roles ("tit for tat"), multiple instances of the same model can be forced into different reasoning directions, effectively expanding the collective reasoning subspace. Third, it assumes that the model's parametric knowledge is sufficient to generate correct answers, but the challenge lies in the reasoning path search to access that knowledge correctly; external debate provides the necessary search mechanism. Fourth, it assumes that a judge model, guided by an adaptive prompt, can reliably evaluate the divergent paths and select the correct one. The conclusive evidence supporting these assumptions is that on the Counter-Intuitive Arithmetic Reasoning

dataset, GPT-3.5-Turbo's [26] accuracy improved from 26.0% with a standard prompt to 37.0% with the MAD framework, proving that the multi-agent prompt structure successfully guided the models to search a broader and more accurate reasoning path than they could find on their own, despite using the same underlying parametric knowledge.

The **Multiagent Debate** [59] framework operates on several principal assumptions about language models and their interaction with prompt structures. First, it assumes that language model behaviors are such that individual instances of the same model, when given the same prompt, will generate a diverse range of answers, effectively exploring different parts of the reasoning space. The paper observes that "given an initial query... individual model instances propose a diverse range of answers despite being the same model class." This diversity is crucial, as it creates multiple reasoning paths for the debate to consider. Second, the method assumes that a model's learned parametric knowledge can be more reliably accessed and validated through a process of social consensus. The framework is built on the premise that when a model is uncertain, it is more easily persuaded by other agents, leading to a convergence on a more accurate answer. This is linked to the idea of reasoning direction, where the debate prompts guide the models away from incorrect paths and toward a consistent, correct one. Third, the prompt design itself is based on the assumption that a prompt structure which asks a model to consider the responses of others will activate its internal capabilities for critique and verification, a behavior not naturally triggered by a standard, single-instance prompt. This is evident in the use of the consensus prompt: "Using the solutions from other agents as additional advice, can you provide an updated answer?" which is designed to steer the model's reasoning path search toward a more collaborative and self-correcting process. Finally, an implicit but principal assumption is that the language models themselves possess the inherent ability to act as reliable critics of their own and others' outputs, a capability that is harnessed through the multi-turn debate structure. The core assumption that debate leads to more accurate convergence is strongly supported by the quantitative results in Table 1, where the Multi-Agent (Debate) method with 3 instances in two rounds of debating achieves an accuracy of $85.0 \pm 3.5\%$ on GSM8K Grade School Math split [22], significantly outperforming single-agent methods. Furthermore, the paper directly tests the assumption about model uncertainty and persuasion in Section 3.2, finding that "on facts that the language model was confident in... it was very difficult to convince an agent to change their opinion," while uncertain facts lead to consensus on a more accurate answer, as illustrated in Figure 9. This confirms that the debate mechanism effectively leverages model behavior to navigate the reasoning space toward greater factual validity.

4.4 Case Study: Persona-Aware Prompt Does not Always Make Sense

4.4.1 Problem Definition

The preceding literature review has documented the diverse ways researchers have proposed and applied persona-aware prompt methods $\mathcal{P}_{\text{profs}} \subseteq \mathcal{P}$ across various tasks, while the reasoning assumptions analysis has revealed the underlying beliefs about language model $\mathcal{M} = \{M_{(\text{a-gen})}, M_{(\text{a-eval})}, M_{(\text{m-eval})}, \pi_\theta\}$ behaviors and prompt instructional functions $\mathcal{I} = \{i_1, i_2, \ldots, i_L\}$ that these methods depend upon. A recurring pattern across both the methodological innovations and their foundational assumptions is the reliance on relatively simple, coarse-grained persona descriptions, most commonly phrases such as "you are an expert" or "you are an excellent professor", with the implicit expectation that such assignments will reliably enhance model performance. However, the reasoning assumptions review demonstrated that many of these beliefs about how personas shape model reasoning $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D}) = \mathfrak{R}_{\mathcal{I},\mathcal{D}} \rightarrow \mathcal{R}_{\mathcal{I},\mathcal{D}} \rightarrow \rho_{\mathcal{I},\mathcal{D}}$ remain unverified or only partially supported by empirical evidence. This gap between widespread practice and empirical validation motivates the present case study: a systematic investigation into whether fine-grained distinctions in persona specification produce measurable differences in generation quality. Specifically, this research examines how variations in professor titles, represented as distinct prompt methods $p_{(\text{profs},k)} \in \mathcal{P}_{\text{profs}}$ where each $p_{(\text{profs},k)}$ corresponds to a specific academic rank, affect the quality of mathematical definition generation across a comprehensive dataset of 439 math terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ spanning 18 subject areas $S = \{s_1, s_2, \ldots, s_{18}\}$. By holding all other prompt instructions $\mathcal{I}$ nearly identical while varying only the professor title, this study isolates the effect of persona specificity on model output, testing whether the common baseline of "you are an excellent professor" $p_{\text{e-pro}} \in \mathcal{P}_{\text{base}}$ represents an optimal choice or whether more nuanced role specifications $p_{(\text{profs},k)} \in \mathcal{P}_{\text{profs}}$ yield systematically different performance outcomes as measured by evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$.

4.4.2 Dataset

The dataset employed in the Professor-Rank-Aware Prompt research is identical to the extended dataset developed and utilized in the AUDIT prompt study, maintaining the same structure and content across all mathematical subjects and terminologies. This dataset comprises 18 mathematical subjects $S = \{s_1, s_2, \ldots, s_{18}\}$ with a total of 921 mathematical terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$, yielding 8,289 term-specific generation tasks when applied across eight proposed Professor-Rank-Aware Prompt variations $\mathcal{P}_{\text{profs}} = \{p_{(\text{profs},1)}, p_{(\text{profs},2)}, \ldots, p_{(\text{profs},8)}\}$ and the baseline excellent-professor-aware prompt $p_{\text{e-pro}} \in \mathcal{P}_{\text{base}}$. This wide coverage across diverse mathematical domains enables a robust evaluation of how different prompt methods under various professor-rank

settings, each serving as an indication of optimal or suboptimal mathematical generation capabilities, can be elicited through fine-grained rank specifications embedded within the prompt instructions $\mathcal{I}$. The scale and breadth of the dataset ensure that observed performance patterns can be reliably attributed to the nuanced variations in persona context rather than subject-specific idiosyncrasies.

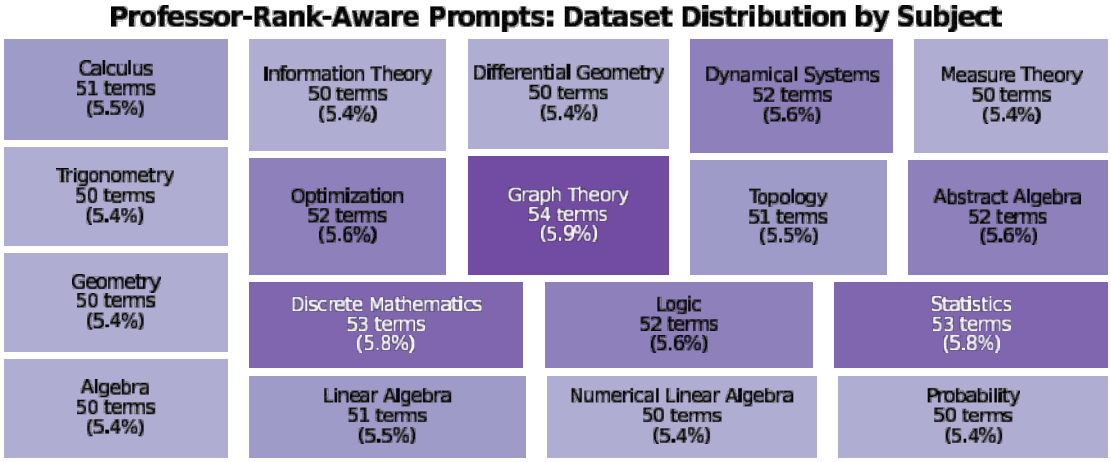


Figure 4.1: Professor-Rank-Aware Prompt: Dataset Distribution by Subject

4.4.3 Methods

Proposed Prompt Methods

Building upon the persona-aware prompt methods reviewed in the literature, eight distinct professor-aware prompt variations were designed to reflect the hierarchy and specialization present within academic mathematics departments. These variations capture fine-grained distinctions in academic rank, Distinguished Professor, Emeritus Professor, Full Professor, Associate Professor, Assistant Professor, and Lecturer, as well as distinctions in professional focus between Research Professor and Teaching Professor. Critically, each prompt maintains identical structural components and instructional content: all include the directive to generate a definition with detailed mathematical explanations in a single paragraph using LaTeX formatting, with the explicit instruction to avoid truncation. Only the professor title itself varies across prompts, ensuring that any observed performance differences can be attributed solely to the persona specification rather than to other prompt elements. This controlled manipulation represents a direct empirical test of whether fine-grained title differences influence generation quality, a question raised but not systematically addressed in the prior literature.

Professor-Rank-Aware Prompt Methods

Distinguished Professor:
you are a distinguished professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

Emeritus Professor:
you are an emeritus professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

Full Professor
you are a full professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

Research Professor
you are a research professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

Teaching Professor
you are a teaching professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

Associate Professor
you are an associate professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

Assistant Professor
you are an assistant professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

Lecturer
you are a lecturer teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

Algorithm 18 Pseudocode: Professor-Rank-Aware Prompts (8 Variations)

Require: Math subjects $S = \{s_1, s_2, \ldots, s_m\}$, Math terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ for each subject, Professor-rank-aware prompt set $\mathcal{P}_{\text{profs}} = \{p_{(\text{profs},1)}, p_{(\text{profs},2)}, \ldots, p_{(\text{profs},n)}\}$ with n professor rank settings, Generation LLM $M_{\text{a-gen}} \in \mathcal{M}$ accessed via API, Evaluation LLM $M_{\text{a-eval}} \in \mathcal{M}$ accessed via API, Eleven evaluation metrics $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$

Ensure: Evaluation results $\mathcal{R}$

for each subject $s_i \in S$ **do**
  for each term $t_{(i,j)} \in T_{s_i}$ **do**
    for each Professor-Rank-Aware Prompt $p_{(\text{profs},k)} \in \mathcal{P}_{\text{profs}}$ **do**
      Adopt professor rank persona via $p_{(\text{profs},k)}$
      Generate final definition $g_{(i,j)}^{(p_{(\text{profs},k)})}$ via $M_{\text{a-gen}}$
      for each evaluation metric $e_k \in \mathcal{E}$ **do**
        Evaluate generated final definition $g_{(i,j)}^{(p_{(\text{profs},k)})}$ using metric e_k via $M_{\text{a-eval}}$
        Store evaluation score $r_{(i,j,p_{(\text{profs},k)},k)}$
      end for
    end for
  end for
end for
Compute aggregate scores for each Professor-Rank-Aware Prompt
return Evaluation results $\mathcal{R}$ for professor-rank-aware methods

Baseline Prompt Method

The baseline prompt method employs the "Excellent Professor" persona, representing the most common approach to persona-aware prompting found throughout the literature reviewed earlier. This generic positive descriptor appears frequently across persona-aware prompt research and broader LLM applications, where phrases such as "you are an excellent professor," "you are an expert," or "you are a knowledgeable assistant" are routinely employed under the assumption that such positive characterizations will elicit high-quality responses. The reasoning assumptions review revealed that this practice rests on unverified beliefs about how such coarse-grained personas influence model behavior. By comparing the eight proposed fine-grained title variations against this commonly used baseline across 921 mathematical terms, this study provides the first systematic evaluation of whether the generic "excellent professor" persona is empirically justified, or whether the specific academic rank encoded in the persona systematically affects definition quality.

Baseline Prompt Method

Excellent-Professor-Aware Prompt:
you are an excellent professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

The corresponding psdudocode implementation is presented in algorithm 10.

4.4.4 Experiment Setup

Hyperparameters

The hyperparameters employed in this study are identical to those utilized in the Cumulative Thinking Prompt and AUDIT Prompt, ensuring methodological consistency across the three prompt-relevant investigations presented in this dissertation. This consistency guarantees that the performance analytics and comparative evaluations among the eight proposed Professor-Rank-Aware Prompt variations $\mathcal{P}_{\text{profs}} = \{p_{(\text{profs},1)}, p_{(\text{profs},2)}, \ldots, p_{(\text{profs},8)}\}$ and the baseline excellent-professor-aware prompt $p_{\text{e-pro}} \in \mathcal{P}_{\text{base}}$ are conducted under identical experimental conditions, thereby supporting reproducibility and enabling meaningful cross-study comparisons. The hyperparameter configurations are detailed in Table 2.4.

4.4.5 Evaluation Metrics

To evaluate how the prompt methods with professor-rank settings influence both absolute and relative performance in mathematical definition generation, this study employs a comprehensive evaluation framework proposed and proved in the Cumulative Thinking Prompt (in table 2.5), that comprises eleven distinct metrics. These metrics collectively assess multiple dimensions of definition quality, ranging from formal precision and conceptual completeness to structural coherence and adherence to academic conventions. By applying

this consistent evaluation framework across all Professor-Rank-Aware Prompt variations $\mathcal{P}_{\text{profs}} = \{p_{(\text{profs},1)}, p_{(\text{profs},2)}, \ldots, p_{(\text{profs},8)}\}$ and the baseline excellent-professor-aware prompt $p_{\text{e-pro}}$, the study enables systematic comparison of how fine-grained persona specifications affect generation outcomes across eighteen mathematical subjects $S = \{s_1, s_2, \ldots, s_{18}\}$.

4.5 Results and Analytics

4.5.1 Holistic Performance Analytics

This section presents the evaluation results of eight proposed Professor-Rank-Aware Prompt variations $\mathcal{P}_{\text{profs}} = \{p_{(\text{profs},1)}, p_{(\text{profs},2)}, \ldots, p_{(\text{profs},8)}\}$ compared against the excellent-professor-aware prompt $p_{\text{e-pro}} \in \mathcal{P}_{\text{base}}$ as baseline. All nine prompt methods were evaluated using the same dataset of 921 math terms $T_{s_i} = \{t_{(i,1)}, t_{(i,2)}, \ldots, t_{(i,n_i)}\}$ distributed across 18 mathematical subjects $S = \{s_1, s_2, \ldots, s_{18}\}$, which represents an extension from the 439 terms used in the Cumulative Thinking Prompt study to 921 terms in the AUDIT prompt research. The same evaluation framework $\mathcal{E} = \{e_1, e_2, \ldots, e_{11}\}$ was applied consistently across all three studies, enabling systematic comparison of how fine-grained variations in professor-rank specification influence mathematical definition generation quality.

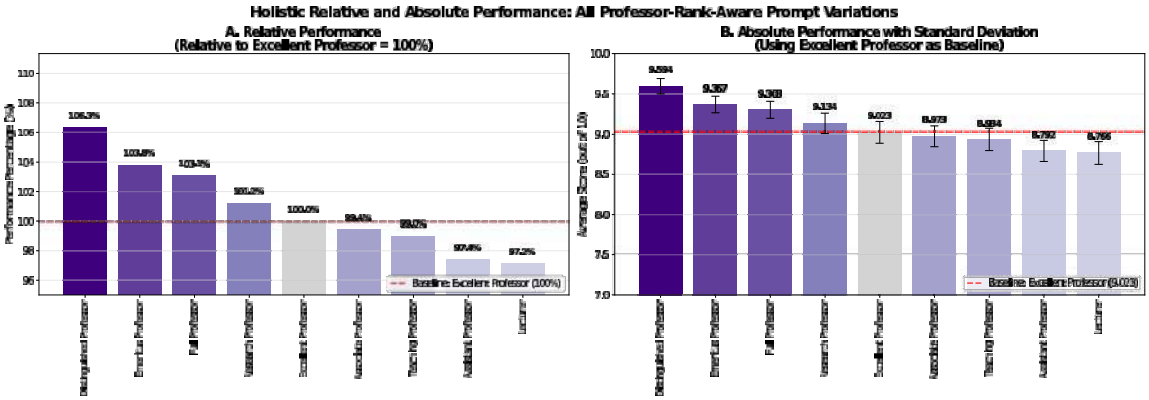


Figure 4.2: Professor-Rank-Aware Prompt: Holistic Relative and Absolute Performance (All Variations)

The holistic relative performance analysis, with the excellent-professor-aware prompt $p_{\text{e-pro}}$ normalized to 100%, reveals a clear pattern wherein higher academic ranks generally correspond to superior performance, yet the relationship is not strictly monotonic. The distinguished-professor-aware prompt achieves the highest relative performance at 106.3%, followed by emeritus-professor-aware prompt at 103.8%, full-professor-aware prompt at 103.1%, and research-professor-aware prompt at 101.2%. Notably, the baseline excellent-professor-aware prompt (100.0%) is surpassed by four higher-ranked variations but outperforms the remaining four lower-ranked variations: associate-professor-aware prompt (99.4%), teaching-professor-aware prompt (99.0%), assistant-professor-aware prompt (97.4%), and lecturer-aware prompt (97.2%). These results indicate that the coarse-grained, overly general "excellent professor" persona setting, which lacks specificity

regarding rank, role emphasis, or academic seniority, cannot guarantee optimal generation quality, at least for college STEM students' mathematical learning tasks. The substantial performance gap of 9.1 percentage points between the highest-performing distinguished-professor-aware prompt and the lowest-performing lecturer-aware prompt underscores the importance of precise persona specification.

Table 4.3: Professor-Rank-Aware Prompt: Holistic Relative Performance (All Variations)

Prompt Variation	Holistic Relative Performance (%)	Performance Difference (%)
Distinguished-Professor-Aware Prompt	106.3%	6.3%
Emeritus-Professor-Aware Prompt	103.8%	3.8%
Full-Professor-Aware Prompt	103.1%	3.1%
Research-Professor-Aware Prompt	101.2%	1.2%
Excellent-Professor-Aware Prompt	100.0%	0.0%
Associate-Professor-Aware Prompt	99.4%	-0.6%
Teaching-Professor-Aware Prompt	99.0%	-1.0%
Assistant-Professor-Aware Prompt	97.4%	-2.6%
Lecturer-Aware Prompt	97.2%	-2.8%

Notes: relative performances of eight proposed Professor-Rank-Aware Prompt variations vs excellent-professor-aware prompt as baseline, across eighteen math subjects. Positive difference values indicate higher relative performances, negative differece values indicate lower relative performances.

