From the holdings of the Department of Special Collections , Princeton University Library, Princeton, NJ

These documents are either in the public domain, or can only be used for educational and research purposes (Fair use) as per U.S. Copyright law (text below). By accessing this file, all users agree that their use falls in the public domain or within fair use as defined by the copyrightlaw. We do not charge any permission or usage fees for the publication of images of material inour collections, including those provided by Princeton University Library, via our website, catalog, or directly from staff.For more information regarding copyright, credit, and citation please review the information onthe Copyright, Credit and Citation Guidelines page of our website.
https://library.princeton.edu/special-collections/policies/copyright-credit-and-citation-guidelines. If you have any questions, including how to order higher-quality images from the collection, contact Special Collections Public Services via our Ask Us! Form. https://library.princeton.edu/special-collections/ask-us

---

Master's Theses, 2022-2026 Electrical and Computer Engineering

2025

Optimizing Efficiency of Cognitive Agents with Improved Performance

Wang, Chenyu

Princeton, NJ : Princeton University

---

http://arks.princeton.edu/ark:/88435/dsp01z890rx674

U.S. Copyright law: The copyright law of the United States (Title 17, §108, United States Code) governs the making of photocopies or other reproductions of copyrighted material. Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be used for any purpose other than private study, scholarship, or research. If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of fair use that user may be liable for copyright infringement. This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve a violation of copyright law. https://www.copyright.gov/title17/title17.pdf

Optimizing Efficiency of Cognitive Agents with Improved Performance

Chenyu Wang

A Master's Thesis
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Master of Science in Engineering

Recommended for Acceptance
by the Department of
Electrical and Computer Engineering
Adviser: Tom Griffiths

May 2025

© Copyright by Chenyu Wang, 2025.

All rights reserved.

Abstract

This thesis addresses critical challenges in deploying cognitive agents, focusing on optimizing computational efficiency and performance in both multi-model collaboration and single-model inference scenarios. Cognitive agents based on large language models (LLMs) exhibit remarkable reasoning capabilities but face significant practical constraints, including high computational costs and substantial performance degradation under limited model size.

First, we explore optimization strategies for multi-model collaboration among cognitive agents. Existing collaborative LLM systems commonly suffer from inefficiencies due to fixed computational strategies that do not adapt to query complexity or model capabilities. We introduce complexity-aware collaboration, which dynamically allocates computational resources based on the estimated difficulty of each query, and capability-aware collaboration, selecting strategies that align with the intrinsic strengths of participating models. These adaptive approaches significantly reduce computational costs, achieving up to a $6.26\times$ reduction in token usage while maintaining or improving performance across diverse reasoning benchmarks.

Second, we investigate single-model optimization under long-context conditions. Through theoretical analysis and empirical evaluation, we identify that computational errors introduced by popular zero-shot compression techniques, particularly low-bit quantization, escalate as context length increases. In contrast, certain pruning strategies demonstrate robustness irrespective of context length. We propose a targeted compression strategy that selectively quantizes a small subset of sensitive weights at higher precision, effectively mitigating performance degradation without significantly increasing computational overhead.

By integrating insights from adaptive multi-model collaboration and targeted single-model compression, this work provides comprehensive and practical solutions for enhancing the cost-effectiveness and stability of cognitive agents. Our findings support efficient deployment of advanced reasoning capabilities in resource-constrained environments and lay a robust foundation for future research in scalable, adaptive agent cognitive systems.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Prof. Tom Griffiths, for his invaluable guidance, continuous support, and insightful suggestions throughout this research. My sincere thanks also go to my collaborators Prof. Yilun Du, Prof. Tushar Krishna, and Prof. Vijay Janapa Reddi, as well as my student collaborators Zishen Wan, Yihan Wang, and Hongkun Chen, for their dedication, collaborative efforts, and significant contributions.

I am also deeply thankful to the Department of Electrical and Computer Engineering at Princeton University for providing financial support and fostering an open and inspiring research environment that allowed me the freedom to pursue my academic interests.

I owe immense gratitude to my parents, whose unwavering support—both emotional and material—was essential in navigating the many challenges encountered over the past two years.

I also wish to thank everyone else who helped me along my academic journey. Though I cannot name each of you individually, your support has meant a great deal to me.

Contents

Abstract 3

Acknowledgements 4

List of Tables 7

List of Figures 8

Author's Statements 10

1 Introduction **11**

2 Related Work **15**

2.1 LLMs and Cognitive Agents 15

2.2 Multi-model Collaboration 17

2.2.1 Consistency-based Methods 18

2.2.2 Interaction-based Methods 19

2.3 Single-model Compression Techniques 21

3 Optimization Methods **24**

3.1 Multi-model Optimization for Cognitive Agents 25

3.1.1 Overview and Analysis of Existing Collaborative Cognitive Agents . . 25

3.1.2 Complexity-aware Collaboration 27

3.1.3 Capability-aware Collaboration 28

3.2 Single Model Optimization for Cognitive Agents 29

3.2.1 Theoretical Analysis of Existing Model Compression Methods under Long Context . . . 30
3.2.2 Empirical Evaluation of Theoretical Hypothesis . . . 31
3.2.3 Improving Quantization under Long Context with Hybrid Quantization 33

4 Experiments **37**
4.1 Experimental Setup . . . 38
4.2 Multi-Model Collaboration Results . . . 39
4.2.1 Experiments with Frontier LLMs . . . 39
4.2.2 Experiments with Computationally Efficient LLMs . . . 40
4.3 Single-Model Experiments . . . 42
4.3.1 Empirical Evaluation Setup . . . 42
4.3.2 Evaluation Setup for Hybrid Quantization . . . 43
4.4 Evaluation of Combined Single- and Multi-Model Optimization . . . 44

5 Analysis of Interaction-based Methods on Small LLMs **46**
5.1 Diagnostic Analysis of Inter-model Dialogues . . . 46
5.2 Diagnosed Causes of Interaction-based Method Failures . . . 47
5.2.1 Failure to Adhere to Prompt Instructions . . . 47
5.2.2 Superficial Engagement in Debate . . . 48
5.2.3 Ineffective or Inconsistent Debate Strategies . . . 48
5.2.4 Error Propagation and Amplification . . . 49

6 Conclusion **50**

Bibliography **52**

References **52**

List of Tables

4.1 Performance of frontier models on the MATH and GPQA datasets. 39
4.2 Multi-model collaboration results on the MATH dataset using frontier LLMs. 40
4.3 Multi-model collaboration results on the GPQA dataset using frontier LLMs. 40
4.4 Accuracy of individual computationally efficient models on the MATH and GPQA datasets. 41
4.5 Performance of multi-model collaboration methods using computationally efficient LLMs. Accuracy is measured on the MATH and GPQA datasets, with Upper Bound reflecting the oracle choice of the best model per question. . . 42
4.6 End-to-end compute reduction relative to the FP16 baseline obtained by combining hybrid 4-bit quantization with complexity-aware collaboration. The overall reduction is the product of the token ratio and the per-token FLOP ratio. 45

List of Figures

3.1 Performance of existing multi-model cognitive agents: (a) accuracy on frontier models, (b) token usage on frontier models, and (c) accuracy on computationally efficient models. The dashed lines indicate oracle upper bounds, illustrating both the computational inefficiency (high token consumption) and the ineffectiveness (limited accuracy gains on small models) of current approaches. 26

3.2 KL divergence under long-context for W3 and W3A8 quantization. Weight-activation quantization shows more divergence as context grows. 32

3.3 KL divergence under different bit-widths for weight-activation quantization. Lower precision (e.g., W4A8) leads to higher divergence, especially beyond 24K tokens. 33

3.4 KL divergence under different bit-widths for weight-only quantization. W4 again exhibits sharp error growth under long context. 34

3.5 KL divergence for Wanda pruning across sparsity levels. Error remains flat across varying context lengths, indicating robustness. 35

3.6 KL divergence for magnitude pruning. Similar to Wanda, context length has negligible effect on divergence. 35

3.7 Hybrid weight-activation quantization (e.g., W3A8 with 2% W8): preserving a small subset of large weights substantially stabilizes KL divergence across longer contexts. 35

3.8 Hybrid weight-only quantization (e.g., W4 with 2% W8): similar stabilizing effect is observed, especially compared to naive low-bit quantization. 36

Author's Statements

Portions of this thesis are adapted and substantially extended from two prior works by the author: "Evaluating Zero-Shot Long-Context LLM Compression" (arXiv:2406.06773, final report for COS 598D at Princeton) and "Optimizing Cost and Performance for Compositional Language Models" (manuscript submitted to COLM 2025). All derived sections have been significantly updated and are presented here with the approval of my advisor. Any remaining errors are solely my responsibility.