The holistic absolute performance comparison further substantiates the influence of professor-rank specification on generation quality, with four variations achieving higher absolute scores than the baseline excellent-professor-aware prompt (9.023) while four variations obtained lower scores. The distinguished-professor-aware prompt demonstrates the most substantial improvement with a performance difference of +0.571, followed by emeritus-professor-aware prompt (+0.344), full-professor-aware prompt (+0.280), and research-professor-aware prompt (+0.111). Conversely, the lecturer-aware prompt exhibits the largest decline (−0.257), with assistant-professor-aware prompt (−0.231), teaching-professor-aware prompt (−0.089), and associate-professor-aware prompt (−0.051) showing progressively smaller decrements. Given that the only differences across these nine prompt methods $\mathcal{P} = \{p_{\text{e-pro}}\} \cup \mathcal{P}_{\text{profs}}$ are the settings of the professor rank, while the remainder of the prompt instructions $\mathcal{I} = \{i_1, i_2, \ldots, i_L\}$ and output expectations remain consistent, this performance divergence strongly suggests that when no specific attention has been paid to identifying optimal professor-relevant settings, the optimal generation quality is difficult to achieve. The results imply that the reasoning navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ executed by the language model $M_{\text{a-gen}} \in \mathcal{M}$ is sensitive to the granularity and specificity of persona cues, which shape the model's traversal of the reasoning space $\mathfrak{R}$ toward different reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$.

Detailed summary statistics reveal the consistency and variability of performance across the nine prompt variations. The distinguished-professor-aware prompt achieves the highest holistic absolute performance (9.594) with the lowest standard deviation (0.096), indicating

both superior quality and remarkable stability across all 18 mathematical subjects. The emeritus-professor-aware prompt (9.367, std = 0.104) and full-professor-aware prompt (9.303, std = 0.105) similarly exhibit high performance with low variability. In contrast, the lecturer-aware prompt yields the lowest holistic absolute performance (8.766) with the highest standard deviation (0.141), suggesting that lower academic rank specifications not only reduce average quality but also introduce greater inconsistency across subject domains. The performance range between the maximum (9.706 for distinguished-professor-aware prompt) and minimum (8.486 for lecturer-aware prompt) scores further illustrates the substantial impact of professor-rank specification on definition generation quality. These findings collectively demonstrate that persona-aware prompting $\mathcal{P}_{\text{profs}}$ with precise, senior-rank specifications consistently guides the language model toward optimal reasoning subspaces and directions, whereas coarse-grained or junior-rank specifications yield suboptimal navigation outcomes for college STEM mathematical learning tasks.

Table 4.4: Professor-Rank-Aware Prompt: Absolute Performance Difference (Proposed vs. Baseline)

Prompt Variation	Baseline	Holistic Performance	Performance Difference	Std	Min	Max
Distinguished-Professor-Aware Prompt	9.023	9.594	0.571	0.096	9.356	9.706
Emeritus-Professor-Aware Prompt	9.023	9.367	0.344	0.104	9.185	9.509
Full-Professor-Aware Prompt	9.023	9.303	0.280	0.105	9.135	9.495
Research-Professor-Aware Prompt	9.023	9.134	0.111	0.126	8.934	9.345
Excellent-Professor-Aware Prompt	9.023	9.023	0.000	0.135	8.798	9.240
Associate-Professor-Aware Prompt	9.023	8.973	-0.051	0.129	8.709	9.214
Teaching-Professor-Aware Prompt	9.023	8.934	-0.089	0.139	8.748	9.179
Assistant-Professor-Aware Prompt	9.023	8.792	-0.231	0.132	8.545	9.028
Lecturer-Aware Prompt	9.023	8.766	-0.257	0.141	8.486	9.040

Notes: holistic absolute performance of eight proposed Professor-Rank-Aware Prompt variations with excellent-professor-aware prompt as baseline, across eighteen math subjects, positive difference values indicate higher absolute performance, negative difference values indicate lower absolute performance.

4.5.2 Subject-Level Absolute Performance Analytics

The subject-level absolute performance analysis reveals distinct patterns across the nine prompt variations, with the distinguished-professor-aware prompt consistently achieving the highest scores across all 18 mathematical subjects, ranging from 9.356 in Trigonometry to 9.706 in Optimization, demonstrating superior stability and effectiveness regardless of subject domain. The emeritus-professor-aware and full-professor-aware prompts follow as the second and third strongest performers respectively, with emeritus-professor showing particularly strong results in Abstract Algebra (9.481), Linear Algebra (9.478), and Differential Geometry (9.441), while full-professor achieves its highest scores in Abstract Algebra (9.495) and Information Theory (9.394). Notably, the baseline excellent-professor-aware prompt $p_{\text{e-pro}}$ occupies a middle position, outperforming the associate-professor, teaching-professor, assistant-professor, and lecturer variations but consistently falling

behind the four highest-ranked professor variations across all subjects. The research-professor-aware prompt demonstrates performance closely aligned with the baseline, with marginal advantages in subjects such as Topology (9.345 vs. 9.190) and Linear Algebra (9.295 vs. 9.117), suggesting that the research-focused persona yields modest improvements over the generic excellent professor specification.

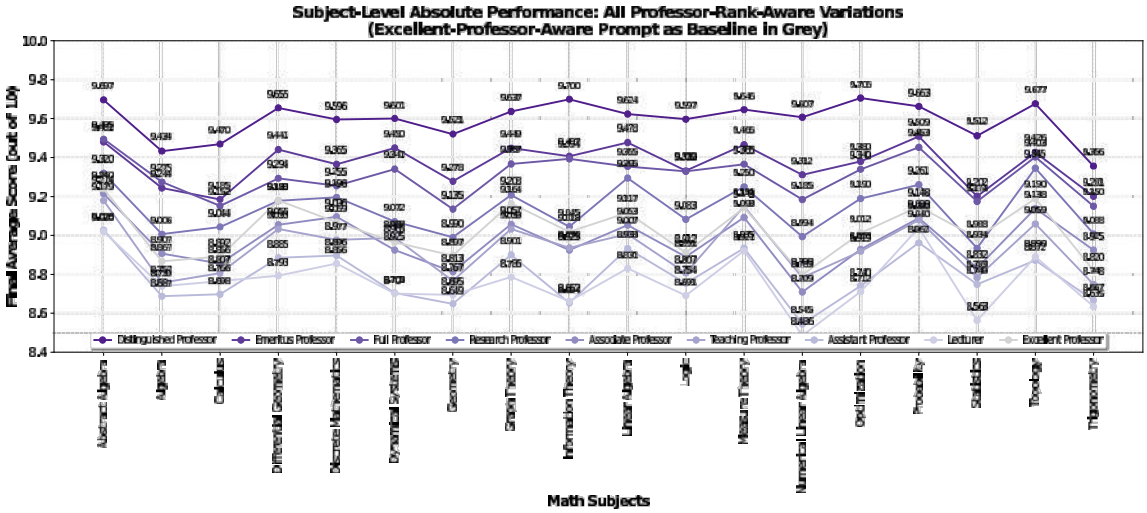


Figure 4.3: Professor-Rank-Aware Prompt: Subject-Level Absolute Performance (All Variations)

Table 4.5: Professor-Rank-Aware Prompt: Subject-Level Absolute Performance (All Variations)

Subject	D-P	E-P	F-P	R-P	Ex-P	Ao-P	T-P	Ai-P	Lect
Abstract Algebra	9.697	9.481	9.495	9.320	9.240	9.214	9.179	9.028	9.020
Algebra	9.434	9.244	9.275	9.006	8.867	8.907	8.757	8.687	8.739
Calculus	9.470	9.185	9.152	9.044	8.892	8.856	8.807	8.698	8.766
Differential Geometry	9.655	9.441	9.294	9.178	9.180	9.055	9.033	8.885	8.793
Discrete Mathematics	9.596	9.365	9.255	9.196	9.069	9.096	8.977	8.896	8.856
Dynamical Systems	9.601	9.450	9.341	9.072	8.965	8.925	8.988	8.705	8.702
Geometry	9.521	9.278	9.135	8.990	8.897	8.813	8.767	8.649	8.695
Graph Theory	9.637	9.449	9.367	9.208	9.164	9.057	9.033	8.901	8.786
Information Theory	9.700	9.407	9.394	9.045	9.018	8.925	8.936	8.654	8.662
Linear Algebra	9.624	9.478	9.355	9.295	9.117	9.053	9.007	8.933	8.831
Logic	9.597	9.332	9.329	9.083	8.912	8.891	8.807	8.754	8.691
Measure Theory	9.646	9.466	9.366	9.250	9.143	9.093	9.151	8.935	8.921
Numerical Linear Algebra	9.607	9.312	9.185	8.994	8.798	8.709	8.783	8.545	8.486
Optimization	9.706	9.380	9.340	9.190	9.012	8.928	8.919	8.740	8.712
Probability	9.663	9.509	9.453	9.261	9.148	9.089	9.080	8.962	9.040
Statistics	9.512	9.202	9.174	8.934	8.988	8.832	8.783	8.749	8.563
Topology	9.677	9.426	9.403	9.345	9.190	9.138	9.059	8.872	8.890
Trigonometry	9.356	9.201	9.150	9.008	8.820	8.925	8.748	8.667	8.635

Abbreviations: D-P = Distinguished-Professor-Aware Prompt, E-P = Emeritus-Professor-Aware Prompt, F-P = Full-Professor-Aware Prompt, R-P = Research-Professor-Aware Prompt, T-P = Teaching-Professor-Aware Prompt, Ao-P = Associate-Professor-Aware Prompt, Ai-P = Assistant-Professor-Aware Prompt, Lect = Lecturer-Aware Prompt, Ex-P = Excellent-Professor-Aware Prompt

For the lower-ranked prompt variations, a clear descending performance gradient emerges that mirrors the academic hierarchy. The associate-professor-aware prompt

maintains competitive performance in subjects such as Abstract Algebra (9.214) and Topology (9.138), while the teaching-professor-aware prompt shows relative strength in Measure Theory (9.151) and Probability (9.080), indicating that the teaching-focused persona may offer advantages in conceptually dense subjects requiring pedagogical clarity. The assistant-professor-aware and lecturer-aware prompts consistently occupy the lowest positions across all subjects, with the lecturer-aware prompt exhibiting particular difficulty in Numerical Linear Algebra (8.486), Optimization (8.712), and Statistics (8.563), where the absolute performance gaps relative to the distinguished-professor-aware prompt reach as high as 1.220 points. The subject-specific patterns further reveal that certain domains amplify the performance differentials between high-ranked and low-ranked professor variations. Numerical Linear Algebra shows the widest spread, with the distinguished-professor-aware prompt (9.607) outperforming the lecturer-aware prompt (8.486) by 1.121 points, while Trigonometry exhibits the narrowest range, with distinguished-professor (9.356) leading lecturer (8.635) by 0.721 points. These subject-level findings substantiate that the precision and seniority of professor-rank specification $\mathcal{P}_{\text{profs}}$ fundamentally shape the language model's reasoning navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$, with the effects being particularly pronounced in mathematically rigorous domains requiring high formal precision and conceptual depth.

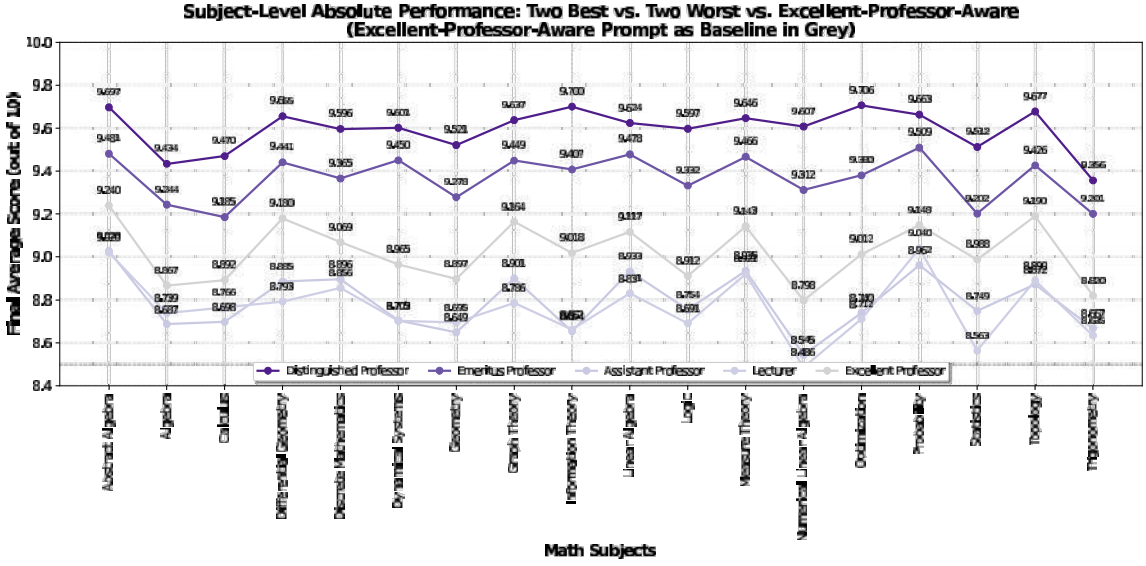


Figure 4.4: Professor-Rank-Aware Prompt: Subject-Level Absolute Performance (Selected Variations)

Now we will try to interpret the subject-level absolute performance patterns across five selected Professor-Rank-Aware Prompt variations, the two best performing (distinguished-professor and emeritus-professor), the two least performing (assistant-professor and lecturer), and the baseline excellent-professor, revealing a consistent and interpretable performance hierarchy across all 18 mathematical subjects. The distinguished-professor-aware prompt demonstrates superior performance across every subject, with scores ranging from 9.356 in Trigonometry to 9.706 in Optimization, exhibiting remarkable stability with

a standard deviation of only 0.096. The emeritus-professor-aware prompt consistently ranks second, achieving its highest scores in Probability (9.509), Linear Algebra (9.478), and Abstract Algebra (9.481), while maintaining strong performance even in subjects where the distinguished-professor shows relative moderation, such as Trigonometry (9.201). The baseline excellent-professor-aware prompt occupies a middle position, outperforming the assistant-professor and lecturer variations across all subjects but consistently falling behind both distinguished-professor and emeritus-professor variations. Notably, the performance gap between distinguished-professor and excellent-professor is most pronounced in subjects requiring precise formal structure and conceptual rigor, including Numerical Linear Algebra (9.607 vs. 8.798, gap of 0.809), Algebra (9.434 vs. 8.867, gap of 0.567), and Calculus (9.470 vs. 8.892, gap of 0.578). Conversely, the assistant-professor and lecturer prompts consistently occupy the lowest positions, with lecturer showing particular difficulty in Numerical Linear Algebra (8.486), Statistics (8.563), and Information Theory (8.662), while assistant-professor achieves its lowest scores in Algebra (8.687), Geometry (8.649), and Trigonometry (8.667). These patterns indicate that the precision and seniority of professor-rank specification directly influence the quality of mathematical definition generation, with the distinguished-professor and emeritus-professor personas guiding the language model $M_{\text{a-gen}}$ toward more accurate and formally precise reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ within the reasoning space $\mathfrak{R}$.

Table 4.6: Professor-Rank-Aware Prompt: Subject-Level Absolute Performance (Selected Variations)

Subject	**Distinguished Professor**	**Emeritus Professor**	**Assistant Professor**	**Lecturer**	**Excellent Professor**
Abstract Algebra	9.697	9.481	9.028	9.020	9.240
Algebra	9.434	9.244	8.687	8.739	8.867
Calculus	9.470	9.185	8.698	8.766	8.892
Differential Geometry	9.655	9.441	8.885	8.793	9.180
Discrete Mathematics	9.596	9.365	8.896	8.856	9.069
Dynamical Systems	9.601	9.450	8.705	8.702	8.965
Geometry	9.521	9.278	8.649	8.695	8.897
Graph Theory	9.637	9.449	8.901	8.786	9.164
Information Theory	9.700	9.407	8.654	8.662	9.018
Linear Algebra	9.624	9.478	8.933	8.831	9.117
Logic	9.597	9.332	8.754	8.691	8.912
Measure Theory	9.646	9.466	8.935	8.921	9.143
Numerical Linear Algebra	9.607	9.312	8.545	8.486	8.798
Optimization	9.706	9.380	8.740	8.712	9.012
Probability	9.663	9.509	8.962	9.040	9.148
Statistics	9.512	9.202	8.749	8.563	8.988
Topology	9.677	9.426	8.872	8.890	9.190
Trigonometry	9.356	9.201	8.667	8.635	8.820

The performance differentials across these five selected prompt variations provide compelling evidence for the value of fine-grained, trial-and-error enhanced professor-rank context settings in facilitating more reasoned definition generation and more informed mathematical definition reasoning explorations and selections. The distinguished-professor-

aware prompt, with its explicit connotations of academic seniority, research excellence, and comprehensive subject mastery, consistently achieves the highest performance across all subjects, suggesting that this persona context setting effectively primes the language model to access optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ characterized by high formal precision, conceptual completeness, and structural coherence. Similarly, the emeritus-professor-aware prompt, which conveys accumulated wisdom, pedagogical depth, and sustained scholarly contribution, enables the model to achieve performance levels that substantially exceed the baseline, particularly in subjects requiring integrated understanding across multiple conceptual layers such as Probability and Linear Algebra. In contrast, the assistant-professor and lecturer prompts, which connote earlier career stages, developing expertise, and potentially narrower instructional scope, consistently underperform relative to the baseline, with the lecturer-aware prompt exhibiting the greatest difficulty in subjects demanding rigorous formal notation and precise mathematical reasoning. This performance gradient strongly suggests that the explicit meaning embedded within professor-rank settings, specifically, the implied capabilities, knowledge depth, and reasoning sophistication associated with each rank, directly shapes how the language model navigates the holistic reasoning space $\mathfrak{R}$ toward optimal reasoning subspaces and directions.