Throughout the writing process, AI tools were used only for brainstorming section outlines and improving English fluency. All ideas, analyses, and final text remain the sole responsibility of the author, and usage was approved by the thesis adviser.

Chapter 1

Introduction

Recent advancements in large language models (LLMs) have significantly enhanced artificial intelligence, enabling a wide range of new applications across various domains [1–10]. Among these emerging applications, building cognitive agents based on LLMs has gained particular attention [11]. A cognitive agent is defined as an intelligent entity capable of reasoning, learning from experience, explaining itself, being aware of its capabilities, and responding effectively to unexpected situations [12].Recently, researchers have explored the possibility of realizing cognitive agents by composing multiple LLM invocations, where each LLM handles specific reasoning subtasks or verification roles, collectively forming a more capable cognitive system. The powerful reasoning, contextual understanding, and adaptive learning capabilities demonstrated by recent LLMs have made them particularly suitable as foundational components for such cognitive agents, enabling these agents to tackle complex tasks that were previously beyond reach [13].

Despite these critical efficiency and performance challenges, most current research on cognitive agents emphasizes enhancing accuracy and capabilities without adequately addressing optimization of computational efficiency [14]. Consequently, although cognitive agents achieve notable performance improvements, their substantial computational costs continue to hinder large-scale deployment. Bridging this gap—balancing computational efficiency

with high performance—is crucial for the broader adoption and practical use of LLM-based cognitive agents.

In this thesis, we build on several key observations regarding efficiency optimization in cognitive agents. First, at the multi-model collaboration level, we observe that existing cognitive agents typically apply static processing strategies uniformly across all queries [13, 15, 16], regardless of query difficulty or individual model capability. Our experiments demonstrate that this static design frequently leads to computational redundancy—simpler queries consume excessive resources—and capability mismatch, where less capable models are inappropriately tasked with challenging inference scenarios, resulting in suboptimal performance. Second, we find that cognitive agents often share large amounts of contextual information across multiple model invocations, making efficient handling of extended input contexts particularly important. As the context length grows, the computational cost of each individual LLM inference increases super-linearly [17, 18], making the optimization of single-model inference costs under long-context conditions crucial for building efficient cognitive agents. In principle, model compression techniques—particularly quantization—can mitigate these computational costs [19–21]. However, our analysis reveals critical issues: conventional zero-shot compression methods tend to accumulate significant computational errors as context length grows, substantially degrading overall model performance [22].

Building upon these observations, this thesis systematically explores efficiency optimization strategies for cognitive agents from two distinct but interconnected perspectives:

Multi-model Optimization: To address these observed inefficiencies in multi-model cognitive agents, we introduce two novel adaptive strategies. First, we propose a complexity-aware collaboration strategy that dynamically allocates computational resources based on the predicted difficulty of individual queries, significantly reducing redundancy by ensuring simpler queries are processed efficiently without unnecessary computational overhead. Second, through extensive experimentation, we discover that aligning collaboration meth-

ods closely with individual model capabilities notably enhances overall system effectiveness. Motivated by this insight, we propose the capability-aware collaboration strategy, which carefully selects processing strategies that match the intrinsic strengths of each model. By ensuring that each model tackles subtasks best suited to its capabilities, we achieve superior overall performance given the same computational budget. Our proposed adaptive methods significantly enhance efficiency and effectiveness in cognitive agents across diverse reasoning tasks.

Single-model Optimization: At the single-model inference level, we primarily focus on exploring the effectiveness of model compression techniques in handling long-context scenarios—a situation frequently encountered by cognitive agents due to extensive contextual information shared among model invocations. Through detailed theoretical analysis and empirical experiments, we identify significant limitations of conventional zero-shot compression methods in long-context settings. Specifically, our findings indicate that low-bit quantization techniques often exacerbate computational errors as context length increases, substantially degrading inference accuracy. To address this critical issue, we propose targeted compression strategies that selectively apply higher-precision quantization to a small subset of sensitive model parameters. Our approach effectively mitigates error accumulation while maintaining computational efficiency, thereby ensuring robust single-model inference performance even when processing extensive contexts characteristic of practical cognitive-agent applications.

By addressing these two optimization dimensions, our research makes three primary contributions:

- We provide a comprehensive two-level optimization framework for cognitive agents, covering both multi-model and single-model inference scenarios. Our unified approach systematically addresses efficiency bottlenecks prevalent in existing cognitive agent architectures.

- At the multi-model level, we introduce adaptive collaboration strategies—including complexity-aware and capability-aware collaboration—that dynamically adjust computational resource allocation and inference strategies based on query complexity and model capabilities. These strategies significantly reduce computational overhead while enhancing inference performance across diverse reasoning tasks.

- At the single-model level, we conduct detailed theoretical and empirical analyses of model compression techniques in long-context scenarios, identifying key factors leading to computational error accumulation. We then propose targeted quantization strategies that selectively employ higher precision for critical parameters, effectively mitigating performance degradation without compromising computational efficiency.

The remainder of this thesis is organized as follows. Chapter 2 reviews relevant literature on cognitive agents, focusing on LLM-based multi-model collaboration and model compression under long-context settings. Chapter 3 introduces our proposed optimization framework, including both complexity-aware and capability-aware collaboration methods, as well as single-model compression strategies. Chapter 4 presents detailed experimental results demonstrating the performance and efficiency benefits of our methods. Chapter 5 provides a diagnostic analysis of interaction-based collaboration failures in small LLMs. Chapter 6 concludes with a summary of contributions and implications for future research.

Chapter 2

Related Work

In this chapter, we provide a review of recent advancements in LLMs and cognitive agents (Section 2.1), highlighting the development of LLM capabilities and their use as building blocks for cognitive agents. We then discuss current methods and limitations in multi-model collaboration for cognitive agents (Section 2.2), examining various existing approaches. Finally, we review the state-of-the-art in single-model compression techniques (Section 2.3).

2.1 LLMs and Cognitive Agents

Large language models (LLMs) refer to a family of machine learning models trained on vast amounts of text data to understand and generate human language [23, 24]. These models typically contain billions of parameters, enabling them to perform a wide range of natural language tasks effectively. The transformative impact of LLMs on artificial intelligence has been primarily driven by advances such as the Transformer architecture [25], which allows efficient processing of long textual inputs and significantly improves model performance on complex language tasks.

The practical usefulness of LLMs has developed significantly in recent years. Early language models, such as recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks, faced difficulties in capturing long-range dependencies within language,

limiting their effectiveness on complex linguistic tasks [26–28]. The introduction of the Transformer architecture in 2017 provided a fundamental shift, enabling models to more effectively handle linguistic complexity through self-attention mechanisms. Building upon this, OpenAI's GPT (Generative Pre-trained Transformer) series demonstrated that systematically increasing both the number of parameters and the size of training datasets could substantially enhance model capabilities [29]. Subsequent models—including GPT-3 and GPT-4, as well as related models such as Meta's LLaMA, Google's Gemini, and Alibaba's Qwen—further confirmed this relationship, showcasing strong generalization performance across various tasks even when provided with minimal or no task-specific examples [1–9]. These developments collectively laid the foundation for the extensive adoption of LLMs as crucial components in sophisticated artificial intelligence systems.

A cognitive agent is an intelligent system designed to perform complex tasks through sophisticated reasoning, adaptive learning, interpretability, and self-awareness of its capabilities, effectively responding to dynamic and uncertain environments [30, 31]. Unlike simple reactive systems, cognitive agents integrate reasoning and action in a deliberate manner, enabling them to plan, adjust strategies, and reflect upon past actions and outcomes to improve future performance [11].

Recent research has increasingly leveraged LLMs as foundational components for cognitive agents [11, 32–34]. This integration is primarily driven by LLMs' advanced capabilities, including their contextual understanding, powerful reasoning, and adaptability to varied scenarios. For instance, the ReAct framework highlights how LLMs can synergistically interleave reasoning and actions, improving performance on complex tasks by dynamically adjusting plans and effectively responding to real-time feedback [35]. In practical applications such as question-answering systems and interactive decision-making benchmarks (e.g., HotpotQA [36] and ALFWorld [37]), cognitive agents built upon LLMs have demonstrated superior interpretability, reduced error rates, and enhanced trustworthiness compared to traditional approaches.

Moreover, the Reflexion framework further enhances cognitive agents by incorporating verbal reinforcement learning, allowing these agents to self-reflect on previous actions and outcomes [38]. This self-reflection mechanism stores experiences in episodic memory, thereby enabling the agent to iteratively refine its decision-making processes. For example, in coding benchmarks like HumanEval [39], Reflexion-based cognitive agents achieved significant improvements over state-of-the-art approaches, demonstrating an ability to learn efficiently from trial-and-error without extensive retraining.