The stark contrast between the fine-grained professor-rank specifications and the coarse-grained excellent-professor baseline further underscores the importance of precise persona context settings for achieving optimal generation quality. The baseline excellent-professor prompt, despite its ostensibly positive connotations of teaching excellence, lacks the specific rank-based cues that convey relative academic seniority, research emphasis, or pedagogical specialization. This coarse-grained specification leaves the language model to infer or approximate the intended reasoning posture, resulting in performance that, while respectable, is consistently surpassed by the more precisely specified distinguished-professor and emeritus-professor prompts across all 18 subjects. The inferior performance of the assistant-professor and lecturer prompts relative to the baseline further demonstrates that not all fine-grained specifications yield improvements; rather, it is the alignment between the rank-based connotations, seniority, expertise depth, and authority, and the cognitive demands of the definition generation task that determines effectiveness. In this specific analytics and interpretations, the selection of two best performing, two least performing, and the excellent-professor-aware prompt serves to isolate and illuminate the values embedded within reasoning context settings, in this research, the settings of professor ranks with explicit meaning that includes capabilities in generating higher or lower quality mathematical term definitions. The findings demonstrate that when the persona context conveys appropriate levels of academic seniority and subject mastery, the language model $M_{\text{a-gen}}$ is guided toward more reasoned definition generation and more informed navigation of the reasoning space $\mathfrak{R}$, whereas coarse-grained, imprecise persona specifications, or those implying insufficient expertise, fail to unlock the model's

full generative potential for college STEM mathematical learning tasks.

4.5.3 Subject-Level Relative Performance Analytics

The subject-level relative performance analysis across all eight proposed Professor-Rank-Aware Prompt variations, benchmarked against the excellent-professor-aware prompt normalized to 100%, reveals distinct clustering patterns that correspond to the academic seniority and role specialization embedded within each persona specification. The highest-performing cluster, comprising distinguished-professor (range: 104.9%–109.2%), emeritus-professor (102.4%–105.8%), full-professor (101.2%–104.7%), and research-professor variations (99.4%–102.2%), demonstrates consistent performance gains across virtually all subjects, with distinguished-professor achieving its most substantial advantages in Numerical Linear Algebra (109.2%), Logic (107.7%), and Optimization (107.7%), where the demands for formal precision and conceptual rigor are highest. The research-professor variation, while still outperforming the baseline in most subjects, shows marginal gains and even slight underperformance in Statistics (99.4%), suggesting that a research-oriented persona may not uniformly translate to pedagogical effectiveness across all mathematical domains. The middle-performing baseline excellent-professor (100.0%) is closely trailed by associate-professor (range: 98.3%–101.2%) and teaching-professor (97.7%–100.3%), with these variations showing mixed patterns, associate-professor modestly exceeds the baseline in Algebra (100.5%), Discrete Mathematics (100.3%), and Trigonometry (101.2%), while teaching-professor achieves parity or slight gains in Dynamical Systems (100.3%) and Measure Theory (100.1%), indicating that certain mid-tier academic roles may offer contextual advantages in specific subject areas. The lowest-performing cluster, comprising assistant-professor (96.0%–98.3%) and lecturer (95.3%–98.8%), exhibits consistent underperformance across all subjects, with lecturer showing its most pronounced deficits in Statistics (95.3%), Graph Theory (95.9%), and Information Theory (96.0%), where the implicit connotations of limited seniority and developing expertise appear insufficient to guide the language model $M_{\text{a-gen}}$ toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$. This clustered performance pattern underscores that the precision of professor-rank context settings directly correlates with generation quality, with senior, research-intensive personas consistently enabling superior reasoning navigation, while mid-tier and junior personas yield marginal or negative performance relative to the baseline, demonstrating that coarse-grained or insufficiently authoritative persona specifications fail to unlock the model's full generative potential for college STEM mathematical learning tasks.

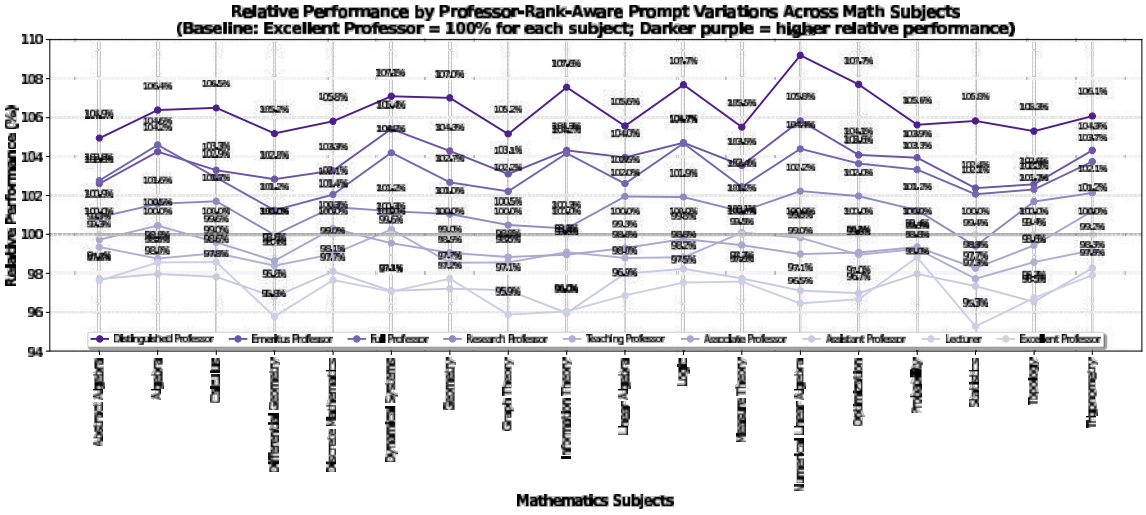


Figure 4.5: Professor-Rank-Aware Prompt: Subject-Level Performance (All Variations)

By examining the subject-level relative performances of the two best performing prompt variations (distinguished-professor and emeritus-professor), the two least performing variations (assistant-professor and lecturer), and the baseline excellent-professor prompt, we can observe that specific professor-rank settings indeed elicit markedly higher relative performance gains while also causing significant performance declines, attributable solely to variations in the professor rank context embedded within the prompts to instruct the language model $M_{\text{a-gen}}$ to produce outputs under different rank-based reasoning postures. The distinguished-professor prompt consistently achieves the highest relative performance across all 18 subjects, with gains ranging from 104.9% in Abstract Algebra to an exceptional 109.2% in Numerical Linear Algebra, while the emeritus-professor prompt follows with substantial improvements between 102.4% in Statistics and 105.8% in Numerical Linear Algebra, demonstrating that senior-rank personas effectively guide the model toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$. In stark contrast, the assistant-professor and lecturer prompts consistently underperform relative to the baseline, with the assistant-professor showing deficits from 96.0% in Information Theory to 98.3% in Trigonometry, and the lecturer prompt exhibiting even more pronounced declines from 95.3% in Statistics to 98.8% in Probability, with particularly severe underperformance in Numerical Linear Algebra (96.5%), Differential Geometry (95.8%), and Graph Theory (95.9%). This striking performance divergence, a spread of up to 13.9 percentage points between distinguished-professor (109.2%) and lecturer (95.3%) in Numerical Linear Algebra, emerges exclusively from differences in the professor-rank context specifications, as all other prompt instructions $\mathcal{I} = \{i_1, i_2, \ldots, i_L\}$ remain identical across these variations. These findings conclusively demonstrate that the explicit meaning embedded within professor-rank settings, encompassing implied academic seniority, expertise depth, and reasoning sophistication, directly shapes how the language model navigates the holistic reasoning space $\mathfrak{R}$ toward either optimal or suboptimal reasoning

subspaces and directions, with senior-rank personas eliciting substantially higher quality mathematical definition generation while junior-rank personas consistently constrain the model's generative potential for college STEM mathematical learning tasks.

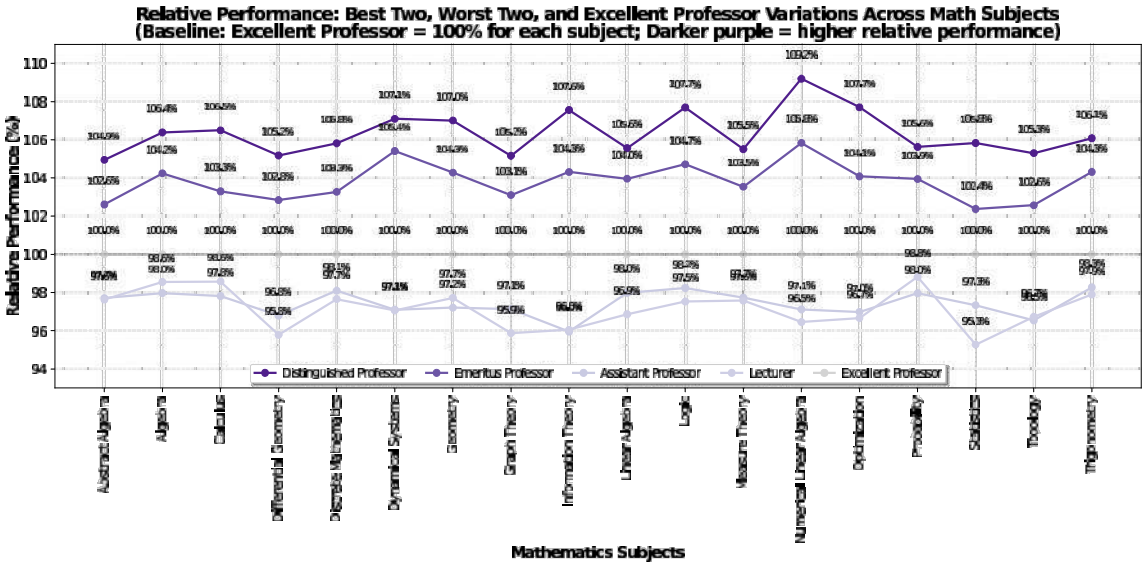


Figure 4.6: Professor-Rank-Aware Prompt: Subject-Level Performance (Selected Variations)

Table 4.7: Professor-Rank-Aware Prompt: Subject-Level Relative Performance (Selected Variations)

Subject	Distinguished Professor	Emeritus Professor	Excellent Professor	Assistant Professor	Lecturer
Abstract Algebra	104.9%	102.6%	100.0%	97.7%	97.6%
Algebra	106.4%	104.2%	100.0%	98.0%	98.6%
Calculus	106.5%	103.3%	100.0%	97.8%	98.6%
Differential Geometry	105.2%	102.8%	100.0%	96.8%	95.8%
Discrete Mathematics	105.8%	103.3%	100.0%	98.1%	97.7%
Dynamical Systems	107.1%	105.4%	100.0%	97.1%	97.1%
Geometry	107.0%	104.3%	100.0%	97.2%	97.7%
Graph Theory	105.2%	103.1%	100.0%	97.1%	95.9%
Information Theory	107.6%	104.3%	100.0%	96.0%	96.0%
Linear Algebra	105.6%	104.0%	100.0%	98.0%	96.9%
Logic	107.7%	104.7%	100.0%	98.2%	97.5%
Measure Theory	105.5%	103.5%	100.0%	97.7%	97.6%
Numerical Linear Algebra	109.2%	105.8%	100.0%	97.1%	96.5%
Optimization	107.7%	104.1%	100.0%	97.0%	96.7%
Probability	105.6%	103.9%	100.0%	98.0%	98.8%
Statistics	105.8%	102.4%	100.0%	97.3%	95.3%
Topology	105.3%	102.6%	100.0%	96.5%	96.7%
Trigonometry	106.1%	104.3%	100.0%	98.3%	97.9%

4.5.4 Performance Stability Analytics

The performance stability analysis reveals that the distinguished-professor-aware prompt achieves absolute rank stability, securing the 1st position across all 18 mathematical subjects with a mean rank of 1.00 and a rank stability of 100.00%, indicating that

this persona specification consistently guides the language model $M_{\text{a-gen}}$ toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ regardless of subject domain characteristics. The emeritus-professor and full-professor prompts demonstrate near-perfect stability with mean ranks of 2.11 and 2.89 respectively, each achieving 99.83% stability, where emeritus-professor secures 2nd place in 16 subjects and 3rd place in 2 subjects, while full-professor claims 2nd in 2 subjects and 3rd in 16 subjects, revealing a controlled rank exchange that preserves their collective dominance as the second and third strongest performers. The research-professor prompt maintains consistent 4th-place positioning in 16 subjects with only two deviations to 5th place, yielding a mean rank of 4.11 and stability of 99.83%, suggesting that the research-focused persona, while outperforming the baseline, exhibits slightly greater susceptibility to subject-specific contextual factors. In the middle tier, the excellent-professor baseline shows notable rank variability with a mean rank of 5.17 and stability of 99.39%, occupying 4th place in 2 subjects, 5th in 11 subjects, and 6th in 5 subjects, indicating that the coarse-grained "excellent professor" specification without explicit rank-based cues leads to inconsistent positioning across subjects. The associate-professor and teaching-professor prompts demonstrate progressively declining stability, with associate-professor achieving a mean rank of 6.06 and securing 5th, 6th, and 7th positions across subjects, while teaching-professor exhibits the lowest stability at 99.13% with a mean rank of 6.67, suggesting that mid-tier academic roles introduce greater rank volatility. The assistant-professor and lecturer prompts, while maintaining relatively high stability percentages (99.48% each) due to their consistent placement at the bottom of the ranking hierarchy, show a controlled rank exchange where assistant-professor secures 8th in 12 subjects and 9th in 6 subjects, while lecturer claims 8th in 6 subjects and 9th in 12 subjects, with mean ranks of 8.33 and 8.67 respectively. This rank stability evolution demonstrates that the precision and seniority of professor-rank settings directly determine both the absolute performance levels and the consistency of subject-aware performance ranking, where senior, explicitly defined personas yield superior rank stability and holistic positioning, while coarse-grained or junior specifications result in greater rank volatility and suboptimal holistic outcomes.

Table 4.8: Professor-Rank-Aware Prompt: Subject-Level Ranking Stability (All Variations)

Prompt Title	Holistic Rank	Mean Rank	Rank Stability	Obtained Ranks
Distinguished Professor	1	1.00	100.00%	Rank 1st: 18
Emeritus Professor	2	2.11	99.83%	Rank 2nd: 16, Rank 3th: 2
Full Professor	3	2.89	99.83%	Rank 2nd: 2, Rank 3th: 16
Research Professor	4	4.11	99.83%	Rank 4th: 16, Rank 5th: 2
Excellent Professor	5	5.17	99.39%	Rank 4th: 2, Rank 5th: 11, Rank 6th: 5
Associate Professor	6	6.06	99.39%	Rank 5th: 3, Rank 6th: 11, Rank 7th: 4
Teaching Professor	7	6.67	99.13%	Rank 5th: 2, Rank 6th: 2, Rank 7th: 14
Assistant Professor	8	8.33	99.48%	Rank 8th: 12, Rank 9th: 6
Lecturer	9	8.67	99.48%	Rank 8th: 6, Rank 9th: 12

Abbreviations: D-P = Distinguished-Professor-Aware Prompt, E-P = Emeritus-Professor-Aware Prompt, F-P = Full-Professor-Aware Prompt, R-P = Research-Professor-Aware Prompt, T-P = Teaching-Professor-Aware Prompt, Ao-P = Associate-Professor-Aware Prompt, Ai-P = Assistant-Professor-Aware Prompt, Lect = Lecturer-Aware Prompt, Ex-P = Excellent-Professor-Aware Prompt

4.5.5 Concrete Examples

Distinguished-Professor-Aware Prompt for Math Term Definition Generation

You are a distinguished professor teaching college level mathematics, generate the definition of smooth manifold in differential geometry with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

A smooth manifold of dimension n is a topological space M equipped with a smooth structure, which consists of the following data: First, M must be a second-countable Hausdorff topological space. Second, it must possess a smooth atlas $\mathcal{A} = \{(U_\alpha, \phi_\alpha)\}_{\alpha \in I}$, where $\{U_\alpha\}$ is an open cover of M and each $\phi_\alpha : U_\alpha \to \phi_\alpha(U_\alpha) \subset \mathbb{R}^n$ is a homeomorphism onto an open subset of $\mathbb{R}^n$, called a coordinate chart. The pair (U_α, ϕ_α) is called a chart. The crucial smoothness condition is that for any two charts (U_α, ϕ_α) and (U_β, ϕ_β) with $U_\alpha \cap U_\beta \neq \emptyset$, the transition map $\phi_\beta \circ \phi_\alpha^{-1} : \phi_\alpha(U_\alpha \cap U_\beta) \to \phi_\beta(U_\alpha \cap U_\beta)$ is a C^∞ diffeomorphism between open subsets of $\mathbb{R}^n$; that is, it is infinitely differentiable and has an infinitely differentiable inverse. This condition ensures that calculus can be done consistently on the manifold by working in local coordinates. The maximal atlas containing $\mathcal{A}$, which includes all charts compatible with these transition maps, is called the smooth structure on M. A function $f : M \to \mathbb{R}$ is said to be smooth if for every chart (U_α, ϕ_α), the composition $f \circ \phi_\alpha^{-1} : \phi_\alpha(U_\alpha) \to \mathbb{R}$ is a smooth (C^∞) function in the usual Euclidean sense.