Another notable advancement is the Tree of Thoughts (ToT) framework, which generalizes the popular Chain-of-Thought prompting method [40]. ToT enhances LLMs' capabilities by allowing them to explore multiple reasoning pathways simultaneously, using self-evaluation mechanisms to determine the most promising paths. This structured approach significantly boosts performance in tasks requiring complex planning and strategic reasoning, such as puzzle-solving and creative writing tasks.

Furthermore, recent innovations like MetaGPT [41], Mixture-of-Agents [13], LLM-Debate [15] showcase the potential of multi-agent collaborations, where structured workflows and clear role definitions dramatically improve cognitive agents' coherence and problem-solving efficiency.

Overall, these advancements illustrate the significant potential of integrating LLMs into cognitive agent architectures. By harnessing the combined power of deliberate reasoning, self-reflection, strategic exploration, and collaborative workflows, LLM-based cognitive agents offer substantial promise in tackling diverse and complex real-world challenges.

2.2 Multi-model Collaboration

Multi-model collaboration represents a significant approach in realizing cognitive agents by harnessing multiple invocations of LLMs to improve reasoning capabilities [40, 42, 43]. This approach is vital because it exploits the diverse reasoning pathways and complementary

strengths of various models, enhancing the overall accuracy, robustness, and interpretability of cognitive agents.

Currently, two major classes of multi-model collaboration methods have emerged: consistency-based methods and interaction-based methods. Consistency-based methods involve multiple invocations of the same or different LLMs to explore diverse reasoning paths and then aggregate the results to find the most reliable and consistent answer [44, 45]. Interaction-based methods, on the other hand, emulate more human-like reasoning by facilitating communication and debate between different LLM instances to iteratively refine and enhance the quality of their outputs [15, 46–48].

2.2.1 Consistency-based Methods

A prominent example within consistency-based methods is the Self-Consistency approach [44]. This method generates multiple reasoning paths through repeated LLM invocations and selects the final answer based on majority voting or other aggregation strategies. For instance, the original Self-Consistency method introduced significantly enhances reasoning accuracy by mitigating the inherent inaccuracies of single-model outputs through consensus-based selection.

Further optimizing this strategy, adaptive consistency methods like Adaptive-Consistency [49] and Difficulty-Adaptive Self-Consistency (DSC) [50] have been developed. Adaptive-Consistency dynamically adjusts the number of samples based on problem difficulty and observed consistency, stopping additional sampling once confidence is established. DSC further refines this approach by pre-assessing problem difficulty to allocate computational resources efficiently, significantly reducing redundant computation and maintaining high accuracy across arithmetic, commonsense, and symbolic reasoning benchmarks.

Soft Self-Consistency (Soft-SC) [51] further extends consistency-based approaches by addressing the limitations of binary majority voting in contexts with multiple valid answer representations. Instead of exact matching, Soft-SC employs continuous scoring derived

from model likelihoods, enabling nuanced selection among candidate actions and demonstrating significant performance improvements in interactive scenarios such as coding and online shopping.

2.2.2 Interaction-based Methods

Interaction-based methods focus on improving reasoning through structured, iterative communication among multiple LLM agents, closely emulating human collaborative processes such as debate, discussion, and verification [15].

One notable method in this category is the multi-agent debate framework [15]. In this approach, multiple LLM instances independently propose initial solutions to a given reasoning task. These instances subsequently engage in structured iterative debates, during which agents critique and refine their reasoning based on peer interactions. Through these debate rounds, the overall quality of reasoning significantly improves, particularly in terms of factual accuracy, logical consistency, and robustness against individual model hallucinations. The multi-agent debate framework has demonstrated substantial effectiveness across a wide range of tasks, from complex commonsense questions to specialized logical reasoning problems, without necessitating internal model modifications or extensive retraining.

Another critical advancement is the Mixture-of-Agents (MoA) framework [13]. MoA employs a layered collaborative structure wherein agents in earlier layers generate initial outputs, subsequently serving as input for deeper layers. Each subsequent layer synthesizes and refines information from previous layers, enabling a systematic iterative enhancement of outputs. Crucially, MoA explicitly differentiates the roles of agents, distinguishing between those primarily generating diverse proposals (Proposers) and those adept at synthesizing and aggregating information (Aggregators). Extensive experimental results have shown MoA achieving state-of-the-art performance on multiple challenging benchmarks, including AlpacaEva [52] and MT-Bench [53], clearly outperforming both single-model baselines and

conventional ensemble methods. These results underscore the substantial value of explicit role differentiation and structured refinement inherent in the MoA design.

Similarly important is the Multi-Agent Verification (MAV) framework [16], which introduces specialized agents known as Aspect Verifiers (AVs). Each AV independently verifies specific aspects of candidate outputs generated by other models, producing binary verification results. MAV leverages these simplified verification outcomes, aggregating multiple AVs' judgments through straightforward voting mechanisms. This approach effectively enhances the accuracy and robustness of the reasoning process, demonstrating superior scalability as both the number of candidate outputs and the number of verifiers increase. MAV has exhibited clear advantages over traditional single-model verification approaches across complex reasoning benchmarks, notably on mathematical and general reasoning datasets.

Other recent efforts have further expanded the exploration of multi-agent collaboration. RECONCILE framework organizes diverse LLM agents into a structured round-table discussion [54] . Each agent provides an initial answer along with a confidence estimate, followed by iterative discussions where agents present human-like explanations and dynamically update their confidence. Experimental results show improved reasoning performance across multiple benchmarks.

Self-Supervised Prompt Optimization (SPO) framework optimizes prompts by comparing model outputs in a pairwise manner [55], guided by evaluations from LLMs themselves. This approach enables high-quality reasoning without the need for external supervision and reduces computational costs.

Archon framework integrates multiple inference-time techniques such as ensembling, critiquing, verification, and ranking [56]. Using Bayesian optimization, Archon searches for effective configurations of collaborative strategies and achieves consistent performance gains across diverse tasks.

Weak-for-Strong Harnessing (W4S) framework trains a smaller meta-agent to design workflows for stronger LLMs [57]. W4S formulates workflow optimization as a multi-turn

Markov decision process and employs reinforcement learning to iteratively improve performance. Without fine-tuning the strong models, W4S effectively leverages their capabilities and demonstrates robust generalization across seen and unseen tasks.

These developments collectively reflect the growing interest in improving the robustness, efficiency, and adaptability of multi-model collaboration strategies.

2.3 Single-model Compression Techniques

The efficiency of single-model inference directly determines the overall efficiency of cognitive agents. In practical cognitive-agent scenarios, a large volume of contextual information is repeatedly processed, significantly increasing computational demands, especially as context length scales. Thus, optimizing single-model efficiency becomes crucial for managing computational costs and ensuring system scalability.

To address computational constraints, researchers commonly leverage model compression techniques, which aim to reduce the size and computational overhead of neural networks without substantially compromising their performance [58, 59]. In general, model compression methods seek to simplify a model's parameters or computations, enabling faster inference and reduced memory usage, thus making advanced reasoning capabilities accessible even in resource-constrained environments [60].

Within the context of LLMs, two predominant compression methods are quantization and pruning [61, 62], each targeting distinct aspects of model optimization.

Quantization refers to the reduction of numerical precision of the model's parameters and activations [58]. Instead of representing parameters in high precision formats (e.g., 32-bit floating-point), quantization employs lower precision formats such as INT8 (W8A8 quantization) or even INT4 (W4A16 quantization). By lowering the precision, quantization reduces memory requirements and computational demands significantly, enabling efficient hardware utilization.

Several LLM quantization methods are introduced here:

GPTQ [63] uses approximate second-order Hessian information to guide post-training quantization. By selectively adjusting weights, it achieves high compression ratios (down to 3–4 bits per weight) while maintaining near-original accuracy, even for models with hundreds of billions of parameters. Leveraging second-order information helps mitigate the performance loss typically caused by quantization.

SqueezeLLM [64] applies a sensitivity-based non-uniform quantization strategy combined with Dense-and-Sparse decomposition. By isolating sensitive and extreme weights, it allows the majority to be quantized more aggressively. This enables 3-bit quantization with minimal accuracy loss and improved inference speed.

SmoothQuant [61] addresses activation outliers by shifting quantization complexity from activations to weights through channel-wise scaling. This balanced approach enables effective INT8 quantization for both weights and activations, maintaining accuracy while significantly boosting speed and memory efficiency.

AWQ (Activation-aware Weight Quantization) [65] identifies weights that are critical to accuracy based on their interaction with important activations. It introduces per-channel scaling to protect these weights, minimizing quantization errors and supporting aggressive quantization (e.g., INT4, INT3) with minimal performance degradation, making it well-suited for resource-constrained deployment.

Pruning, on the other hand, involves systematically removing redundant or less important parameters from a neural network, thereby creating sparse models that require fewer computational resources for inference [58]. Pruning techniques can be categorized into structured and unstructured pruning [66, 67]. Structured pruning methods eliminate entire parameter blocks or neurons, which aligns well with hardware implementations, whereas unstructured methods remove individual weights based on their magnitude or importance scores. For instance, magnitude pruning retains weights of higher absolute value under the assumption they contribute more significantly to model performance. More advanced unstructured

pruning methods like Wanda leverage sophisticated heuristics to identify critical weights effectively, enabling high sparsity without considerable loss in model capability.

Several LLM pruning works are also introduced here for reference:

SparseGPT [68] introduces a one-shot pruning mechanism that effectively induces sparsity levels of up to 60% in massive GPT-family models, such as OPT-175B [69] and BLOOM-176B [70], without any retraining or fine-tuning. It achieves this by transforming the pruning challenge into large-scale sparse regression problems, subsequently solving these with efficient approximation techniques. Notably, SparseGPT maintains performance even at extremely high sparsity levels, significantly surpassing traditional magnitude pruning methods which collapse beyond modest sparsity thresholds [68]. .

Wanda pruning [71], on the other hand, proposes a simpler yet highly effective pruning criterion tailored to exploit the distinctive property of emergent large magnitude activations observed in large language models. Unlike methods that purely rely on weight magnitudes, Wanda pruning evaluates the importance of each weight by multiplying its absolute magnitude with the norm of the corresponding input activation. By applying this metric on a per-output basis, Wanda pruning effectively identifies sparse subnetworks that preserve critical functional elements without requiring weight updates or retraining. This approach demonstrates competitive performance with SparseGPT, notably outperforming traditional magnitude pruning across diverse benchmarks and model scales [71]. .

Chapter 3

Optimization Methods

In this chapter, we introduce the methodological framework developed to simultaneously optimize both the efficiency and performance of cognitive agents, which heavily rely on LLMs. Given that cognitive agents commonly leverage multiple interacting models to tackle complex reasoning tasks, we begin by focusing specifically on the multi-model optimization paradigm. Multi-model optimization represents a unique and essential problem for cognitive-agent systems, where multiple LLMs collaborate through structured interactions such as debates, verification, or hierarchical consistency checks. In response to the computational inefficiencies inherent in fixed collaboration strategies, we propose methods that dynamically adjust computational allocation based on task complexity and model-specific capabilities, thereby significantly reducing resource usage without compromising reasoning accuracy.

Building upon the multi-model optimization approach, we further incorporate an additional layer of optimization focused on individual models—single-model optimization—to achieve even greater system-wide improvements. By introducing single-model compression techniques, particularly through quantization and pruning methods, we effectively reduce the computational demands associated with long-context inference scenarios. These single-agent enhancements not only further optimize inference efficiency but also help mitigate cumulative computational errors under extended contexts, ultimately complementing and

amplifying the benefits provided by multi-model strategies. Our proposed integrated optimization approaches ensure that cognitive agents deliver robust performance and efficiency across a diverse array of challenging reasoning tasks.

3.1 Multi-model Optimization for Cognitive Agents

3.1.1 Overview and Analysis of Existing Collaborative Cognitive Agents

Recent developments in cognitive agent architectures have increasingly incorporated multiple LLMs to amplify reasoning and decision-making capabilities. This multi-model collaboration, leveraging the collective intelligence of multiple LLM invocations, has enabled significant advancements in complex problem-solving tasks. Nonetheless, the practical deployment of these multi-model cognitive agents faces two critical, yet inadequately addressed challenges: computational inefficiency and ineffective utilization of smaller, more computationally efficient (thus with weaker single-model performance) models.

Computational inefficiency prominently emerges from two distinct dimensions. First, current multi-model cognitive agents often employ the best available frontier models—such as GPT-4 [2] or Claude 3.5 [72]—in empirical evaluations without considering their high computational costs. These state-of-the-art models individually incur significant computational expenses, and invoking them repeatedly exacerbates this inefficiency. Our empirical evaluations specifically reveal that many queries are already correctly solved during initial inference stages, making subsequent iterative reasoning steps redundant. Thus, excessive repeated calls substantially inflate computational costs without corresponding accuracy improvements, severely limiting scalability and practicality.

In parallel, the ineffective utilization of smaller, computationally efficient LLMs poses another significant challenge. When optimizing for reduced computational costs by substituting frontier models with smaller, computationally efficient models, traditional interactive

strategies, including iterative debates and verification mechanisms, become ineffective. Our experiments clearly demonstrate that these computationally efficient models have inherently limited instruction-following and meta-cognition capabilities. Consequently, complex iterative frameworks frequently yield negligible or even negative performance improvements compared to simpler aggregation methods such as consistency-based frameworks or majority voting.

Figure 3.1 clearly illustrates these two key issues of inefficiency and ineffectiveness in current multi-model collaboration methods. Panel (a) shows that complex iterative methods such as debate and verification offer only marginal accuracy improvements over simpler strategies like consistency checks or even single-model inference. Meanwhile, panel (b) vividly depicts the excessive computational resources consumed by verification-based methods, despite their minimal accuracy advantage. Panel (c) further highlights the ineffectiveness of complex iterative strategies when applied to smaller, computationally efficient models, clearly demonstrating negligible performance improvements compared to consistency, which is a very simple aggregation method.

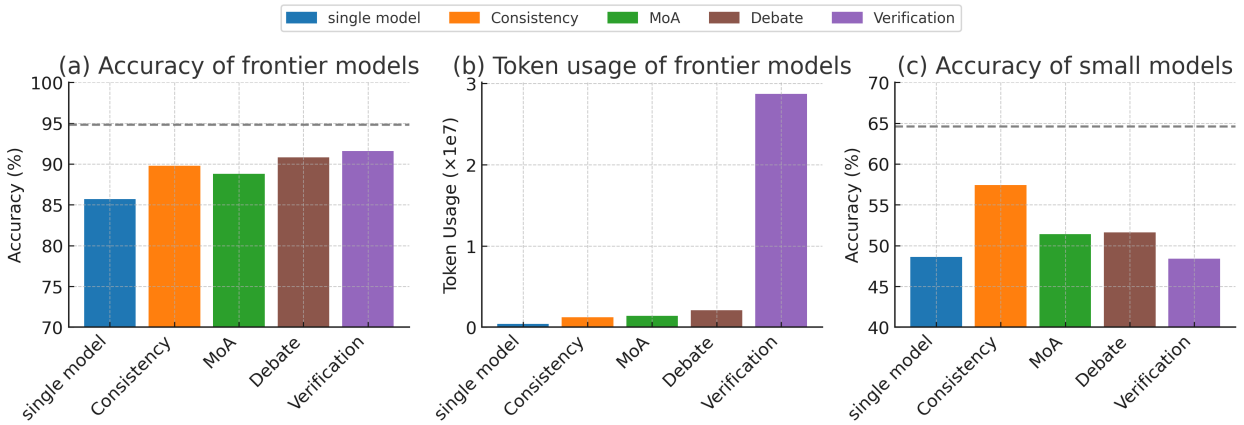


Figure 3.1: Performance of existing multi-model cognitive agents: (a) accuracy on frontier models, (b) token usage on frontier models, and (c) accuracy on computationally efficient models. The dashed lines indicate oracle upper bounds, illustrating both the computational inefficiency (high token consumption) and the ineffectiveness (limited accuracy gains on small models) of current approaches.

To systematically address these dual challenges of inefficiency and ineffectiveness, this work introduces a dynamic cognitive agent architecture designed explicitly around adaptive strategies informed by query complexity and model capabilities. Specifically, this architecture integrates two key optimization dimensions:

Firstly, complexity-aware collaboration dynamically allocates computational resources according to the estimated complexity of each query. By selectively employing simplified processing strategies for simpler queries and reserving comprehensive iterative reasoning for more challenging tasks, complexity-aware collaboration significantly minimizes redundant computational effort.

Secondly, capability-aware collaboration explicitly matches collaborative strategies to the inherent strengths and limitations of participating models. Recognizing the restricted reasoning abilities of smaller, computationally efficient models, capability-aware collaboration prioritizes simpler yet robust aggregation mechanisms, such as hierarchical consistency checks, enhancing accuracy and computational efficiency for resource-constrained scenarios.

In summary, the proposed dynamic cognitive agent architecture systematically addresses computational inefficiencies from redundant frontier model invocations and the ineffective use of computationally efficient models. By integrating complexity-aware and capability-aware collaborations, this architecture offers a well-motivated, scalable, and cost-effective solution for practical multi-model cognitive agent deployment.