You are a distinguished professor teaching college level mathematics, generate the definition of interpretation in logic with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In mathematical logic, an interpretation $\mathcal{I}$ is a precise mathematical structure that assigns meaning to the symbols of a formal language, thereby determining the truth values of its well-formed formulas. Formally, for a first-order language $\mathcal{L}$, an interpretation consists of a non-empty set D called the domain (or universe) of discourse, together with: (1) for each constant symbol c in $\mathcal{L}$, an element $c^{\mathcal{I}} \in D$; (2) for each n-ary function symbol f in $\mathcal{L}$, an n-ary function $f^{\mathcal{I}} : D^n \to D$; and (3) for each n-ary predicate symbol P in $\mathcal{L}$, an n-ary relation $P^{\mathcal{I}} \subseteq D^n$. Given an interpretation $\mathcal{I}$, a variable assignment σ maps each variable to an element of D. The interpretation then induces a valuation function $[\![\cdot]\!]^{\mathcal{I},\sigma}$ that recursively maps each term t to an element $[\![t]\!]^{\mathcal{I},\sigma} \in D$ and each formula ϕ to a truth value in $\{\mathbf{T}, \mathbf{F}\}$. For an atomic formula $P(t_1, \ldots, t_n)$, we have $[\![P(t_1, \ldots, t_n)]\!]^{\mathcal{I},\sigma} = \mathbf{T}$ iff $([\![t_1]\!]^{\mathcal{I},\sigma}, \ldots, [\![t_n]\!]^{\mathcal{I},\sigma}) \in P^{\mathcal{I}}$. The truth values for compound formulas are defined inductively using the standard rules for logical connectives (e.g., $[\![\phi \wedge \psi]\!]^{\mathcal{I},\sigma} = \mathbf{T}$ iff both $[\![\phi]\!]^{\mathcal{I},\sigma} = \mathbf{T}$ and $[\![\psi]\!]^{\mathcal{I},\sigma} = \mathbf{T}$) and quantifiers: $[\![\forall x \phi]\!]^{\mathcal{I},\sigma} = \mathbf{T}$ iff for every $d \in D$, $[\![\phi]\!]^{\mathcal{I},\sigma[x/d]} = \mathbf{T}$, where $\sigma[x/d]$ is the assignment that maps x to d and agrees with σ on all other variables. A formula ϕ is true in the interpretation $\mathcal{I}$ (written $\mathcal{I} \models \phi$) if $[\![\phi]\!]^{\mathcal{I},\sigma} = \mathbf{T}$ for every variable assignment σ. Thus, an interpretation provides the semantic framework that connects syntactic expressions to their meaning in a specific mathematical structure.

Emeritus-Professor-Aware Prompt for Math Term Definition Generation

You are an emeritus professor teaching college level mathematics, generate the definition of subset in discrete mathematics with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In discrete mathematics, given two sets A and B, we say that A is a **subset** of B, denoted $A \subseteq B$, if and only if every element of A is also an element of B. Formally, this is expressed using the universal quantifier and implication: $A \subseteq B \iff \forall x \, (x \in A \to x \in B)$. This definition includes the possibility that A and B are equal, meaning $A = B \implies A \subseteq B \wedge B \subseteq A$. If we wish to specify that A is a subset of B but $A \neq B$, we use the term **proper subset**, denoted $A \subset B$ or $A \subsetneq B$, defined as $A \subseteq B \wedge \exists y \, (y \in B \wedge y \notin A)$. The empty set, $\emptyset$, is a subset of every set, i.e., $\forall B, \ \emptyset \subseteq B$, which follows vacuously from the definition since there is no x for which $x \in \emptyset$ is true, making the implication $x \in \emptyset \to x \in B$ true for all x. The concept of subset establishes a partial order on the power set $\mathcal{P}(S)$ of any set S, with reflexivity $(A \subseteq A)$, antisymmetry $(A \subseteq B \wedge B \subseteq A \implies A = B)$, and transitivity $(A \subseteq B \wedge B \subseteq C \implies A \subseteq C)$.

You are an emeritus professor teaching college level mathematics, generate the definition of interior in topology with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In topology, given a topological space (X, τ) and a subset $A \subseteq X$, the interior of A, denoted $\text{int}(A)$ or A°, is defined as the union of all open sets contained in A. Formally, $\text{int}(A) = \bigcup \{U \in \tau \mid U \subseteq A\}$. Equivalently, it is the largest open subset of X contained within A, meaning that if V is any open set with $V \subseteq A$, then $V \subseteq \text{int}(A)$. The interior can also be characterized as the set of all interior points of A, where a point $x \in X$ is an interior point of A if there exists an open neighborhood U of x (i.e., $U \in \tau$ and $x \in U$) such that $U \subseteq A$. This definition directly yields key properties: $\text{int}(A)$ is open, $\text{int}(A) \subseteq A$, and A is open if and only if $A = \text{int}(A)$. The interior operator satisfies $\text{int}(X) = X$, $\text{int}(A) \cap \text{int}(B) = \text{int}(A \cap B)$, and $\text{int}(\text{int}(A)) = \text{int}(A)$.

Full-Professor-Aware Prompt for Math Term Definition Generation

You are a full professor teaching college level mathematics, generate the definition of integrable system in dynamical systems with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In dynamical systems theory, an integrable system is a Hamiltonian system with n degrees of freedom that possesses n independent, Poisson-commuting first integrals $F_i : \mathcal{M} \to \mathbb{R}$, $i = 1, \dots, n$, defined on a $2n$-dimensional symplectic manifold $(\mathcal{M}, \omega)$, such that $\{F_i, F_j\} = 0$ for all i, j, where $\{\cdot, \cdot\}$ denotes the Poisson bracket, and the differentials $dF_1, \dots, dF_n$ are linearly independent almost everywhere on $\mathcal{M}$. According to the Liouville–Arnold theorem, the joint level set $\mathcal{M}_{\mathbf{c}} = \{x \in \mathcal{M} : F_i(x) = c_i,\ i = 1, \dots, n\}$ is an n-dimensional Lagrangian submanifold invariant under the Hamiltonian flow. When compact and connected, $\mathcal{M}_{\mathbf{c}}$ is diffeomorphic to an n-torus $\mathbb{T}^n = (\mathbb{R}/2\pi\mathbb{Z})^n$, and there exist action-angle coordinates $(\mathbf{I}, \boldsymbol{\theta}) \in U \times \mathbb{T}^n$, with $U \subset \mathbb{R}^n$ open, such that the symplectic form becomes $\omega = \sum_{i=1}^{n} dI_i \wedge d\theta_i$ and the Hamiltonian H depends only on the action variables $\mathbf{I}$, i.e., $H = H(\mathbf{I})$. Consequently, Hamilton's equations simplify to $\dot{\mathbf{I}} = -\frac{\partial H}{\partial \boldsymbol{\theta}} = \mathbf{0}$ and $\dot{\boldsymbol{\theta}} = \frac{\partial H}{\partial \mathbf{I}} = \boldsymbol{\omega}(\mathbf{I})$, yielding linear flow on the torus: $\boldsymbol{\theta}(t) = \boldsymbol{\theta}_0 + \boldsymbol{\omega}(\mathbf{I})\, t \mod 2\pi$, which is quasi-periodic with frequencies $\boldsymbol{\omega} = (\omega_1, \dots, \omega_n)$. This complete integrability implies the dynamics are regular and solvable by quadratures, contrasting with non-integrable systems that typically exhibit chaotic behavior.

You are a full professor teaching college level mathematics, generate the definition of experiment in probability with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In probability theory, an **experiment** (or **random experiment**) is a well-defined procedure or process whose outcome cannot be predicted with certainty before it is performed, but for which the set of all possible individual outcomes is known. Formally, an experiment is characterized by a triple $(\Omega, \mathcal{F}, P)$, where:
- Ω is the **sample space**, the non-empty set of all possible elementary outcomes ω.
- $\mathcal{F}$ is a σ-algebra on Ω, representing the collection of all events (subsets of Ω) to which probabilities can be assigned. It satisfies: (i) $\Omega \in \mathcal{F}$, (ii) if $A \in \mathcal{F}$ then $A^c \in \mathcal{F}$, and (iii) if $A_1, A_2, \dots \in \mathcal{F}$ then $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$.
- $P : \mathcal{F} \to [0, 1]$ is a **probability measure** satisfying the axioms: (i) $P(\Omega) = 1$, (ii) $P(A) \geq 0$ for all $A \in \mathcal{F}$, and (iii) countable additivity: for any sequence of pairwise disjoint events $\{A_i\}_{i=1}^{\infty}$ in $\mathcal{F}$, $P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$.
Each performance of the experiment is a **trial**, yielding a single outcome $\omega \in \Omega$. An **event** $A \in \mathcal{F}$ occurs if the outcome ω belongs to A, and its likelihood is quantified by $P(A)$.

Research-Professor-Aware Prompt for Math Term Definition Generation

You are a research professor teaching college level mathematics, generate the definition of isometry in differential geometry with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In differential geometry, an isometry is a smooth map $\phi : (M, g) \to (N, h)$ between two Riemannian manifolds that preserves the Riemannian metric. Formally, ϕ is an isometry if it is a diffeomorphism and its pullback satisfies $\phi^* h = g$. This means that for any point $p \in M$ and any tangent vectors $X_p, Y_p \in T_p M$, the inner product is preserved: $g_p(X_p, Y_p) = h_{\phi(p)}(d\phi_p(X_p), d\phi_p(Y_p))$, where $d\phi_p : T_p M \to T_{\phi(p)} N$ is the differential of ϕ at p. Consequently, an isometry preserves all geometric quantities derived from the metric, including lengths of curves, angles between curves, and intrinsic curvatures. If such a map exists, the manifolds M and N are said to be isometric, meaning they are geometrically indistinguishable from the perspective of their Riemannian structures.

You are a research professor teaching college level mathematics, generate the definition of hypothesis testing in statistics with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.



In statistics, hypothesis testing is a formal inferential procedure used to decide between two competing conjectures about a population parameter θ based on sample data. The process begins by stating a null hypothesis $H_0 : \theta \in \Theta_0$ (often representing a default or no-effect state) and an alternative hypothesis $H_1 : \theta \in \Theta_1$ (where $\Theta_0 \cap \Theta_1 = \emptyset$). A test statistic $T = T(X_1, \ldots, X_n)$, which is a function of the random sample, is calculated. Under the assumption that H_0 is true, the sampling distribution of T is determined. A significance level α (commonly 0.05) is chosen, defining the probability of a Type I error: $\alpha = P(\text{reject } H_0 \mid H_0 \text{ is true})$. The critical region or rejection region region C is then constructed such that $P(T \in C \mid H_0) \leq \alpha$. If the observed value of the test statistic t_{obs} falls within C, H_0 is rejected in favor of H_1; otherwise, H_0 is not rejected. The conclusion is often supplemented with a p-value, defined as $p = P(T \text{ is at least as extreme as } t_{\text{obs}} \mid H_0)$, where a small p-value provides evidence against H_0. The power of the test, $1 - \beta$, where $\beta = P(\text{fail to reject } H_0 \mid H_1 \text{ is true})$ is the Type II error probability, measures the test's ability to correctly reject a false null hypothesis.

4.6 Conclusion

4.6.1 Research Contributions

This study establishes that fine-grained professor-rank settings serve as explicit and effective mechanisms for eliciting language models' exploration of optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$ where more valuable mathematical information for definition generation resides. The distinguished-professor, emeritus-professor, and full-professor specifications, with their explicit connotations of academic seniority, research excellence, and comprehensive subject mastery, consistently guide the language model $M_{\text{a-gen}}$ toward reasoning subspaces characterized by high formal precision, conceptual completeness, and structural coherence. In contrast, the assistant-professor and lecturer specifications, which imply developing expertise and narrower instructional scope, lead the model toward less optimal reasoning subspaces, resulting in lower generation quality. This finding demonstrates that the precision of persona context settings directly influences the model's ability to navigate the holistic reasoning space $\mathfrak{R}$ effectively, enabling generation tasks to be implemented with higher quality and reduced computational struggle, as the model requires fewer iterative refinements when primed with appropriately authoritative persona specifications.

The ranking of professor titles established in this research, from distinguished professor at the highest performance tier to lecturer at the lowest, can be regarded as a reliable signal for predicting whether specific downstream tasks can be performed with certain quality expectations. The consistent performance hierarchy observed across all 18 mathematical subjects, with distinguished-professor achieving 106.3% relative performance, emeritus-professor achieving 103.8%, full-professor achieving 103.1%, and lecturer at 97.2%, demonstrates that professor-rank specifications provide a valid proxy for expected generation quality. This finding offers compelling evidence that the commonly employed "excellent professor" persona setting, which lacks the specificity and rank-based cues that convey relative academic seniority and expertise depth, is not an optimal choice for tasks requiring high-quality mathematical definition generation. The baseline excellent-professor

prompt, despite its ostensibly positive connotations, was surpassed by four higher-ranked specifications and outperformed the four lower-ranked variations, confirming that coarse-grained, untested persona settings cannot guarantee optimal outcomes.

This study provides a new avenue for exploring how different context settings can constrain and guide language models to generate more controlled outputs with higher performance and lower cost. The professor-rank specifications investigated in this research represent a training-free intervention that requires no model fine-tuning, no additional computational resources beyond standard inference, and no specialized technical expertise to implement. This accessibility makes such context-based optimization strategies applicable across a wide range of fields beyond mathematical definition generation, including technical documentation, educational content creation, scientific writing, and domain-specific knowledge synthesis. The finding that simple modifications to persona specifications, changing a few words in the prompt instructions $\mathcal{I}$, can yield performance improvements of up to 6.3% relative to a strong baseline demonstrates that significant gains can be achieved without the computational expense and expertise demands of fine-tuning or reinforcement learning approaches, democratizing access to enhanced language model performance for practitioners across diverse domains.

In summary, this research contributes three key advancements to the understanding and application of persona-aware prompting $\mathcal{P}_{\text{profs}}$. First, it demonstrates that fine-grained professor-rank specifications serve as explicit navigation signals that guide language models toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$, enabling higher quality generation with reduced computational struggle. Second, it establishes that the ranking of professor titles functions as a reliable quality signal, providing empirical evidence that the coarse-grained "excellent professor" baseline is suboptimal for tasks requiring precise mathematical definition generation. Third, it introduces a training-free, computationally accessible approach to performance optimization that can be readily adopted across diverse fields and applications, demonstrating that careful attention to context specification can unlock significant language model capabilities without the barriers of specialized technical expertise or extensive computational resources. Together, these contributions advance both the theoretical understanding of how persona context shapes reasoning navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ and the practical methodology for optimizing language model outputs in educational and technical domains.

4.6.2 Future Directions

Building upon the findings of this Professor-Rank-Aware Prompt research, several promising future directions emerge for extending this work. The first new research direction is to expand the multiple professor-rank settings developed in this study to more challenging mathematical problems with different context configurations. Rather than focusing solely

on definition generation, future research could apply this fine-grained approach to explore possible, unique, and reasonable reasoning directions for difficult or unsolved mathematical problems. This could be implemented either with a single large language model instance that switches between different persona settings to examine a problem from multiple angles, or with multiple LLM instances collaborating together, where each instance adopts a distinct reasoning style. Some instances could follow conventional reasoning behaviors, while others might employ innovative or unconventional approaches, collectively navigating toward less explored or never explored reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$. This collaborative, multi-perspective approach could help researchers discover novel solution pathways that might remain hidden when relying on a single reasoning approach.

The second new research direction involves expanding the target audience beyond college STEM students to encompass diverse learner populations with varying mathematical backgrounds and learning needs. The fine-grained persona settings demonstrated in this study could be adapted and extended for social science and liberal arts students who may require more accessible explanations, middle school students encountering mathematical concepts for the first time, and workplace professionals who need practical understanding of mathematics and artificial intelligence fundamentals to remain competitive in an increasingly AI-driven economy. This expansion would require developing and evaluating a richer set of fine-grained context settings beyond academic ranks, including specifications for the language model instance's persona (e.g., "patient tutor," "practical practitioner," "conceptual bridge-builder"), background (e.g., "explaining to non-technical audiences," "connecting to social science applications"), expression style (e.g., "using analogies," "emphasizing intuition," "minimizing formal notation"), and generation difficulty settings (e.g., "introductory," "intermediate," "advanced"). By systematically varying these dimensions, future research could identify optimal configurations for different learner profiles, making language model applications for target tasks more customizable and accessible to populations traditionally underserved by mathematically intensive educational materials.

The third new research direction is to adopt the design methodology of this research into the broader design, optimization, and application of LLM agent systems for end users. The systematic approach developed in this study, proposing multiple context settings, evaluating their performance, and verifying which configurations consistently yield better results, can serve as a template for building agent systems that maintain a queue of verified and validated context settings. These settings could be dynamically initialized by the system on-the-fly based on the task characteristics, user preferences, or historical performance data, or they could be manually selected by users when they interact with the LLM agent to reason toward optimal solutions. This approach would allow users to choose, for example, whether they want a distinguished professor persona for rigorous formal

explanations, a teaching professor persona for clear instructional guidance, or a research professor persona for exploring cutting-edge connections. By making such customizable, training-free optimization strategies accessible to end users, this research contributes to the broader goal of making advanced language model capabilities practical and usable for a wide range of real-world applications without requiring specialized technical expertise.

Chapter 5

Holistic Conclusion and Discussions

5.1 What we Have Achieved

The three sequential prompt-relevant studies presented in this dissertation collectively establish a progressive understanding of how different reasoning structures and context specifications influence large language model performance on mathematical definition generation tasks. The first study introduced the Cumulative Thinking Prompt p_{ct}, a linear reasoning method $\mathcal{L} = \{\ell_1 \to \ell_2 \to \cdots \to \ell_R\}$ that operationalized structured knowledge accumulation prior to generation. By instructing the language model $M_{\text{a-gen}}$ to perform five sequential recalling steps $\{r_1, r_2, \ldots, r_5\}$ progressing from essential to abstract understanding, this method demonstrated that explicit preparatory reasoning could significantly enhance definition quality compared to conventional baselines. The Cumulative Thinking Prompt achieved a holistic absolute performance of 8.447 across 439 mathematical terms, achieving higher average scores than the direct prompt, step-by-step prompt, and excellent-professor-aware prompt, and suggesting that deliberate knowledge scaffolding before generation is associated with improved definition quality. This study laid the foundation for understanding how linear reasoning navigation $\mathcal{N}_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ can be structured to guide models toward optimal reasoning subspaces $\mathcal{R}_{\mathcal{I},\mathcal{D}} \subseteq \mathfrak{R}$ and reasoning directions $\rho_{\mathcal{I},\mathcal{D}}$.

The second study advanced this understanding by introducing the AUDIT prompt, an iterative reasoning framework $\mathcal{I} = \{\iota_1(\overset{\circlearrowleft}{\to} \iota_2)^{\delta_1}(\overset{\circlearrowleft}{\to} \iota_3)^{\delta_2}\}$ that incorporated recursive quality audits and refinement cycles into the generation process. Unlike linear reasoning, which produces a single output without intermediate feedback, the AUDIT prompt enabled the language model to generate an initial definition, receive structured audit feedback, refine its output, and repeat this cycle to progressively enhance quality. The evaluation demonstrated that AUDIT V3, which underwent two full audit-refinement cycles, achieved a holistic relative performance of 114.42% against the baseline and outperformed the Cumulative Thinking Prompt by 11.61 percentage points, with exceptionally low variability (std = 0.058) across 921 mathematical terms spanning 18 subjects. This study established

that iterative reasoning, with its capacity for dynamic adjustment of reasoning directions based on self-assessed quality, is associated with higher and more consistent output quality relative to linear reasoning approaches under the evaluated conditions. The finding that AUDIT V1 performed nearly identically to the Cumulative Thinking Prompt suggested that the performance difference was associated specifically with the iterative refinement mechanism rather than with structural complexity alone.

The third study extended the investigation beyond reasoning structure to examine the role of persona specificity in guiding language model behavior. The Professor-Rank-Aware Prompt Framework proposed eight distinct professor-rank specifications $\mathcal{P}_{\text{profs}} = \{p_{(\text{profs},1)}, p_{(\text{profs},2)}, \ldots, p_{(\text{profs},8)}\}$, each carrying explicit connotations of academic seniority, research emphasis, teaching focus, or career stage. The evaluation revealed a clear performance hierarchy aligned with rank, with distinguished-professor achieving 106.3% relative performance, emeritus-professor achieving 103.8%, full-professor achieving 103.1%, and lecturer at 97.2%. This demonstrated that fine-grained persona specifications serve as effective navigation signals, guiding the language model toward reasoning subspaces containing more valuable mathematical information. Critically, the coarse-grained excellent-professor baseline, which lacked specific rank-based cues, was outperformed by four higher-ranked variations and outperformed four lower-ranked variations, indicating that the coarse-grained 'excellent professor' specification yielded lower relative performance than four fine-grained alternatives for tasks requiring precise definition generation. Together, these three studies establish that both the structure of reasoning (linear vs. iterative) and the precision of context specifications (coarse-grained vs. fine-grained) fundamentally shape language model navigation of the reasoning space $\mathfrak{R}$, with iterative reasoning and precise persona settings yielding superior and more stable performance across diverse mathematical domains.

5.2 What we Should Improve

The first area for improvement concerns the transition from static to adaptive prompt configurations. Currently, the three proposed methodologies rely on fixed structural parameters: the Cumulative Thinking Prompt employed a predetermined sequence of five recalling steps with fixed directional constraints (e.g., "from essential to abstract"); the AUDIT Prompt utilized a rigid two-cycle audit-refinement loop; and the Professor-Rank-Aware Prompt was limited to eight specific academic persona variations. These static hyperparameters constrain the applicability of the methods to diverse user profiles and task complexities. Future research should aim to generalize these structures into dynamic, customizable frameworks where parameters such as the depth of knowledge accumulation, the number of refinement cycles, and the granularity of persona specifications can be adaptively adjusted. By enabling users to configure these elements based on

specific generation needs and cognitive loads, the proposed methods could achieve broader generalizability across varied educational and professional contexts.

The second area for improvement involves expanding the adaptability of the evaluation framework. The current set of eleven metrics was rigorously designed to assess definition quality against the standards of college-level STEM mathematics, emphasizing formal precision, conceptual completeness, and theoretical correctness. However, this focus may not adequately capture the pedagogical needs of diverse learner populations, such as social science students requiring intuitive, entry-level explanations or workplace professionals seeking practice-oriented applications. Future work should develop a modular evaluation architecture where metrics are dynamically selectable and definable based on target audience characteristics. This would allow for the customization of evaluation criteria, shifting emphasis from formal rigor to pedagogical clarity or practical relevance, thereby ensuring that generated definitions are assessed against quality standards that are aligned with specific learning objectives and user backgrounds.

The third area for improvement pertains to the holistic integration of reasoning strategies, persona specifications, and evaluation mechanisms into a unified, modular framework. While this dissertation has validated each component individually, a comprehensive system that allows for the on-demand assembly and decomposition of these modules remains to be developed. Such a framework would enable the dynamic selection of linear or iterative reasoning paths, fine-grained persona contexts, and tailored evaluation metrics, creating unique, task-specific prompt-engineering pipelines. Realizing this vision requires extensive empirical validation across a wider range of domains and significant computational resources. Therefore, this limitation reflects not a methodological deficit, but a pragmatic trade-off imposed by the temporal and funding constraints of doctoral research, pointing toward a substantial avenue for future large-scale investigation.

Chapter 6

Holistic Future Research Directions

6.1 What We Can Expect

The three prompt methodologies developed and validated in this dissertation - namely, the Cumulative Thinking Prompt p_{ct} for linear reasoning with progressive knowledge accumulation, the AUDIT Prompt p_{audit} for iterative reasoning with recursive content refinement, and the Professor-Rank-Aware Prompt $\mathcal{P}_{profs}$ for fine-grained persona specification - have collectively established a robust foundation for enhancing large language model performance on math definition generation tasks. However, the principles underlying these methodologies are not confined to mathematics education; rather, they offer broader implications for how language models $\mathcal{M}$ can be guided by these prompt strategies to navigate complex reasoning spaces $\mathfrak{R}$ across diverse domains, user populations, and application contexts. This chapter explores several promising future research directions that extend beyond the current scope of this research, encompassing disciplinary expansion to natural and social sciences, dynamical customization for flexible configurations $\mathcal{P}_{config}$, and professional applications in workplace environments. Each direction builds upon the empirical findings and methodological innovations presented in the preceding studies, offering a roadmap for future investigations that can further unlock the potential of prompt-guided reasoning navigation $N_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ for more real-world applications.

Future Direction 1: Extend from Math to Natural Science and Social Science

Natural sciences such as physics, chemistry, and computer science share a key characteristic with mathematics: they rely on precise terminology where complex concepts are built upon simpler, foundational definitions. For example, understanding quantum states requires first mastering force and energy, just as neural network architectures depend on basic algorithms. Because of this shared structural dependency, the proposed prompt methods can be directly applied to these fields. The Cumulative Thinking Prompt p_{ct} can guide learners through structured recalling of progressively abstract scientific understandings within extended term sets $T_{s_k}^{ext}$. The AUDIT Prompt's iterative refinement

can help identify and correct misconceptions through discipline-specific quality audits using adapted metrics $\mathcal{E}_{sci} \subseteq \mathcal{E}$. Furthermore, the Professor-Rank-Aware Prompt framework can incorporate adapted academic hierarchies, such as distinguished physicist or research computer scientist, to provide contextualized pedagogical personas $C_k \in \mathcal{A}$.

The adaptation to social sciences, including economics, business, and related disciplines, presents equally promising opportunities, as these fields maintain rigorous methodological standards, formal theoretical frameworks, and precisely defined terminologies that demand systematic mastery. Economics requires learners to progress from foundational concepts such as supply, demand, and elasticity to advanced constructs including game theory, econometric identification, and general equilibrium, where definitions are grounded in formal mathematical structures and empirical validation. Business education encompasses strictly defined vocabularies across finance, accounting, and strategic management, where terms such as discounted cash flow, capital asset pricing, and competitive advantage carry precise technical meanings essential for professional practice and scholarly discourse. The Cumulative Thinking Prompt p_{ct} could guide learners through progressively abstract theoretical understandings while maintaining disciplinary rigor; the AUDIT Prompt's iterative refinement could support recursive engagement with formal models and empirical evidence; and the Professor-Rank-Aware Prompt framework could incorporate personas such as distinguished economist or research strategist that reflect the distinct theoretical orientations and methodological commitments within social science academia, effectively expanding the persona set to $\mathcal{P}^{soc}_{profs} \subset \mathcal{P}$.

Future Direction 2: Extend from Static Settings to Flexible Configurations

Beyond disciplinary extension, the three proposed prompt methods offer substantial flexibility for practical customization to accommodate diverse learner profiles and pedagogical objectives through a configurable parameter set $\mathcal{P}_{config}$. The Cumulative Thinking Prompt's fixed five-step recalling structure $N = 5$ could be dynamically adjusted based on learner proficiency levels, where novice learners in early stages of conceptual acquisition might benefit from fewer recalling steps $N_{recall} < 5$ to reduce cognitive overload and information absorption burden, while advanced learners could engage with extended sequential reasoning $N_{recall} > 5$ to deepen abstract integration. Similarly, the textual constraints embedded within prompt instructions $\mathcal{I}$, such as the directional cues "essential" and "abstract," could be substituted with domain-specific terminology to reflect precise information generation requirements, including "foundational to applied" for engineering contexts, "micro to macro" for economics, and "empirical to theoretical" for social science research methodologies. The AUDIT Prompt's refinement cycle count R could be made configurable based on task complexity or user-defined quality thresholds $\mathcal{E}_{iter}$, enabling learners to determine how many iterative audit-refinement rounds are appropriate for their specific learning goals. The Professor-Rank-Aware Prompt framework could further

accommodate customizable persona granularity, allowing learners or educators to select not only academic ranks but also specialized subfields, such as "distinguished macroeconomist" or "research computer scientist specializing in machine learning," thereby providing highly targeted contextual guidance that aligns with learners' specific areas of interest and study, effectively refining the agent configuration $A_k = (\mathcal{M}_k, C_k, R_k)$.

Future Direction 3: Extend from Learning to Workplace

If we generalize the mathematical definition generation task to reasoning-based content generation, the three proposed prompt methods can be naturally and logically extended to professional workplace contexts, where information seeking and systematic understanding are equally common and essential practices. Professionals across industries regularly encounter specialized terminologies, evolving concepts, and complex technical frameworks that demand precise comprehension and effective communication, from financial analysts mastering new regulatory definitions to healthcare professionals interpreting updated medical protocols. The Cumulative Thinking Prompt p_{ct} could assist workplace professionals in systematically building foundational understanding of unfamiliar domains through structured knowledge accumulation, enabling them to progressively grasp essential concepts before engaging with advanced, context-specific applications. The AUDIT Prompt's iterative refinement mechanism could support professionals in critically evaluating their understanding of new policies, technologies, or market dynamics through recursive quality audits and refinement cycles, thereby reducing the risk of misinterpretation that could lead to costly errors. The Professor-Rank-Aware Prompt framework could be adapted to incorporate workplace-relevant personas $\mathcal{A}_{work}$, such as "distinguished industry expert," "senior consultant," or "research director," providing contextualized guidance that aligns with professional hierarchies and expertise expectations. This extension would not only validate the generalizability of the proposed methodologies beyond academic settings but also contribute to lifelong learning and professional development, empowering workers to efficiently acquire and refine knowledge in rapidly changing industries without requiring formal educational programs, leveraging the reasoning space navigation $N_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ for practical decision-support tasks.

In conclusion, the future directions outlined in this chapter envision a transition from static, domain-specific prompt engineering to dynamic, context-aware reasoning systems. By expanding the application scope to natural and social sciences, adapting configurations to individual learner needs through $\mathcal{P}_{config}$, and extending utility to professional workplace environments, the proposed methodologies promise to transform large language models into versatile cognitive partners. These advancements will not only enhance the precision and reliability of knowledge generation but also democratize access to high-quality educational and professional resources, ensuring that prompt-guided navigation $N_{\mathcal{M}}(\mathcal{I}, \mathcal{D})$ remains a robust and scalable solution for complex reasoning tasks across diverse real-world contexts.

References

[1] H. Liao, S. Hu, Z. Zhu, H. He, and Y. Jin, "Forest for the trees: Overarching prompting evokes high-level reasoning in large language models," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, L. Chiruzzo, A. Ritter, and L. Wang, Eds., Stroudsburg, PA, USA: Association for Computational Linguistics, 2025, pp. 1433–1453. [Online]. Available: https://doi.org/10.18653/v1/2025.naacl-long.66.

[2] S. Sun, Y. Liu, S. Wang, C. Zhu, and M. Iyyer, "PEARL: prompting large language models to plan and execute actions over long documents," *arXiv [cs.CL]*, Y. Graham and M. Purver, Eds., pp. 469–486, 2023. arXiv: 2305.14564 [cs.CL]. [Online]. Available: https://aclanthology.org/2024.eacl-long.29.

[3] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. H. Chi, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *International Conference on Learning Representations*, vol. abs/2203.11171, 2022. eprint: 2203.11171. [Online]. Available: https://openreview.net/pdf?id=1PL1NIMMrw.

[4] Q. Tian, H. Zhu, L. Wang, Y. Li, and Y. Lan, "R^3 prompting: Review, rephrase and resolve for chain-of-thought reasoning in large language models under noisy context," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 1670–1685. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.114.

[5] M. A. Sultan, J. Ganhotra, and R. F. Astudillo, "Structured chain-of-thought prompting for few-shot generation of content-grounded QA conversations," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 16 172–16 187. [Online]. Available: http://dx.doi.org/10.18653/v1/2024.findings-emnlp.948.

[6] H. Nguyen, Y. Liu, C. Zhang, T. Zhang, and P. Yu, "CoF-CoT: Enhancing large language models with coarse-to-fine chain-of-thought prompting for multi-domain NLU tasks," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association

for Computational Linguistics, 2023, pp. 12 109–12 119. [Online]. Available: https://aclanthology.org/2023.emnlp-main.743.

[7] D. Zhou et al., "Least-to-most prompting enables complex reasoning in large language models," in *The Eleventh International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=WZH7099tgfM.

[8] H. Wang et al., "Cue-CoT: Chain-of-thought prompting for responding to in-depth dialogue questions with LLMs," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, 2023, pp. 12 047–12 064. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.806.

[9] Y. Sun, Y. Zhang, L. Qi, and Q. Shi, "TSGP: Two-stage generative prompting for unsupervised commonsense question answering," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Stroudsburg, PA, USA: Association for Computational Linguistics, 2022, pp. 968–980. [Online]. Available: http://dx.doi.org/10.18653/v1/2022.findings-emnlp.68.

[10] X. Ma, S. Mishra, A. Beirami, A. Beutel, and J. Chen, "Let's do a thought experiment: Using counterfactuals to improve moral reasoning," *arXiv [cs.CL]*, 2023. arXiv: 2306.14308 [cs.CL]. [Online]. Available: http://arxiv.org/abs/2306.14308.

[11] J. Liu et al., "Generated knowledge prompting for commonsense reasoning," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 3154–3169. [Online]. Available: https://aclanthology.org/2022.acl-long.225.

[12] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," *arXiv [cs.CL]*, 2022. eprint: 2201.11903 (cs.CL). [Online]. Available: http://arxiv.org/abs/2201.11903.

[13] Y. Zhang, X. Wang, L. Wu, and J. Wang, "Enhancing chain of thought prompting in large language models via reasoning patterns," *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, vol. 39, no. 24, T. Walsh, J. Shah, and Z. Kolter, Eds., pp. 25 985–25 993, 2025, ISSN: 2159-5399,2374-3468. [Online]. Available: https://doi.org/10.1609/aaai.v39i24.34793.

[14] S. Diao, P. Wang, Y. Lin, R. Pan, X. Liu, and T. Zhang, "Active prompting with chain-of-thought for large language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Stroudsburg, PA, USA: Association for Computational Linguistics, 2024, pp. 1330–1350. [Online]. Available: https://aclanthology.org/2024.acl-long.73/.

[15] X. Ye and G. Durrett, "Explanation selection using unlabeled data for chain-of-thought prompting," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 619–637. [Online]. Available: https://aclanthology.org/2023.emnlp-main.41.

[16] X. Wan, R. Sun, H. Dai, S. Arik, and T. Pfister, "Better zero-shot reasoning with self-adaptive prompting," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 3493–3514. [Online]. Available: https://aclanthology.org/2023.findings-acl.216.

[17] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic chain of thought prompting in large language models," in *The Eleventh International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=5NTt8GFjUHkr.

[18] Z. Jie and W. Lu, "Leveraging training data in few-shot prompting for numerical reasoning," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 10 518–10 526. [Online]. Available: https://aclanthology.org/2023.findings-acl.668.

[19] X. Yuan et al., "Instance-adaptive zero-shot chain-of-thought prompting," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=31xWlIdxTm.

[20] H. Sun, A. Hüyük, and M. van der Schaar, "Query-dependent prompt evaluation and optimization with offline inverse RL," in *The Twelfth International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=N6o0ZtPzTg.

[21] G. Zhao, B.-J. Yoon, G. Park, S. Jha, S. Yoo, and X. Qian, "Pareto prompt optimization," *International Conference on Learning Representations*, vol. 2025, pp. 6770–6785, 2025. [Online]. Available: https://proceedings.iclr.cc/paper_files/paper/2025/hash/13b45b44e26c353c64cba9529bf4724f-Abstract-Conference.html.

[22] K. Cobbe et al., "Training verifiers to solve math word problems," *arXiv [cs.LG]*, 2021. arXiv: 2110.14168 [cs.LG]. [Online]. Available: http://arxiv.org/abs/2110.14168.

[23] D. Hendrycks et al., "Measuring massive multitask language understanding," *arXiv [cs.CY]*, 2020. arXiv: 2009.03300 [cs.CY]. [Online]. Available: https://openreview.net/forum?id=d7KBjmI3GmQ.

[24] R. Y. Pang et al., "QuALITY: Question answering with long input texts, yes!" In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds., Seattle, United States: Association for Computational Linguistics, 2022, pp. 5336–5358. [Online]. Available: http://dx.doi.org/10.18653/v1/2022.naacl-main.391.

[25] OpenAI et al., "GPT-4 technical report," *arXiv [cs.CL]*, 15 3 2023. arXiv: 2303.08774 [cs.CL]. [Online]. Available: http://arxiv.org/abs/2303.08774.

[26] T. Brown et al., "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020. [Online]. Available: https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

[27] A. Patel, S. Bhattamishra, and N. Goyal, "Are NLP models really able to solve simple math word problems?" In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 2080–2094. [Online]. Available: http://dx.doi.org/10.18653/v1/2021.naacl-main.168.