3.1.2 Complexity-aware Collaboration

A key insight from our analysis is that not all queries require the same depth of collaborative reasoning. Simple queries can typically be resolved by initial consensus among models, while more intricate queries necessitate deeper, iterative interactions. However, most existing cognitive agents adopt uniform collaboration strategies, leading to unnecessary computational redundancy for straightforward tasks and inadequate processing for complex scenarios.

We propose complexity-aware collaboration, which dynamically adjusts computational resources based on query complexity. Specifically, we use initial response consistency as a practical heuristic for complexity estimation. Queries generating consistent initial responses across multiple models are deemed simpler, thus concluding collaboration immediately. Conversely, queries eliciting divergent initial responses trigger additional iterative reasoning steps.

Algorithm 1 summarizes this complexity-aware collaborative process clearly:

Algorithm 1 Complexity-aware Collaboration

Input: Models $M_1, \ldots, M_n$, query x, iteration threshold T
Output: Final answer $\hat{y}$

for $i = 1, \ldots, n$ **do**
    $y_i \leftarrow M_i(x)$ ▷ initial independent answers
end for
if answers y_i are consistent (majority agreement exists) **then**
    return majority answer ▷ terminate early for simplicity
end if
for $t = 1, \ldots, T$ **do**
    $x_{refined} \leftarrow$ refine prompt using previous answers y_i
    for $i = 1, \ldots, n$ **do**
        $y_i \leftarrow M_i(x_{refined})$ ▷ iterative reasoning step
    end for
    if answers y_i converge or stabilize **then**
        break ▷ stop iterations when stabilized
    end if
end for
$\hat{y} \leftarrow$ aggregate final answers y_i (majority or weighted voting)
return $\hat{y}$

3.1.3 Capability-aware Collaboration

Beyond complexity, effective collaboration must also align strategies with the inherent strengths of individual cognitive agent models. Iterative interactive methods—such as verification or debate—show substantial benefits when applied to powerful frontier models but often fail to improve outcomes when applied to smaller, computationally efficient models.

This disparity arises primarily due to limited reasoning and instruction-following capabilities within computationally efficient models.

Motivated by this observation, we introduce capability-aware collaboration. This strategy explicitly matches collaboration techniques to the capabilities of participating models. For computationally efficient models, we propose hierarchical consistency—a lightweight and robust aggregation method. This method first queries each model independently, then further assesses response reliability through additional sampling if initial consensus is not reached.

Algorithm 2 illustrates this capability-aware strategy in detail:

Algorithm 2 Capability-aware Collaboration (Hierarchical Consistency)

Input: Models $M_1, \ldots, M_n$, query x, additional samples K
Output: Final answer $\hat{y}$

for $i = 1, \ldots, n$ **do**
    $y_i \leftarrow M_i(x)$ ▷ initial independent answers
end for
if all y_i identical **then**
    return y_1 ▷ unanimous consensus reached
end if
for $i = 1, \ldots, n$ **do**
    Generate K additional samples from $M_i(x)$
    $A_i \leftarrow$ most frequent answer among samples
    $C_i \leftarrow$ (frequency of A_i)$/K$ ▷ self-consistency scores
end for
$\hat{y} \leftarrow$ answer A_i associated with highest score C_i
return $\hat{y}$

3.2 Single Model Optimization for Cognitive Agents

Cognitive agents commonly involve frequent exchanges and sharing of extensive contextual information among multiple interacting models. As the context length expands, the computational complexity associated with individual model inference increases significantly thus posing substantial efficiency challenges. Consequently, optimizing single model performance under extended context lengths becomes crucial for maintaining overall system efficiency. In this section, we explore targeted strategies to enhance single model inference

efficiency, specifically addressing challenges unique to long-context scenarios. By identifying and mitigating computational errors exacerbated by conventional model compression techniques under extended contexts, our approach ensures robust inference performance, effectively complementing multi-model collaborative strategies.

3.2.1 Theoretical Analysis of Existing Model Compression Methods under Long Context

LLMs are predominantly constructed using transformer architectures, which fundamentally operate as sequence models. Each token generation involves computing attention scores against hidden states of all preceding tokens, inevitably leading to computational overhead. While compression techniques effectively reduce inference times, they simultaneously introduce computational inaccuracies affecting both hidden states and model outputs. These inaccuracies compound as more tokens are processed, escalating computational errors especially in long-context scenarios.

To illustrate this theoretically, consider the transformer architecture handling a sequence, where each token t has an associated query vector $\mathbf{q}_t$ and corresponding key vectors $\mathbf{k}_1, \ldots, \mathbf{k}_t$ from previous tokens. The attention scores $\mathbf{a}_t$ are computed as follows:

$$\mathbf{a}_t = \text{softmax}\left(\frac{\mathbf{q}t[k_1^T \cdots kt^T]}{\sqrt{d_k}}\right) \tag{3.1}$$

with d_k denoting the dimensionality of key vectors for appropriate scaling.

The hidden state $\mathbf{h}_t$ for token t is subsequently derived by:

$$\mathbf{h}_t = [v_1^T, \cdots, v_t^T]\mathbf{a}_t \tag{3.2}$$

where $\mathbf{v}_t$ represents corresponding value vectors.

Compression methods introduce noise ($\epsilon \sim \mathcal{N}(0, \sigma^2)$) into these key and value vectors, altering the calculation to:

$$\mathbf{h}t \approx \sum i = 1^t \text{softmax}\left(\frac{\mathbf{q}ik_i}{\sqrt{d_k}}\right) v_i + \sum i = 1^t \left(\text{softmax}\left(\frac{\mathbf{q}ik_i}{\sqrt{d_k}}\right) \epsilon vi + \frac{\partial a_{ti}}{\partial k_i} v_i \epsilon_{ki}\right) \quad (3.3)$$

This formulation highlights that compressed models accumulate noise from each token, with the variance of this cumulative noise increasing linearly as the context length expands, thereby significantly magnifying computational errors in long contexts.

3.2.2 Empirical Evaluation of Theoretical Hypothesis

To validate the theoretical insight that compression-induced noise may accumulate in long-context inference, we conduct a series of experiments on the LLaMA-2-7B-32K [73] model. Specifically, we examine how different compression strategies—pruning and quantization—affect model output distribution as the input context length increases.

We consider two representative pruning techniques: magnitude pruning, which removes weights based on their absolute value [58], and Wanda pruning [71], which additionally considers activation norms to better preserve critical weights. For quantization, we test both weight-only and weight-activation schemes across different bit-widths, with a focus on low-bit ($\leq$ 4-bit) settings.

To quantify deviation caused by compression, we compute the KL divergence between the output distributions of compressed and uncompressed models across a range of context lengths. This metric captures how much the compressed model's predictions diverge from the original baseline.

The experimental results are summarized in Figures 3.2, 3.3, 3.4, 3.5, and 3.6.

Figure 3.2 illustrates that when using 3-bit quantization (W3), the KL divergence grows rapidly with increasing context length, especially for the joint weight-activation quantization

(W3A8). This clearly supports our hypothesis that quantization error accumulates over long sequences.

In Figures 3.3 and 3.4, we observe that this effect is more pronounced as the bit-width decreases. Specifically, W4 quantization shows a sharp increase in KL divergence around 24K tokens, while W8 quantization maintains low and stable error. These results demonstrate that low-bit quantization is particularly vulnerable under long-context settings.

In contrast, Figures 3.5 and 3.6 reveal that both Wanda and magnitude pruning are relatively robust to context length. Across a range of sparsity levels (10% to 50%), the KL divergence remains nearly flat as the context length increases. This suggests that pruning—even at high sparsity—does not introduce cumulative errors in the same way quantization does.

In summary, the empirical results strongly support our theoretical claim: pruning maintains stable output distributions under long input sequences, while low-bit quantization, if not carefully applied, leads to significant degradation due to the accumulation of approximation error.



Figure 3.2: KL divergence under long-context for W3 and W3A8 quantization. Weight-activation quantization shows more divergence as context grows.