[28] Y. Tay et al., "UL2: Unifying language learning paradigms," *arXiv [cs.CL]*, 2022. eprint: 2205.05131 (cs.CL). [Online]. Available: http://arxiv.org/abs/2205.05131.

[29] L. Banarescu et al., "Abstract meaning representation for sembanking," in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 2013, pp. 178–186. [Online]. Available: https://aclanthology.org/W13-2322/.

[30] H. Li, A. Arora, S. Chen, A. Gupta, S. Gupta, and Y. Mehdad, "MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 2950–2962. [Online]. Available: http://dx.doi.org/10.18653/v1/2021.eacl-main.257.

[31] J. FitzGerald et al., "MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, 2023, pp. 4277–4302. [Online]. Available: https://aclanthology.org/2023.acl-long.235.

[32] Y. Fu, H. Peng, A. Sabharwal, P. Clark, and T. Khot, "Complexity-based prompting for multi-step reasoning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=yf1icZHC-l9.

[33] B. Lake and M. Baroni, "Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks," en, in *International Conference on Machine Learning*, PMLR, 2018, pp. 2873–2882. [Online]. Available: https://proceedings.mlr.press/v80/lake18a.html.

[34] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "CommonsenseQA: A question answering challenge targeting commonsense knowledge," in *Proceedings of the 2019 Conference of the North*, J. Burstein, C. Doran, and T. Solorio, Eds., Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4149–4158. [Online]. Available: http://dx.doi.org/10.18653/v1/N19-1421.

[35] Y. Niu, F. Huang, J. Liang, W. Chen, X. Zhu, and M. Huang, "A semantic-based method for unsupervised commonsense question answering," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, 2021, pp. 3037–3049. [Online]. Available: http://dx.doi.org/10.18653/v1/2021.acl-long.237.

[36] P. Banerjee, K. K. Pal, A. Mitra, and C. Baral, "Careful selection of knowledge to solve open book question answering," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 6120–6129. [Online]. Available: http://dx.doi.org/10.18653/v1/P19-1615.

[37] M. Sap, H. Rashkin, D. Chen, R. Le Bras, and Y. Choi, "Social IQa: Commonsense reasoning about social interactions," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4463–4473. [Online]. Available: http://dx.doi.org/10.18653/v1/D19-1454.

[38] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," en, *Journal of machine learning research: JMLR*, vol. 21, 140:1–140:67, 2019, ISSN: 1532-4435,1533-7928. eprint: 1910.10683. [Online]. Available: http://dx.doi.org/10.5555/3455716.3455856.

[39] D. Khashabi et al., "UNIFIEDQA: Crossing format boundaries with a single QA system," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 1896–

1907. [Online]. Available: http://dx.doi.org/10.18653/v1/2020.findings-emnlp.171.

[40] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An open multilingual graph of general knowledge," en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017, ISSN: 2159-5399,2374-3468. [Online]. Available: http://dx.doi.org/10.1609/aaai.v31i1.11164.

[41] B. Y. Lin, S. Lee, R. Khanna, and X. Ren, "Birds have four legs?! NumerSense: Probing numerical commonsense knowledge of pre-trained language models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 6862–6868. [Online]. Available: http://dx.doi.org/10.18653/v1/2020.emnlp-main.557.

[42] A. Talmor et al., "CommonsenseQA 2.0: Exposing the limits of AI through gamification," *arXiv [cs.CL]*, 2022. eprint: 2201.05320 (cs.CL). [Online]. Available: http://arxiv.org/abs/2201.05320.

[43] T. Khot, P. Clark, M. Guerquin, P. Jansen, and A. Sabharwal, "QASC: A dataset for question answering via sentence composition," en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 8082–8090, 2020, ISSN: 2159-5399,2374-3468. [Online]. Available: http://dx.doi.org/10.1609/aaai.v34i05.6319.

[44] A. Chowdhery et al., "PaLM: Scaling language modeling with pathways," *arXiv [cs.CL]*, 2022. arXiv: 2204.02311 [cs.CL]. [Online]. Available: http://arxiv.org/abs/2204.02311.

[45] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant, "*did Aristotle use a laptop?*A question answering benchmark with implicit reasoning strategies," en, *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 346–361, 2021, ISSN: 2307-387X. [Online]. Available: https://doi.org/10.1162/tacl_a_00370.

[46] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[47] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," *arXiv [cs.CL]*, 2023. arXiv: 2307.09288 [cs.CL]. [Online]. Available: http://arxiv.org/abs/2307.09288.

[48] S. Roy and D. Roth, "Solving general arithmetic word problems," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1743–1752. [Online]. Available: http://dx.doi.org/10.18653/v1/D15-1202.

[49] S. Aggarwal, D. Mandowara, V. Agrawal, D. Khandelwal, P. Singla, and D. Garg, "Explanations for CommonsenseQA: New Dataset and Models," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, 2021, pp. 3050–3065. [Online]. Available: http://dx.doi.org/10.18653/v1/2021.acl-long.238.

[50] O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom, "E-SNLI: Natural language inference with natural language explanations," *Advances in Neural Information Processing Systems*, vol. 31, 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/4c7a167bb329bd92580a99ce422d6fa6-Abstract.html.

[51] M. Chen et al., "Evaluating large language models trained on code," *arXiv [cs.LG]*, 2021. arXiv: 2107.03374 [cs.LG]. [Online]. Available: http://arxiv.org/abs/2107.03374.

[52] A. Amini, S. Gabriel, S. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi, "MathQA: Towards interpretable math word problem solving with operation-based formalisms," in *Proceedings of the 2019 Conference of the North*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 2357–2367. [Online]. Available: http://dx.doi.org/10.18653/v1/N19-1245.

[53] C. Yang et al., "Large language models as optimizers," in *The Twelfth International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=Bb4VGOWELI.

[54] P. Zhou et al., "SELF-DISCOVER: Large language models self-compose reasoning structures," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=BROvXhmzYK&referrer=%5Bthe%20profile%20of%20Xiang%20Ren%5D(%2Fprofile%3Fid%3D~Xiang_Ren1).

[55] J. Wang, Q. Sun, X. Li, and M. Gao, "Boosting language models reasoning with chain-of-knowledge prompting," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Stroudsburg, PA, USA: Association for Computational Linguistics, 2024, pp. 4958–4981. [Online]. Available: https://aclanthology.org/2024.acl-long.271/.

[56] J. Hong, J. T. Wang, C. Zhang, L. I. Zhangheng, B. Li, and Z. Wang, "DP-OPT: Make large language model your privacy-preserving prompt engineer," in *The Twelfth International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=Ifz3IgsEPX.

[57] W. Tan, D. Chen, J. Xue, Z. Wang, and T. Chen, "Teaching-inspired integrated prompting framework: A novel approach for enhancing reasoning in large language models," *International Conference on Computational Linguistics*, O. Rambow et al., Eds., pp. 827–839, Oct. 2024. eprint: 2410.08068. [Online]. Available: https://aclanthology.org/2025.coling-industry.69/.

[58] J. Jung et al., "Maieutic prompting: Logically consistent reasoning with recursive explanations," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Stroudsburg, PA, USA: Association for Computational Linguistics, 2022, pp. 1266–1279. [Online]. Available: http://dx.doi.org/10.18653/v1/2022.emnlp-main.82.

[59] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving factuality and reasoning in language models through multiagent debate," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=zj7YuTE4t8.

[60] S. Hao et al., "Reasoning with language model is planning with world model," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, 2023, pp. 8154–8173. [Online]. Available: https://aclanthology.org/2023.emnlp-main.507.

[61] P. Wei, D. Dimitriadis, Y. Xu, and M. Shen, "Don't just demo, teach me the principles: A principle-based multi-agent prompting strategy for text classification," *arXiv [cs.CL]*, 2025. arXiv: 2502.07165 [cs.CL]. [Online]. Available: https://doi.org/10.48550/arXiv.2502.07165.

[62] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, "Synthetic prompting: Generating chain-of-thought demonstrations for large language models," in *Proceedings of the 40th International Conference on Machine Learning*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., ser. Proceedings of Machine Learning Research, vol. 202, PMLR, 2023, pp. 30 706–30 775. [Online]. Available: https://proceedings.mlr.press/v202/shao23a.html.

[63] Vicuna Team, *Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality*, en, https://lmsys.org/blog/2023-03-30-vicuna, Accessed: 2025-8-16.

[64] M. Abadi et al., "Deep learning with differential privacy," en, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA: ACM, 2016, ISBN: 9781450341394. [Online]. Available: http://dx.doi.org/10.1145/2976749.2978318.

[65] S. E. Robertson and S. Walker, "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval," en, in *SIGIR '94*, London: Springer London, 1994, pp. 232–241, ISBN: 9783540198895,9781447120995. [Online]. Available: http://dx.doi.org/10.1007/978-1-4471-2099-5_24.

[66] M. J. Hosseini, H. Hajishirzi, O. Etzioni, and N. Kushman, "Learning to solve arithmetic word problems with verb categorization," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 523–533. [Online]. Available: http://dx.doi.org/10.3115/v1/D14-1058.

[67] A. Patel, S. Bhattamishra, and N. Goyal, "Are NLP models really able to solve simple math word problems?" *arXiv [cs.CL]*, 2021. eprint: 2103.07191 (cs.CL). [Online]. Available: http://arxiv.org/abs/2103.07191.

[68] Z. Zhou et al., "Learning by analogy: Diverse questions generation in math word problem," *arXiv [cs.CL]*, 2023. eprint: 2306.09064 (cs.CL). [Online]. Available: http://arxiv.org/abs/2306.09064.

[69] W. Ling, D. Yogatama, C. Dyer, and P. Blunsom, "Program induction by rationale generation: Learning to solve and explain algebraic word problems," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2017, pp. 158–167. [Online]. Available: http://dx.doi.org/10.18653/v1/P17-1015.

[70] R. Battiti, "Maximum satisfiability problem," en, in *Encyclopedia of Optimization*, Cham: Springer Nature Switzerland, 2025, pp. 1–7, ISBN: 9783030546212. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-54621-2_364-1.

[71] S. Singh et al., "COM2SENSE: A commonsense reasoning benchmark with complementary sentences," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, 2021, pp. 883–898. [Online]. Available: http://dx.doi.org/10.18653/v1/2021.findings-acl.78.

[72] K. Valmeekam, M. Marquez, A. Olmo, S. Sreedharan, and S. Kambhampati, "PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change," *arXiv [cs.CL]*, 2022. arXiv: 2206.10498 [cs.CL]. [Online]. Available: http://arxiv.org/abs/2206.10498.

[73] R. Coulom, "Efficient selectivity and backup operators in monte-carlo tree search," en, in *Computers and Games*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 72–83, ISBN: 9783540755371,9783540755388. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-75538-8_7.

[74] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," en, in *Lecture Notes in Computer Science*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 282–293, ISBN: 9783540453758,9783540460565. [Online]. Available: http://dx.doi.org/10.1007/11871842_29.

[75] H. Touvron et al., "LLaMA: Open and efficient foundation language models," *arXiv [cs.CL]*, 2023. eprint: 2302.13971 (cs.CL). [Online]. Available: http://arxiv.org/abs/2302.13971.

[76] A. Saparov and H. He, "Language models are greedy reasoners: A systematic formal analysis of chain-of-thought," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, OpenReview.net, 2023. [Online]. Available: https://openreview.net/forum?id=qFVVBzXxR2V.

[77] N. Wu, M. Gong, L. Shou, S. Liang, and D. Jiang, "Large language models are diverse role-players for summarization evaluation," en, in *Natural Language Processing and Chinese Computing*, Cham: Springer Nature Switzerland, 2023, pp. 695–707, ISBN: 9783031446924,9783031446931. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-44693-1_54.

[78] A. Kong et al., "Better zero-shot reasoning with role-play prompting," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, K. Duh, H. Gomez, and S. Bethard, Eds., Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 4099–4113. [Online]. Available: https://aclanthology.org/2024.naacl-long.228.

[79] P. Chen, S. Zhang, and B. Han, "CoMM: Collaborative multi-agent, multi-reasoning-path prompting for complex problem solving," in *Findings of the Association for Computational Linguistics: NAACL 2024*, K. Duh, H. Gomez, and S. Bethard, Eds., Stroudsburg, PA, USA: Association for Computational Linguistics, Jun. 2024, pp. 1720–1738. [Online]. Available: http://dx.doi.org/10.18653/v1/2024.findings-naacl.112.

[80] Z. Xing et al., "When prompt engineering meets software engineering: CNL-P as natural and robust "APIs" for human-AI interaction," in *The Thirteenth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=SqoL14HDm0.

[81] C.-M. Chan et al., "ChatEval: Towards better LLM-based evaluators through multi-agent debate," *arXiv [cs.CL]*, 2023. arXiv: 2308.07201 [cs.CL]. [Online]. Available: http://arxiv.org/abs/2308.07201.

[82] P. H. Luz de Araujo, P. Röttger, D. Hovy, and B. Roth, "Principled personas: Defining and measuring the intended effects of persona prompting on task performance," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language*

Processing, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds., Suzhou, China: Association for Computational Linguistics, 2025, pp. 26 845–26 874. [Online]. Available: http://dx.doi.org/10.18653/v1/2025.emnlp-main.1364.

[83] H. K. Choi and Y. Li, "PICLe: Eliciting diverse behaviors from large language models with persona in-context learning," in *Proceedings of the 41st International Conference on Machine Learning*, R. Salakhutdinov et al., Eds., vol. 235, PMLR, 2024, pp. 8722–8739. [Online]. Available: https://proceedings.mlr.press/v235/choi24e.html.

[84] T. Liang et al., "Encouraging divergent thinking in large language models through multi-agent debate," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, 2024, pp. 17 889–17 904. [Online]. Available: http://dx.doi.org/10.18653/v1/2024.emnlp-main.992.

[85] Z. Wang, S. Mao, W. Wu, T. Ge, F. Wei, and H. Ji, "Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, K. Duh, H. Gomez, and S. Bethard, Eds., Mexico City, Mexico: Association for Computational Linguistics, 2024, pp. 257–279. [Online]. Available: https://aclanthology.org/2024.naacl-long.15.

[86] ClickUp, *RISEN: 5 steps to build context-rich AI prompts*, https://clickup.com/general-resources/playbooks/ai-prompts, Accessed: 2026-3-17.

[87] A. Cre, *AI prompt engineering in commercial real estate: A useful framework*, en, https://www.adventuresincre.com/ai-cre-prompt-engineering/, Accessed: 2026-3-17, 2023.

[88] Gemini Team et al., "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv [cs.CL]*, 2024. eprint: 2403.05530 (cs.CL). [Online]. Available: http://arxiv.org/abs/2403.05530.

[89] OpenAI et al., "GPT-4o system card," *arXiv [cs.CL]*, 25 10 2024. arXiv: 2410.21276 [cs.CL]. [Online]. Available: http://arxiv.org/abs/2410.21276.

[90] A. Dubey et al., "The llama 3 herd of models," *ArXiv preprint*, vol. abs/2407.21783, 2024. arXiv: 2407.21783 [cs.AI]. [Online]. Available: http://arxiv.org/abs/2407.21783.

[91] Anthropic, *The claude 3 model family: Opus, sonnet, haiku*, en, https://www.semanticscholar.org/paper/The-Claude-3-Model-Family%3A-Opus%2C-Sonnet%2C-Haiku/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627, Accessed: 2025-1-24, 2024.

[92] C. K. Sah, X. Lian, T. Xu, and L. Zhang, "FairEval: Evaluating fairness in LLM-based recommendations with personality awareness," *arXiv [cs.IR]*, 2025. eprint: 2504.07801 (cs.IR). [Online]. Available: http://arxiv.org/abs/2504.07801.

[93] A. Madaan, K. Hermann, and A. Yazdanbakhsh, "What makes chain-of-thought prompting effective? a counterfactual study," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, 2023, pp. 1448–1535. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.101.

[94] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," *arXiv [cs.AI]*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., pp. 8634–8652, 2023. arXiv: 2303.11366 [cs.AI]. [Online]. Available: http://arxiv.org/abs/2303.11366.

[95] K. Gopalakrishnan et al., "Topical-chat: Towards knowledge-grounded open-domain conversations," *arXiv [cs.CL]*, 2023. eprint: 2308.11995 (cs.CL). [Online]. Available: http://arxiv.org/abs/2308.11995.

Appendix A

Appendix

A.1 Statistical Verification Results

A.1.1 Pairwise Comparison Between Cumulative Thinking Prompt and Baselines

Proposed vs. Baseline	Mean Diff	p-value	Significant	Cohen's d
Cumulative Thinking Prompt $_{\text{wins}}$ Direct Prompt	0.3436	< 0.001	Yes	2.52
Cumulative Thinking Prompt $_{\text{wins}}$ Excellent Professor	0.3692	< 0.001	Yes	2.19
Cumulative Thinking Prompt $_{\text{wins}}$ Step-by-Step	0.2765	< 0.001	Yes	2.49

Cumulative Thinking Prompt significantly outperforms all three baseline prompt methods.

A.1.2 Pairwise Comparison between AUDIT Prompt and Baselines

Proposed vs. Baseline	Mean Diff	p-value	Significant	Cohen's d
AUDIT V1 $_{\text{wins}}$ Direct Prompt	0.2077	< 0.001	Yes	1.57
AUDIT V1 $_{\text{wins}}$ Persona-Aware Prompt	0.2303	< 0.001	Yes	1.29
AUDIT V1 $_{\text{wins}}$ Step-by-Step Prompt	0.0938	0.00663	Yes	0.73
AUDIT V1 $\approx$ Cumulative Thinking Prompt	-0.0008	0.97827	No	-0.01
AUDIT V2 $_{\text{wins}}$ Direct Prompt	1.0059	< 0.001	Yes	4.45
AUDIT V2 $_{\text{wins}}$ Persona-Aware Prompt	1.0285	< 0.001	Yes	4.20
AUDIT V2 $_{\text{wins}}$ Step-by-Step Prompt	0.8920	< 0.001	Yes	4.29
AUDIT V2 $_{\text{wins}}$ Cumulative Thinking Prompt	0.7974	< 0.001	Yes	4.13
AUDIT V3 $_{\text{wins}}$ Direct Prompt	1.1618	< 0.001	Yes	4.55
AUDIT V3 $_{\text{wins}}$ Persona-Aware Prompt	1.1843	< 0.001	Yes	4.27
AUDIT V3 $_{\text{wins}}$ Step-by-Step Prompt	1.0478	< 0.001	Yes	4.41
AUDIT V3 $_{\text{wins}}$ Cumulative Thinking Prompt	0.9532	< 0.001	Yes	4.28

Iterative AUDIT V3 and V2 variations significantly outperform all baseline methods, while non-iterative AUDIT V1 performs nearly identical to the Cumulative Thinking Prompt of study 1.

A.1.3 Pairwise Comparison between Professor-Rank-Aware Prompt and Baseline

Proposed vs. Baseline	Mean Diff	p-value	Significant	Cohen's d
Distinguished Professor $_{\text{wins}}$ Excellent Professor	0.5711	< 0.001	Yes	5.89
Emeritus Professor $_{\text{wins}}$ Excellent Professor	0.3437	< 0.001	Yes	4.16
Full Professor $_{\text{wins}}$ Excellent Professor	0.2800	< 0.001	Yes	3.16
Research Professor $_{\text{wins}}$ Excellent Professor	0.1110	< 0.001	Yes	1.59
Assistant Professor $_{\text{defeated by}}$ Excellent Professor	-0.2311	< 0.001	Yes	-3.95
Associate Professor $_{\text{defeated by}}$ Excellent Professor	-0.0507	0.00305	Yes	-0.81
Lecturer $_{\text{defeated by}}$ Excellent Professor	-0.2574	< 0.001	Yes	-2.77
Teaching Professor $_{\text{defeated by}}$ Excellent Professor	-0.0893	< 0.001	Yes	-1.61

Higher ranking professor prompt variations performed better than the baseline, junior ranking professor variations performed worse than the baseline, validating that fine-grained persona settings systematically influence the generation quality in alignment with academic hierarchy.

A.2 Generation Demonstrations for Three Proposed Prompt Methods

The following LLM variation was applied in each of three proposed prompt method experiments.

LLMs	Variations
DeepSeek	V 3.2

A.2.1 Definition Demonstrations for Study 1: Cumulative Thinking Prompt

Category	Math Term	Math Subject
Content	Rank	Linear Algebra

Definition Variation - Cumulative Thinking Prompt - Ranked 1st

recall your understanding about rank in linear algebra from essential to abstract in sequential 5 times, then to generate definition of rank in linear algebra with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format, based on all 5 previous sequential thinking before generating the mathematical definition.

In linear algebra, the rank of a matrix $A \in \mathbb{R}^{m \times n}$ is a fundamental invariant that captures its inherent dimensionality and the potency of the linear transformation it represents. Formally, it is defined as the dimension of the column space (or, equivalently, the row space) of A, denoted $\text{rank}(A) = \dim(\text{Col}(A)) = \dim(\text{Row}(A))$. This equals the maximum number of linearly independent column vectors $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ in A, or equivalently, the number of pivot positions in its row echelon form. The rank determines the structure of solutions to $A\mathbf{x} = \mathbf{b}$ via the Rouché–Capelli theorem and is intrinsically linked to the dimension of the image of the associated map $T_A : \mathbb{R}^n \to \mathbb{R}^m, T_A(\mathbf{x}) = A\mathbf{x}$, where $\text{rank}(A) = \dim(\text{Im}(T_A))$. The Rank-Nullity Theorem, $\text{rank}(A) + \text{nullity}(A) = n$, where $\text{nullity}(A) = \dim(\ker(T_A))$, provides a complete dimensional decomposition of the domain. A matrix has full rank if $\text{rank}(A) = \min(m, n)$; if square and of full rank, it is invertible.

Definition Variation - Step-by-Step Prompt - Ranked 2nd

let's think step by step, definition of rank in linear algebra with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

In linear algebra, the rank of a matrix $A \in \mathbb{R}^{m \times n}$ is a fundamental measure of its linear independence and dimensionality. Formally, it is defined as the dimension of the vector space generated (or spanned) by its columns, known as the column space $\text{Col}(A)$. Equivalently, it is the dimension of the row space $\text{Row}(A)$, a result encapsulated by the theorem $\text{rank}(A) = \text{rank}(A^T)$. Mathematically, if $A = [\mathbf{a}_1 \cdots \mathbf{a}_n]$, then $\text{rank}(A) = \dim(\text{span}\{\mathbf{a}_1, \ldots, \mathbf{a}_n\})$. This integer value, $r = \text{rank}(A)$, satisfies $0 \leq r \leq \min(m, n)$. The rank can be determined via Gaussian elimination as the number of pivot positions in the matrix's row echelon form, or as the number of non-zero singular values in its singular value decomposition SVD, $A = U\Sigma V^T$. Crucially, the rank characterizes the solutions to $A\mathbf{x} = \mathbf{b}$ via the Rank-Nullity Theorem: $\text{rank}(A) + \text{nullity}(A) = n$, where $\text{nullity}(A) = \dim(\text{Nul}(A))$ is the dimension of the null space (kernel). A matrix is full rank if $\text{rank}(A) = \min(m, n)$, indicating its rows and columns are linearly independent.

Definition Variation - Direct Prompt - Ranked 3rd

definition of rank in linear algebra with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

In linear algebra, the rank of a matrix $A \in \mathbb{R}^{m \times n}$ is a fundamental measure of its linear independence and dimensionality. Formally, it is defined as the maximum number of linearly independent column vectors (column rank) or row vectors (row rank), with these two values always being equal. Mathematically, the column rank is the dimension of the column space $\text{Col}(A)$, which is the subspace of $\mathbb{R}^m$ spanned by the columns of A, i.e., $\text{rank}(A) = \dim(\text{Col}(A))$. Equivalently, it is the dimension of the row space $\text{Row}(A)$. The rank can also be characterized as the number of pivot positions in the row echelon form (or reduced row echelon form) of A, or as the number of nonzero singular values in its singular value decomposition SVD. A key theorem is the rank-nullity theorem: $\text{rank}(A) + \text{nullity}(A) = n$, where $\text{nullity}(A) = \dim(\text{Nul}(A))$ is the dimension of the null space (kernel). The rank determines the solvability of $A\mathbf{x} = \mathbf{b}$; a system is consistent if and only if $\text{rank}(A) = \text{rank}([A\ \mathbf{b}])$. Furthermore, a matrix is full rank if $\text{rank}(A) = \min(m, n)$; otherwise, it is rank-deficient.

Definition Variation - Excellent-Professor-Aware Prompt - Ranked 4th

you are an excellent math professor who are good at math teaching, generate definition of rank in linear algebra with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

In linear algebra, the rank of a matrix $A \in \mathbb{R}^{m \times n}$ is a fundamental invariant that measures the dimension of the vector spaces generated by its rows and columns. Formally, the **column rank** is the dimension of the column space $\text{Col}(A)$, which is the subspace of $\mathbb{R}^m$ spanned by the columns of A, i.e., $\text{rank}(A) = \dim(\text{Col}(A))$. Equivalently, the **row rank** is the dimension of the row space $\text{Row}(A) \subseteq \mathbb{R}^n$. A key theorem states that column rank equals row rank, so we simply refer to the rank of A. The rank can be determined via Gaussian elimination: it is the number of pivots (or nonzero rows) in any row-echelon form of A. If $\text{rank}(A) = r$, then A has exactly r linearly independent columns and rows. Furthermore, rank characterizes solutions to $A\mathbf{x} = \mathbf{b}$: the system is consistent if and only if $\text{rank}(A) = \text{rank}([A|\mathbf{b}])$, and the solution set has dimension $n - r$ when consistent. Rank also satisfies $\text{rank}(A) \leq \min(m, n)$, with equality defining a full-rank matrix.

Term-Specific Definition Strengths and Weaknesses
Study 1: Cumulative Thinking Prompt

Prompt Variation with Ranking	Definition Strengths	Definition Weaknesses
Cumulative Thinking Prompt (Ranked 1st)	Highest abstraction and integration with linear transformation theory. Elegant, precise, and theoretically rich with perfect structural coherence.	None significant; it is the benchmark definition.
Step-by-Step Prompt (Ranked 2nd)	Excellent logical flow from definition to properties. Balanced and pedagogically sound with clear notation.	Slightly less integrated with linear transformation viewpoint compared to the top definition.
Direct Prompt (Ranked 3rd)	Comprehensive, covers multiple characterizations (column/row rank, pivots, SVD, rank-nullity, solvability). Excellent theoretical correctness.	Slightly less structured flow than others; information is comprehensive but presented in a denser paragraph.
Excellent-Professor-Aware Prompt (Ranked 4th)	Clear, concise, and well-structured. Effectively links rank to system solvability and solution set dimension.	Slightly less complete than others (omits SVD, explicit rank-nullity statement). Minor notation inconsistency.

Notes: Strength and weakness interpretations were created by chosen large language model against predefined math term definition quality metrics. Different metrics will change the evaluation interpretations with corresponding ranking, which can be reconfigured to meet different downstream task requirements.

Term-Specific Quality Metric Scores and Rankings
Study 1: Cumulative Thinking Prompt

Quality Metrics	CTP	SSP	DP	EP
Term-Specific Average Score	**9.45**	**9.05**	**9.00**	**8.55**
Term-Specific Ranking	**1**	**2**	**3**	**4**

Abbreviations: CTP = Cumulative Thinking Prompt, DP = Direct Prompt, SSP = Step-by-Step Prompt, EP - Excellent-Professor-Aware Prompt

A.2.2 Definition Demonstrations for Study 2: AUDIT Prompt

Category	Math Term	Math Subject
Content	Slope Fields	Calculus

Definition Variation - AUDIT Prompt V3 - Ranked 1st

definition of slope fields in calculus with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the first paragraph as "definition_1", then suggest math definition refinement suggestions in the second paragraph as "quality_audit_1", then generate the first refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the third paragraph as "definition_2", then suggest math definition refinement suggestions to further refine the "definition_2" in the fourth paragraph as "quality_audit_2", then generate the second refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the fifth paragraph as "definition_3:". Make sure each of the sequential definitions should be optimized to be more strictly aligned with the standard math definitions of college level mathematics lecture notes and textbooks.

A slope field, also called a direction field, is the graphical representation associated with a first-order differential equation $y' = f(x, y)$. Formally, given a function $f : D \to \mathbb{R}$ where D is a connected domain in $\mathbb{R}^2$, the slope field is the assignment of a slope $m = f(x, y)$ to each point $(x, y) \in D$, typically visualized by drawing a short line segment with that slope centered at the point. This field is a discrete approximation of the continuous direction field defined by the vector function $\mathbf{v}(x, y) = \langle 1, f(x, y) \rangle$, whose vectors are tangent to the graph of any solution. A solution to the ODE is a differentiable function $y = \phi(x)$ on an interval I such that $\phi'(x) = f(x, \phi(x))$ for all $x \in I$; its graph is an integral curve everywhere tangent to the slope field. Under the conditions of the Fundamental Existence and Uniqueness Theorem for ODEs (if f and $\partial f/\partial y$ are continuous on D), through each point in D there passes exactly one integral curve, making the slope field a complete qualitative portrait of all solutions. Slope fields facilitate the analysis of equilibrium solutions, stability, and long-term behavior, and underpin the geometric intuition for numerical solution techniques like Euler's method, which follows the local linear directions defined by the field.

Definition Variation - AUDIT Prompt V2 - Ranked 2nd

definition of slope fields in calculus with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the first paragraph as "definition_1", then suggest math definition refinement suggestions in the second paragraph as "quality_audit_1", then generate the first refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the third paragraph as "definition_2", then suggest math definition refinement suggestions to further refine the "definition_2" in the fourth paragraph as "quality_audit_2", then generate the second refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the fifth paragraph as "definition_3:". Make sure each of the sequential definitions should be optimized to be more strictly aligned with the standard math definitions of college level mathematics lecture notes and textbooks.

In the study of ordinary differential equations, a slope field for a first-order ODE $dy/dx = f(x, y)$ is constructed on a rectangular domain $D \subset \mathbb{R}^2$. At a finite set of sample points $(x_i, y_i) \in D$, a short line segment of uniform length is drawn centered at the point, with a slope precisely equal to the numerical value $m_i = f(x_i, y_i)$. This slope represents the derivative, dictating the direction a solution curve $y = \phi(x)$ must have if its graph passes through (x_i, y_i). An integral curve (or solution curve) of the ODE is a differentiable function $\phi(x)$ defined on an interval I such that for all $x \in I$, $(x, \phi(x)) \in D$ and $\phi'(x) = f(x, \phi(x))$; its graph is everywhere tangent to the segments of the slope field. The field visualizes the family of all such potential solutions. For autonomous equations, where f is independent of x ($dy/dx = f(y)$), the slope field is invariant under horizontal translation, revealing equilibrium solutions at roots of $f(y) = 0$. Isoclines, curves satisfying $f(x, y) = k$ for constant k, aid in sketching the field systematically.

Definition Variation - Cumulative Thinking Prompt - Ranked 3rd

recall your understanding about slope fields in calculus from essential to abstract in sequential 5 times, then to generate definition of slope fields in calculus with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format, based on all 5 previous sequential thinking before generating the mathematical definition.

In calculus, a slope field (or direction field) is a graphical representation of a first-order ordinary differential equation of the form $\frac{dy}{dx} = f(x, y)$. It is constructed by selecting a grid of points (x_i, y_j) in the xy-plane and, at each point, drawing a short line segment (a minitangent) whose slope is precisely the value $m_{ij} = f(x_i, y_j)$. This field visualizes the differential equation by showing the slope of potential solution curves at every point in the domain. A solution curve $y = \phi(x)$ to the ODE is a function whose graph is tangent to these segments at every point along it; that is, $\phi'(x) = f(x, \phi(x))$ for all x in its interval. Thus, the slope field provides a qualitative picture of the family of all possible solutions, allowing for the analysis of equilibrium points (where $f(x, y) = 0$), isoclines (curves where $f(x, y)$ is constant), and the long-term behavior of the system without requiring an explicit analytic solution.

Definition Variation - AUDIT Prompt V1 - Ranked 4th

definition of slope fields in calculus with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the first paragraph as “definition_1”, then suggest math definition refinement suggestions in the second paragraph as “quality_audit_1”, then generate the first refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the third paragraph as “definition_2”, then suggest math definition refinement suggestions to further refine the “definition_2” in the fourth paragraph as “quality_audit_2”, then generate the second refined definition with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format in the fifth paragraph as “definition_3:”. Make sure each of the sequential definitions should be optimized to be more strictly aligned with the standard math definitions of college level mathematics lecture notes and textbooks.

In calculus, a slope field (or direction field) is a graphical representation of a first-order ordinary differential equation (ODE) of the form $\frac{dy}{dx} = f(x, y)$. At each point (x_0, y_0) in a chosen region of the xy-plane, a short line segment (or arrow) is drawn whose slope is equal to the value of $f(x_0, y_0)$. This slope is the derivative $\frac{dy}{dx}$ evaluated at that point, indicating the instantaneous rate of change of a potential solution curve passing through (x_0, y_0). The collection of all such segments provides a visual portrait of the family of all possible solution curves to the differential equation without solving it analytically. By following the local directions indicated by the slopes, one can sketch approximate solution curves (integral curves) that are tangent to the field at every point. Slope fields are particularly useful for analyzing the qualitative behavior of solutions, such as identifying equilibrium solutions (where $f(x, y) = 0$), stability, and general trends, especially when an explicit solution is difficult or impossible to obtain.

Definition Variation - Direct Prompt - Ranked 4th

definition of slope fields in calculus with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

In calculus, a slope field (or direction field) is a graphical representation of the solutions to a first-order ordinary differential equation (ODE) of the form $\frac{dy}{dx} = f(x, y)$. It is constructed by evaluating the function $f(x, y)$ at a grid of points (x_i, y_i) in the xy-plane and drawing a short line segment at each point with a slope equal to the value $f(x_i, y_i)$. These segments are tangent to the solution curve (or integral curve) passing through that point. Consequently, the slope field visualizes the family of all possible solutions to the ODE without solving it analytically. By following the direction indicated by the segments, one can sketch approximate solution curves that start from a given initial condition (x_0, y_0). This geometric tool is particularly valuable for analyzing the behavior of differential equations that are difficult or impossible to solve explicitly, allowing for the study of equilibrium solutions (where $f(x, y) = 0$), stability, and long-term trends. The pattern of the field reveals how the rate of change dy/dx depends on both the independent variable x and the dependent variable y.

Definition Variation - Step-by-Step Prompt - Ranked 6th

let's think step by step, definition of slope fields in calculus with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

In calculus, a slope field (or direction field) is a graphical representation of the solutions to a first-order differential equation of the form $\frac{dy}{dx} = f(x, y)$. It is constructed by selecting a grid of points (x, y) in the xy-plane and drawing a short line segment at each point with a slope equal to the value of $f(x, y)$ at that point. These segments are tangent to the solution curve, or integral curve, that passes through that point. Consequently, the slope field visualizes the family of all possible solutions to the differential equation, as sketching a curve that follows the direction of the line segments at every point yields a particular solution corresponding to a specific initial condition (x_0, y_0). The pattern revealed by the field indicates the behavior of solutions, such as equilibrium points where $f(x, y) = 0$, and regions of increasing or decreasing slopes. This geometric tool is particularly valuable for analyzing differential equations that are difficult or impossible to solve analytically, providing intuitive insight into the long-term trends and stability of dynamical systems modeled by the equation.