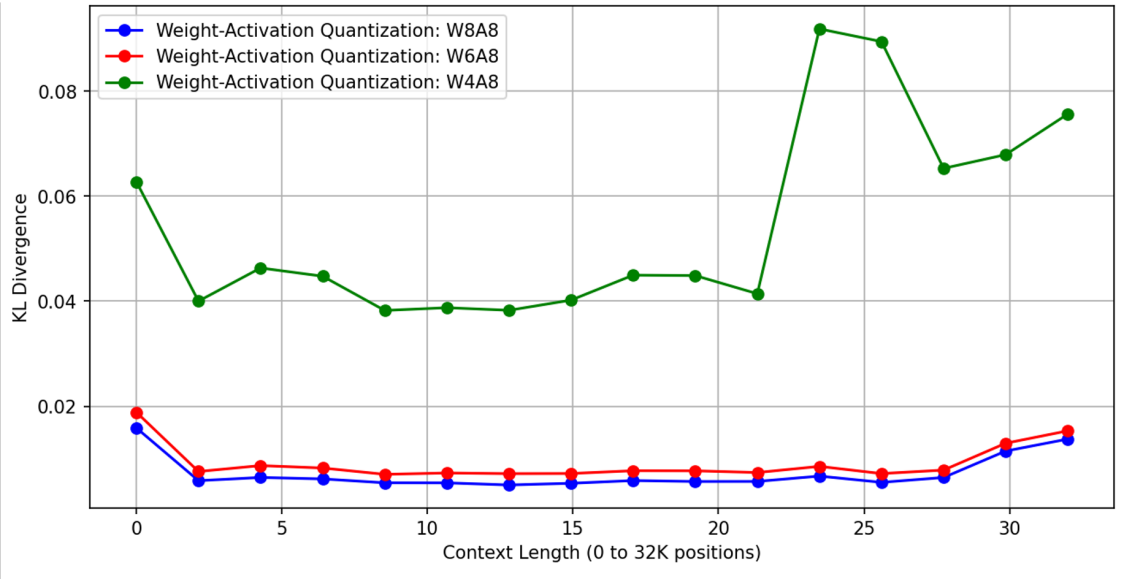


Figure 3.3: KL divergence under different bit-widths for weight-activation quantization. Lower precision (e.g., W4A8) leads to higher divergence, especially beyond 24K tokens.

3.2.3 Improving Quantization under Long Context with Hybrid Quantization

Our previous analysis identified a critical weakness of low-bit quantization under long-context settings: unlike pruning methods, which selectively remove low-magnitude weights, conventional quantization treats all weights equally, including those that may be highly sensitive to distortion. This design choice can lead to a substantial accumulation of computational errors as context length increases.

Motivated by this observation, we hypothesize that large-magnitude weights play a disproportionately important role in maintaining the model's stability under long-context inference. Inspired by the selective nature of pruning, we propose a simple yet effective improvement: instead of quantizing all weights uniformly, we preserve a small fraction (e.g., 2%) of the largest weights at higher precision (e.g., 8-bit), while applying standard low-bit quantization (e.g., 3-bit or 4-bit) to the rest.

Figures 3.7 and 3.8 present the empirical results of this hybrid quantization approach, applied to both weight-activation and weight-only quantization strategies. In both settings,

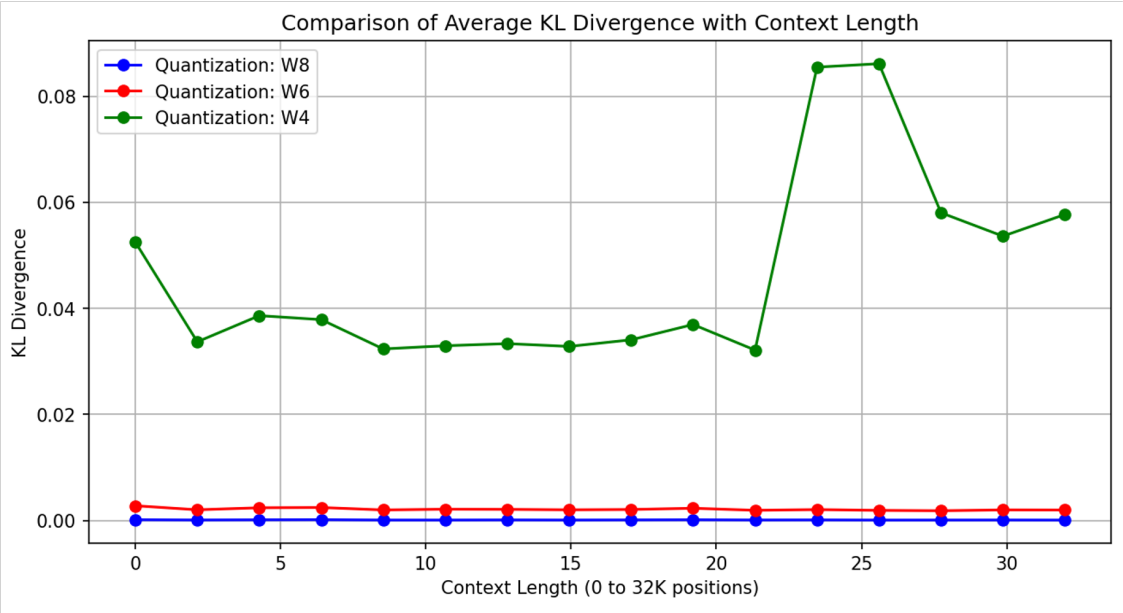


Figure 3.4: KL divergence under different bit-widths for weight-only quantization. W4 again exhibits sharp error growth under long context.

the revised quantization scheme significantly reduces KL divergence across the entire context range, effectively mitigating the error accumulation problem observed in Figures 3.2–3.4.

These findings validate our hypothesis: a small fraction of high-magnitude weights are indeed more error-sensitive, and preserving them at higher precision enables long-context robustness without materially increasing overall model size or compute.

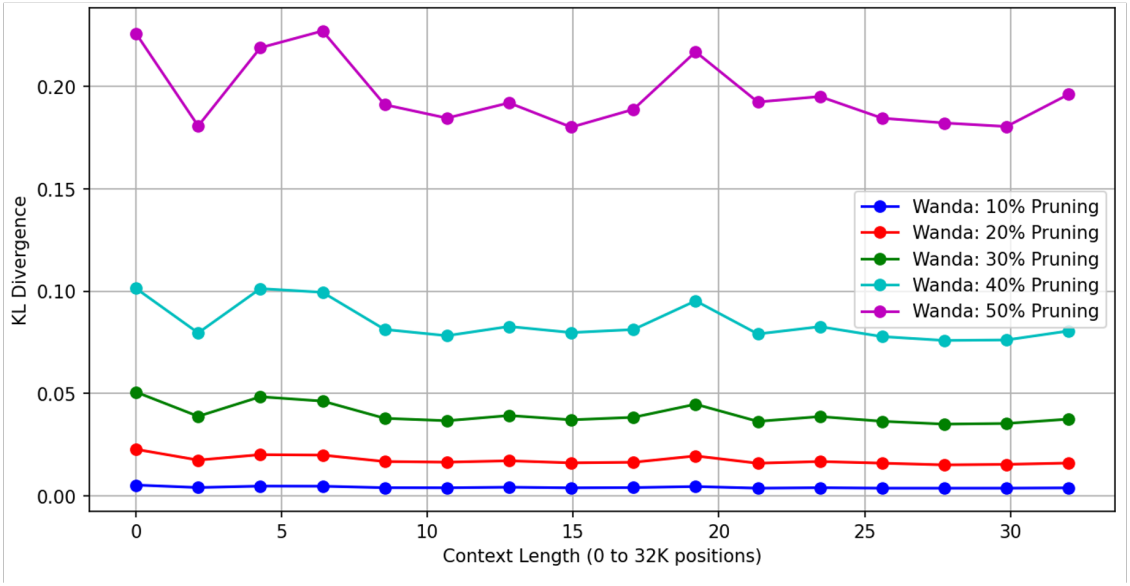


Figure 3.5: KL divergence for Wanda pruning across sparsity levels. Error remains flat across varying context lengths, indicating robustness.