Definition Variation - Excellent-Professor-Aware Prompt - Ranked 7th

you are an excellent math professor who are good at math teaching, generate definition of slope fields in calculus with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

In calculus, a slope field (or direction field) is a graphical representation of the solutions to a first-order differential equation of the form $\frac{dy}{dx} = f(x, y)$. It is constructed by selecting a grid of points (x, y) in the xy-plane and drawing a short line segment at each point with a slope equal to the value of $f(x, y)$ at that point. These segments are tangent to the solution curve, or integral curve, that passes through that point. Therefore, the slope field visually depicts the family of all possible solution curves to the differential equation without solving it analytically. By following the direction indicated by the segments, one can sketch approximate solution curves that satisfy a given initial condition $y(x_0) = y_0$. This tool is particularly valuable for understanding the behavior of solutions, identifying equilibrium solutions (where $f(x, y) = 0$), and analyzing stability in cases where an explicit solution is difficult or impossible to obtain. The pattern formed by the segments illustrates how the rate of change dy/dx depends on both the independent variable x and the dependent variable y.

Term-Specific Definition Strengths and Weaknesses
Study 2: AUDIT Prompt

Prompt Variation with Ranking	Definition Strengths	Definition Weaknesses
AUDIT Prompt V3 (Ranked 1st)	Maximum rigor and completeness. Formal definition of function domain, links to vector field and existence/uniqueness theorem.	Extremely dense and formal; may be overly advanced for a first introduction in some textbooks.
AUDIT Prompt V2 (Ranked 2nd)	High formal precision, explicit domain specification, and covers autonomous equations and isoclines.	Very high level may be less immediately accessible to some learners without context.
Cumulative Thinking Prompt (Ranked 3rd)	Concise yet precise. Introduces formal elements like minitangents and isoclines effectively.	Slightly less explicit about domain/interval conditions for solutions compared to AUDIT versions.
AUDIT Prompt V1 (Ranked 4th)	Clear, thorough, and well-balanced between intuition and formal description.	Less formally rigorous than V2/V3 regarding domain and explicit solution function definition.
Direct Prompt (Ranked 4th)	Clear, balanced, and accessible. Covers practical use and qualitative analysis well.	Slightly less formal than top-tier definitions. Could be more explicit about domain and solution function definition.
Step-by-Step Prompt (Ranked 6th)	Excellent logical flow from construction to interpretation. Strong explanatory hierarchy.	Similar to persona_aware; lacks the highest level of formal mathematical specification.
Excellent-Professor-Aware Prompt (Ranked 7th)	Clear, well-structured, and pedagogically sound. Good flow and emphasis on understanding.	Slightly less formal and precise than others. Lacks explicit mention of domain or formal solution criteria.

Notes: Strength and weakness interpretations were created by chosen large language model against predefined math term definition quality metrics. Different metrics will change the evaluation interpretations with corresponding ranking, which can be reconfigured to meet different downstream task requirements.

Term-Specific Quality Metric Scores and Rankings
Study 2: AUDIT Prompt

Quality Metrics	AUDIT V3	AUDIT V2	CTP	AUDIT V1	DP	SSP	EP
Term-Specific Average Score	**9.77**	**9.23**	**8.27**	**8.05**	**8.05**	**7.86**	**7.82**
Term-Specific Ranking	**1**	**2**	**3**	**4**	**4**	**6**	**7**

Abbreviations: CTP = Cumulative Thinking Prompt, DP = Direct Prompt, SSP = Step-by-Step Prompt, EP - Excellent-Professor-Aware Prompt

A.2.3 Definition Demonstrations for Study 3: Professor-Rank-Aware Prompt

Category	Math Term	Math Subject
Content	Function	Algebra

Definition Variation - Distinguished-Professor-Aware Prompt - Ranked 1st

you are a distinguished professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In algebra, a function f from a set A to a set B, denoted $f : A \to B$, is a specific type of relation that assigns to each element x in the domain A exactly one element y in the codomain B. This unique assignment is written as $y = f(x)$, where x is the independent variable (or input) and y is the dependent variable (or output). Formally, a function is a set of ordered pairs (x, y) with $x \in A$ and $y \in B$, such that no two distinct ordered pairs have the same first element; that is, if $(x_1, y_1) \in f$ and $(x_1, y_2) \in f$, then $y_1 = y_2$. This condition ensures the crucial property of well-definedness, meaning for every input there is a single, unambiguous output. The set of all possible inputs is the domain $\text{dom}(f) = A$, the set of all actual outputs is the range $\{f(x) \mid x \in A\} \subseteq B$, and the entire set B is the codomain. Functions are fundamental for modeling mathematical relationships where one quantity deterministically depends on another.

Definition Variation - Full-Professor-Aware Prompt - Ranked 2nd

you are a full professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In algebra, a function f from a set A to a set B, denoted $f : A \to B$, is a relation that assigns to each element x in the domain A exactly one element y in the codomain B. This unique element is called the image of x under f and is written as $y = f(x)$. Formally, a function is a set of ordered pairs (x, y) with $x \in A$ and $y \in B$, such that no two distinct pairs have the same first element: if $(x_1, y_1) \in f$ and $(x_2, y_2) \in f$, then $x_1 = x_2$ implies $y_1 = y_2$. This condition ensures the unambiguous assignment required for a functional relationship. The set of all first elements is the domain $\text{dom}(f) = A$, and the set of all actual outputs is the range $\text{ran}(f) = \{f(x) \mid x \in A\} \subseteq B$. Functions are fundamental for modeling dependencies between quantities, where each input from the domain determines a specific output in the codomain.

Definition Variation - Emeritus-Professor-Aware Prompt - Ranked 3rd

you are an emeritus professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In algebra, a function is formally defined as a relation between two sets, typically called the domain D and the codomain C, that assigns to each element x in the domain exactly one element y in the codomain. This is denoted by $f : D \to C$, where f is the function's name. The unique element assigned to x is written as $f(x)$, read as "f of x", and is called the image of x under f or the value of f at x. The set of all such images, $\{f(x) \mid x \in D\}$, is called the range or image of the function, which is a subset of the codomain ($\text{Im}(f) \subseteq C$). The defining property—that each input has a single output—is expressed mathematically as: if $(x, y_1) \in f$ and $(x, y_2) \in f$, then $y_1 = y_2$. Functions are fundamental for modeling dependencies between quantities, where the independent variable x from the domain determines the dependent variable $y = f(x)$ in the codomain according to a specific rule or mapping.

Definition Variation - Research-Professor-Aware Prompt - Ranked 4th

you are a research professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In algebra, a function f from a set A to a set B, denoted $f : A \to B$, is a specific type of relation that assigns to each element x in the domain A exactly one element y in the codomain B. This unique assignment is written as $y = f(x)$, where x is the independent variable (or input) and y is the dependent variable (or output). Formally, a function is a set of ordered pairs (x, y) with $x \in A$ and $y \in B$, such that no two distinct ordered pairs have the same first element: if $(x_1, y_1) \in f$ and $(x_1, y_2) \in f$, then $y_1 = y_2$. The set of all possible inputs is the domain $\text{dom}(f) = A$, the set of all actual outputs is the range $\text{ran}(f) = \{f(x) \mid x \in A\} \subseteq B$, and the entire set B is the codomain. Functions are fundamental for modeling mathematical relationships where each input determines a single, unambiguous output, such as in polynomial functions $f(x) = a_n x^n + \cdots + a_1 x + a_0$, rational functions, or exponential functions.

Definition Variation - Excellent-Professor-Aware Prompt - Ranked 5th

you are an excellent math professor who are good at math teaching, generate definition of [term] in [subject] with detailed mathematical explanations in one paragraph of approximately 200 words with used math symbols in LaTeX format.

In algebra, a function is formally defined as a relation between two sets, typically called the domain D and the codomain C, that assigns to each element x in the domain exactly one element y in the codomain. This is denoted by $f : D \to C$, where f is the function's name. The unique element assigned to x is written as $f(x)$, read as "f of x", and is called the image of x under f. The core defining property is that if $(x, y_1) \in f$ and $(x, y_2) \in f$, then it must be that $y_1 = y_2$; this is the condition of being well-defined or single-valued. The set of all actual outputs, $\{f(x) \mid x \in D\}$, is called the range or image of the function, which is a subset of the codomain ($\text{Im}(f) \subseteq C$). Functions are fundamental for modeling dependencies where one quantity (the independent variable from the domain) determines a unique resulting quantity (the dependent variable in the codomain).

Definition Variation - Associate-Professor-Aware Prompt - Ranked 6th

you are an associate professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In algebra, a function f from a set A to a set B, denoted $f : A \to B$, is a relation that assigns to each element x in the domain A exactly one element y in the codomain B. This unique element is called the image of x under f and is written as $y = f(x)$. Formally, a function is a set of ordered pairs (x, y) with $x \in A$ and $y \in B$, such that no two distinct pairs have the same first element: if $(x, y_1) \in f$ and $(x, y_2) \in f$, then $y_1 = y_2$. This ensures the critical property of well-definedness, meaning every input yields a single, unambiguous output. The set of all such outputs, $\{f(x) \mid x \in A\}$, is called the range or image of the function, which is a subset of the codomain B. Functions are fundamental for modeling dependencies between quantities, where x is the independent variable and $f(x)$ is the dependent variable.

Definition Variation - Teaching-Professor-Aware Prompt - Ranked 7th

you are a teaching professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In algebra, a function f from a set A to a set B, denoted $f : A \to B$, is a relation that assigns to each element x in the domain A exactly one element y in the codomain B. This unique element is called the image of x under f and is written as $y = f(x)$. Formally, a function is a set of ordered pairs (x, y) with $x \in A$ and $y \in B$, such that no two distinct pairs have the same first element: if $(x_1, y_1) \in f$ and $(x_1, y_2) \in f$, then $y_1 = y_2$. This ensures the critical property of well-definedness, meaning every input yields a single, unambiguous output. The set of all possible outputs, called the range, is the subset of the codomain B given by $\{f(x) \mid x \in A\}$. Functions are fundamental for modeling relationships between quantities, where the independent variable x determines the dependent variable $f(x)$ according to a specific rule, such as a polynomial $f(x) = ax^2 + bx + c$, an exponential $g(x) = e^x$, or a rational expression $h(x) = \frac{1}{x-1}$ (with domain restrictions).

Definition Variation - Assistant-Professor-Aware Prompt - Ranked 8th

you are an assistant professor teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In algebra, a function is a specific type of relation between two sets, typically called the domain and the codomain, that assigns to each element in the domain exactly one element in the codomain. Formally, a function f from a set A (the domain) to a set B (the codomain) is a rule or correspondence such that for every $x \in A$, there exists a unique $y \in B$ with $y = f(x)$. This uniqueness condition means f is well-defined: if $x_1 = x_2$, then $f(x_1) = f(x_2)$. The set of all possible outputs is called the range or image, defined as $\{f(x) \mid x \in A\} \subseteq B$. Functions are often denoted by $f : A \to B$, and can be represented algebraically by expressions like $f(x) = x^2 + 3$, graphically as curves or points in the Cartesian plane satisfying the vertical line test, or as sets of ordered pairs $\{(x, f(x)) \mid x \in A\}$ where no two distinct pairs have the same first element. Key properties include injectivity (one-to-one: $f(x_1) = f(x_2) \implies x_1 = x_2$), surjectivity (onto: for every $y \in B$, there exists $x \in A$ such that $f(x) = y$), and bijectivity (both one-to-one and onto), which determines invertibility.

Definition Variation - Lecture-Aware Prompt - Ranked 9th

you are a lecturer teaching college level mathematics, generate the definition of [term] in [subject] with detailed mathematical explanations in one paragraph with used math symbols in LaTeX format, make sure to fully and completely generate the definition, as truncated definition is not acceptable.

In algebra, a function f from a set A to a set B, denoted $f : A \to B$, is a specific rule or correspondence that assigns to each element x in the domain A exactly one element y in the codomain B. This assigned element is called the image of x under f and is written as $y = f(x)$. The fundamental defining property is that if $(x, y_1) \in f$ and $(x, y_2) \in f$, then $y_1 = y_2$, ensuring the output is unique for a given input. The set of all such ordered pairs $(x, f(x))$ is called the graph of the function. The range of f, a subset of the codomain B, is the set $\{f(x) \mid x \in A\}$ of all actual output values. Functions are the primary tools for modeling relationships between quantities, where a change in the independent variable x determines a specific change in the dependent variable y.

Term-Specific Definition Strengths and Weaknesses
Study 3: Professor-Rank-Aware Prompt

Prompt Variation with Ranking	Definition Strengths	Definition Weaknesses
Distinguished Professor (Ranked 1st)	Exemplary balance of formal precision, completeness, and clarity. Explicitly defines domain, codomain, range, and well-definedness with flawless flow.	Minimal. Slightly verbose compared to the most concise versions.
Full Professor (Ranked 2rd)	Highly precise and logically ordered. Excellent formal statement of the uniqueness condition and clear distinction of domain, codomain, and range.	Very minor: the formal condition uses $x_1 = x_2$ implication, which is slightly less direct than the x = x version.
Emeritus Professor (Ranked 3rd)	Clear, correct, and well-structured. Strong formal definition with consistent terminology and excellent academic tone.	Slightly less precise in variable naming (D, C vs. A, B) and the formal condition statement is less direct than the top versions.
Research Professor (Ranked 4th)	Solid formal definition with good precision. Includes helpful contextual examples (polynomial, exponential functions).	Examples, while helpful for context, slightly reduce abstraction. Slightly less fluid hierarchy than top versions.
Excellent Professor (Ranked 5th)	Clear, formally correct, and well-phrased. Good emphasis on the core well-definedness condition.	Good but not exceptional in any single metric. Lacks the distinctive completeness or exemplary flow of the top definitions.
Associate Professor (Ranked 6th)	Clear, correct, and concise. Effectively covers core concepts (domain, codomain, range, well-definedness) without extraneous detail.	Slightly less detailed than top-tier definitions. Does not explicitly separate the concepts of 'rule' and 'set of pairs' as clearly.
Teacher Professor (Ranked 7th)	Clear and accessible, with concrete examples and explicit mention of domain restrictions, aiding comprehension.	Examples reduce abstraction level. Slightly less formal precision and rigor than higher-ranked definitions.
Assistant Professor (Ranked 8th)	Most comprehensive, introducing key properties (injectivity, surjectivity) and multiple representations (algebraic, graphical, set).	Lower abstraction due to extended discussion of properties and representations. Some terminology inconsistency (switches between 'rule' and formal set definition).
Lecture-Aware-Prompt (Ranked 9th)	Clear and straightforward, focusing on the core rule-based idea and the graph. Accessible for introductory levels.	Lower rigor and precision; relies on 'rule' concept without fully formalizing it as a set of pairs. Less suitable for advanced notes.

Notes: Strength and weakness interpretations were created by chosen large language model against predefined math term definition quality metrics. Different metrics will change the evaluation interpretations with corresponding ranking, which can be reconfigured to meet different downstream task requirements.

Term-Specific Quality Metric Scores and Rankings
Study 3: Professor-Rank-Aware Prompt

Quality Metrics	D-P	F-P	E-P	R-P	Ex-P	Ao-P	T-P	Ai-P	Lect
Term-Specific Average Score	**9.50**	**9.36**	**9.09**	**8.91**	**8.64**	**8.55**	**8.50**	**8.27**	**8.09**
Term-Specific Ranking	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

Abbreviations: D-P = Distinguished-Professor-Aware Prompt, E-P = Emeritus-Professor-Aware Prompt, F-P = Full-Professor-Aware Prompt, R-P = Research-Professor-Aware Prompt, T-P = Teaching-Professor-Aware Prompt, Ao-P = Associate-Professor-Aware Prompt, Ai-P = Assistant-Professor-Aware Prompt, Lect = Lecturer-Aware Prompt, Ex-P = Excellent-Professor-Aware Prompt

A.3 Prompt Construction and Experiment Reproduction

When attempting to replicate or extend these baseline and proposed prompt methods, readers should be aware of several critical factors that influence the generation outcomes:

A.3.1 Considerations for Prompt Engineering

To achieve superior generation results, readers are strongly advised to:

- **Engineer Prompt Manually or with LLMs**: Construct and optimize the proposed and baseline prompt manually or with the help of large language models
- **Iterate on prompt wording**: Modify the exact phrasing and structure of prompts to better align with the target LLM's instruction-following patterns
- **Adjust temperature settings (When applied with Agent and API)**: Lower temperatures (e.g., 0.1 - 0.3) produce more deterministic and focused outputs, while higher temperatures (e.g., 0.7 - 1.0) generate more diverse and creative content
- **Experiment with prompt instruction ordering**: The sequence of instructions within a prompt can significantly impact output quality
- **Test across different models**: Generate definitions using multiple LLMs (e.g., Qwen, DeepSeek, ChatGPT, Gemini) and compare results.
- **Validate mathematical accuracy**: Always verify generated definitions against standard textbooks and lecture notes, as LLMs may occasionally produce seemingly fantastic but mathematically imprecise content

A.3.2 Evolving LLM Capabilities

Large language models are continuously being updated with changing capabilities. While most (mathematical) reasoning and instruction-following abilities are enhanced over time, some specific capabilities may be decreased or even eliminated in newer versions. Readers

should note that the outputs presented in this appendix were generated using specific model versions and may not be reproducible exactly with later versions.

A.3.3 Probabilistic Nature of LLM Generation

Due to the inherent probabilistic nature of LLM inference, each run produces different outputs. Even with identical prompts and settings, no exact generation can be guaranteed. The stochastic sampling process introduces variation in:

- The specific wording and phrasing of definitions
- The selection and presentation of mathematical examples
- The ordering and emphasis of explanatory details
- The completeness and depth of the generated content

If readers are not satisfied with a particular generation, they are encouraged to regenerate the output multiple times or modify the prompts to achieve better results.

A.3.4 LLM-Specific Behaviors

Different LLMs exhibit varying behaviors in:

- **Instruction understanding**: The ability to correctly parse and follow complex prompt structures
- **Mathematical reasoning**: Accuracy and rigor in generating mathematical content
- **Format adherence**: Compliance with specified formatting requirements (paragraph length, LaTeX usage, etc.)
- **Sequential thinking**: Effectiveness in following multi-step reasoning instructions
- **Role adherence**: Ability to adopt and maintain specified personas (e.g., "research professor," "excellent professor")

Readers are encouraged to adapt these prompt strategies to their specific research domains and content generation needs, keeping in mind the evolving nature of LLM capabilities and behaviors and the importance of human verification of generated content.