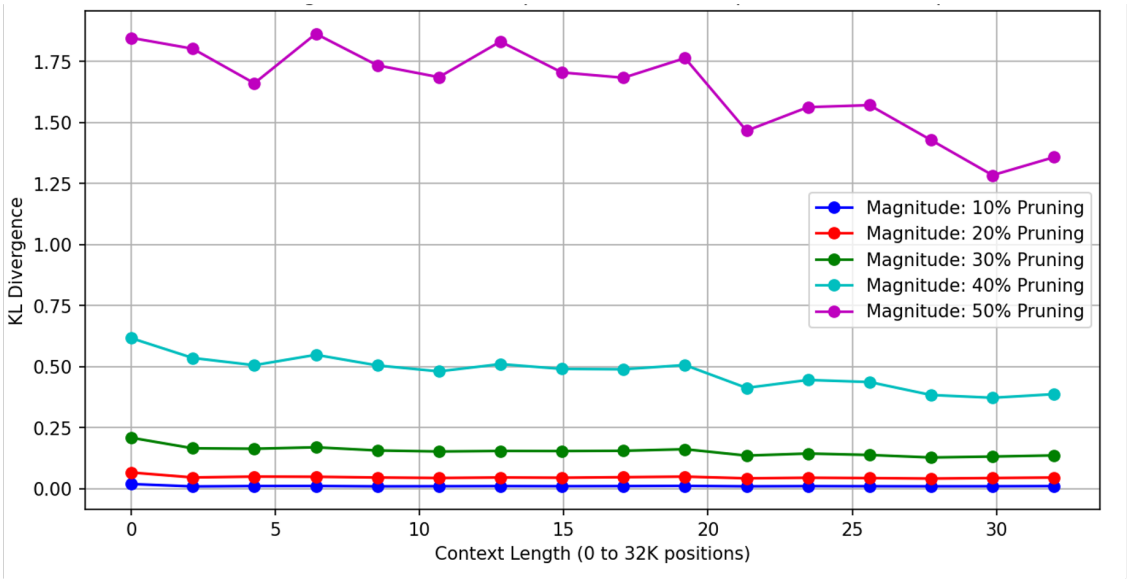
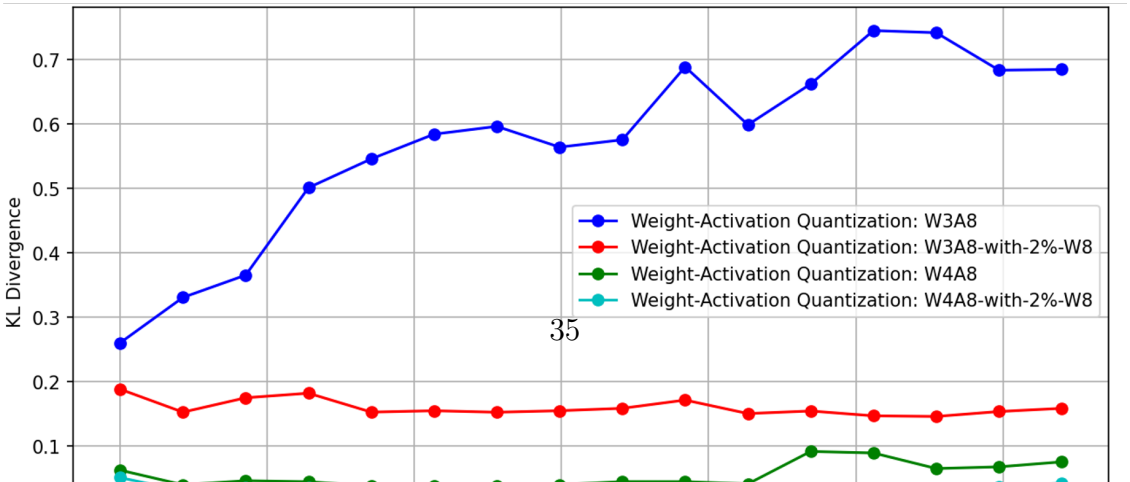


Figure 3.6: KL divergence for magnitude pruning. Similar to Wanda, context length has negligible effect on divergence.



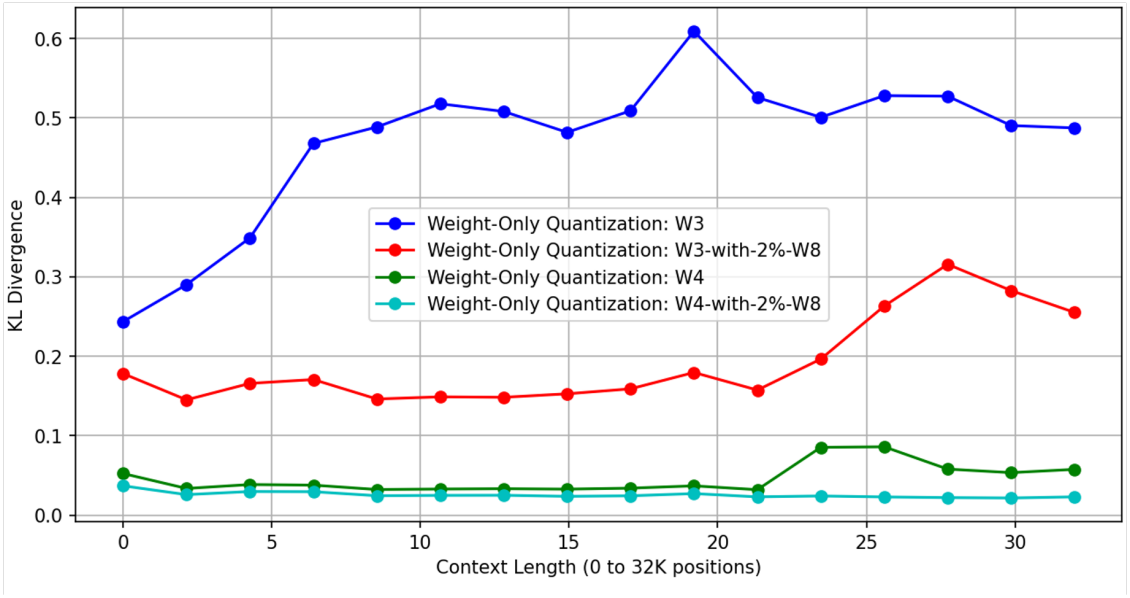


Figure 3.8: Hybrid weight-only quantization (e.g., W4 with 2% W8): similar stabilizing effect is observed, especially compared to naive low-bit quantization.

Chapter 4

Experiments

In this chapter, we present a comprehensive set of experiments designed to evaluate the effectiveness of the proposed optimization strategies for cognitive agents. Our evaluation covers both multi-model and single-model perspectives, reflecting the dual focus of this thesis: improving collaborative reasoning among LLMs and enhancing the computational robustness of individual LLMs under extended input contexts.

We begin by introducing the experimental setup, including the model configurations, datasets, and evaluation metrics. We then evaluate the proposed multi-model collaboration strategies, analyzing how complexity-aware and capability-aware collaboration improves performance and efficiency compared to conventional baselines across both frontier and lightweight models. In the second part of the chapter, we shift focus to single-model compression under long-context conditions, where we empirically assess the limitations of low-bit quantization and demonstrate the benefits of hybrid quantization techniques that preserve a subset of large weights at higher precision. Finally, we show the projected results to combine these two optimization layers together.

4.1 Experimental Setup

To systematically evaluate the proposed optimization strategies for cognitive agents, we conduct extensive experiments using two categories of LLMs: frontier models and computationally efficient models.

For the frontier models, we use DeepSeek V3 [74], Gemini 2.0 Flash [75], and GPT-4o [76], which represent state-of-the-art performance and are accessed via their official APIs to ensure accurate measurements of token-level compute and response correctness. For computationally efficient models, we select widely accessible smaller models, including Mistral-8x7B [77], LLaMA-3.1-8B [78], and Gemma-2-27B [79], known for their reduced inference costs and practical deployment advantages.

We evaluate the performance of our proposed methods on two challenging benchmarks: the MATH dataset, designed for advanced mathematical reasoning tasks [80], and the GPQA dataset, which focuses on general-purpose scientific reasoning [81]. These datasets are selected for their ability to rigorously assess model reasoning capabilities, providing diverse complexity levels suitable for examining the adaptive benefits of our proposed collaboration strategies.

Performance metrics for our experiments include accuracy—defined as the percentage of correct answers provided by the models—and computational efficiency, measured by total token usage (sum of prompt and response tokens) during inference. For assessing compression robustness under long-context conditions, we use KL divergence as an additional metric, quantifying the deviation of compressed model outputs from their uncompressed counterparts.

All experiments involving model compression are conducted using the LLaMA-2-7B-32K model. This choice is consistent with the prior empirical evaluations described in our zero-shot long-context compression study, providing continuity and allowing us to directly validate theoretical hypotheses about the accumulation of computational errors in compressed models.

4.2 Multi-Model Collaboration Results

To rigorously evaluate the effectiveness of the proposed cognitive agent strategies leveraging multi-model collaboration, we conducted experiments utilizing two groups of LLMs: frontier models and computationally efficient models.

4.2.1 Experiments with Frontier LLMs

We first evaluated frontier models—DeepSeek V3, Gemini 2.0 Flash, and GPT-4o—via their official APIs to accurately capture both correctness and computational cost in token usage. Table 4.1 presents the baseline accuracy and computational efficiency of these models on the MATH and GPQA datasets, benchmarks designed to test advanced reasoning capabilities in mathematics and science.

Model	**MATH**		**GPQA**	
	Acc. (%)	**Tokens**	**Acc. (%)**	**Tokens**
DeepSeek V3	87.0	419,513	55.1	173,885
Gemini 2.0 Flash	90.4	361,737	63.6	195,576
GPT-4o	79.8	408,410	51.0	212,037

Table 4.1: Performance of frontier models on the MATH and GPQA datasets.

We applied various multi-model collaboration approaches, comparing standard collaboration methods such as majority voting, Mixture-of-Agents (MoA), Debate, and Verification, against our complexity-aware adaptations. These adaptations dynamically adjust resource allocation based on estimated task complexity, significantly optimizing token efficiency.

The comparative results, summarized in Tables 4.2 and 4.3, illustrate that complexity-aware variants notably enhance computational efficiency while maintaining or slightly improving accuracy. Particularly, the complexity-aware Verification method yielded a token usage reduction of up to 6.26× on the MATH dataset.

Parallel experiments were also conducted with computationally efficient models (Mistral-8x7B, LLaMA-3.1-8B, and Gemma-2-27B). Our proposed hierarchical consistency method

Method	Acc (%)	Token Usage	Rel. Usage	Reduction
Majority Voting	89.8	1,189,660	2.42	–
MoA	88.8	1,387,607	2.82	–
MoA (Ours)	90.4	1,238,803	2.52	1.12
Debate	90.8	2,062,355	4.20	–
Debate (Ours)	90.2	1,379,793	2.81	1.49
Verification	91.6	28,669,198	58.36	–
Verification (Ours)	92.0	4,579,556	9.32	6.26
Oracle	94.8	491,280	1.00	–

Table 4.2: Multi-model collaboration results on the MATH dataset using frontier LLMs.

Method	Acc (%)	Token Usage	Rel. Usage	Reduction
Majority Voting	60.6	581,498	2.10	–
MoA	58.6	720,843	2.61	–
MoA (Ours)	59.6	595,269	2.15	1.21
Debate	65.6	1,960,680	7.09	–
Debate (Ours)	61.1	786,841	2.85	2.48
Verification	64.2	5,318,019	19.24	–
Verification (Ours)	61.6	1,249,373	4.52	4.26
Oracle	78.3	276,337	1.00	–

Table 4.3: Multi-model collaboration results on the GPQA dataset using frontier LLMs.

demonstrated superior performance compared to iterative methods, confirming that lightweight and robust aggregation methods are better suited for models with limited capabilities.

4.2.2 Experiments with Computationally Efficient LLMs

To complement our evaluation of frontier LLMs, we further assess multi-model collaboration strategies on computationally efficient LLMs. These models—including Mistral-8x7B, LLaMA-3.1-8B, and Gemma-2-27B—are chosen for their low inference cost and wide accessibility, making them ideal candidates for deployment in resource-constrained environments.

We adopt the same experimental setup and benchmarks (MATH and GPQA), ensuring a consistent basis for comparison. In this setting, the goal is to examine whether collaborative

strategies designed for powerful agents can still yield benefits when applied to computationally efficient models, or whether alternative, capability-aware approaches are necessary.

Table 4.4 reports the standalone performance of each base model. As shown, their accuracy is substantially lower than frontier models, and the performance gap highlights the limitations of applying standard collaborative protocols such as Mixture-of-Agents, Debate, or Verification without adaptation.

To address this, we explore a lightweight alternative: hierarchical consistency, a capability-aware collaboration strategy tailored to the constraints of computationally efficient models. Rather than relying on inter-agent interactions that require strong instruction-following ability, hierarchical consistency aggregates independently sampled responses from a single model using simple majority voting—thereby reducing both token consumption and system complexity.

Table 4.5 presents the experimental results. Notably, our hierarchical consistency method outperforms all other strategies in MATH accuracy, achieving 63.0%—substantially higher than the next best (Consistency: 57.4%)—and matches the highest-performing method in GPQA (38.89%). These results demonstrate that for computationally efficient agents, simplicity and robustness outweigh complex multi-turn reasoning protocols, which often fail to produce stable gains and significantly increase computational overhead.

In summary, these findings emphasize the value of capability-aware collaboration: cognitive agents built from computationally efficient models benefit more from robust and efficient aggregation schemes than from intricate but unreliable interaction-based strategies. This insight is key to designing practical multi-agent systems in constrained environments.

Model	**MATH Acc (%)**	**GPQA Acc (%)**
LLaMA-3.1-8B	48.6%	23.7%
Mistral-8x7B	31.6%	31.9%
Gemma-2-27B	56.8%	38.8%

Table 4.4: Accuracy of individual computationally efficient models on the MATH and GPQA datasets.

Method	MATH (%)	GPQA (%)	Relative to UB
Majority Voting	56.4	36.6	0.87
MoA	51.4	33.3	0.80
Debate	51.6	36.9	0.81
Verification	48.4	35.4	0.75
Hierarchical Consistency (Ours)	**63.0**	**38.9**	**0.98**
Upper Bound	64.6	39.7	1.00

Table 4.5: Performance of multi-model collaboration methods using computationally efficient LLMs. Accuracy is measured on the MATH and GPQA datasets, with Upper Bound reflecting the oracle choice of the best model per question.

4.3 Single-Model Experiments

This section empirically investigates the behavior of different model compression strategies under long-context conditions and evaluates the effectiveness of our proposed remedy. All experiments are performed using a single LLaMA-2-7B-32K model, chosen for its extended context capacity and suitability for controlled evaluation.

4.3.1 Empirical Evaluation Setup

To measure how compression affects model behavior over long input sequences, we compute the KL divergence between the output logits of compressed models and those of the uncompressed baseline, across varying context lengths. The evaluation is conducted using passages sampled from the WikiText dataset [82], a high-quality corpus for language modeling. This metric captures cumulative distortion introduced by quantization or pruning and provides a consistent baseline for analyzing compression-induced variance.

We evaluate two categories of compression:

- **Quantization:** Including both weight-only and weight-activation schemes. We consider bit-widths ranging from 8-bit (W8) to 3-bit (W3), focusing on low-bit variants due to their greater potential for approximation error.

- **Pruning:** Using both magnitude pruning and the activation-aware Wanda method, sweeping across sparsity levels from 10% to 50%.

The experimental results are summarized in Figures 3.2 to 3.6. As shown in Figure 3.2, 3-bit quantization (W3), especially in the weight-activation setting (W3A8), leads to sharp increases in KL divergence as context length grows. Figures 3.3 and 3.4 demonstrate that this degradation becomes more pronounced at lower bit-widths, with W4 and W3 diverging significantly beyond 24K tokens, whereas W8 remains stable.

In contrast, Figures 3.5 and 3.6 show that pruning strategies—regardless of sparsity level—maintain relatively constant KL divergence across context lengths. This suggests that pruning does not introduce compounding computational errors in the same way that quantization does, confirming our theoretical hypothesis.

4.3.2 Evaluation Setup for Hybrid Quantization

To mitigate the degradation observed in low-bit quantization, we propose a selective quantization strategy that preserves a small subset of large-magnitude weights at higher precision. Specifically, we retain the top 2% of weights—ranked by magnitude—at 8-bit precision, while applying standard low-bit quantization (e.g., 3-bit or 4-bit) to the remainder.

This approach is inspired by the sparsity-aware selectivity of pruning, hypothesizing that larger weights contribute disproportionately to inference stability under long contexts.

Figures 3.7 and 3.8 confirm the effectiveness of this remedy. In both the weight-activation and weight-only cases, hybrid quantization significantly flattens the KL divergence curve, eliminating the error spikes seen in Figures 3.2–3.4. These results validate our hypothesis that preserving a small fraction of critical weights at higher precision can robustly stabilize inference without sacrificing compression efficiency.

4.4 Evaluation of Combined Single- and Multi-Model Optimization

This section estimates the compute savings obtained when the complexity-aware collaboration protocols of Section 4.2 are executed on models compressed with the hybrid 4-bit quantization scheme introduced in Section 4.3. The total cost of inference is written as

$$\text{Cost} = N_{\text{tok}} \times c_{\text{tok}},$$

where N_{tok} is the number of tokens generated during reasoning and c_{tok} is the floating-point operation (FLOP) count required to process one token. Complexity-aware collaboration reduces N_{tok}; quantization lowers c_{tok}. Treating the two factors as approximately independent yields a first-order estimate of the overall reduction.

The FP16 baseline performs one 16-bit multiply–accumulate per weight. After quantization, 98% of weights and activations are represented with 4 bits, while the remaining 2 % of high-magnitude weights are kept at 8 bits. The average bit-width product therefore becomes

$$0.98 \times 4 \times 4 + 0.02 \times 8 \times 8 = 16.96,$$

compared with $16 \times 16 = 256$ for FP16. The ratio $16.96/256 \approx 0.066$ indicates that each token now requires only about 6.6 % of the original bit-operations, corresponding to a 15.1× reduction in arithmetic FLOPs.

Table 4.2 shows that the standard Verification pipeline consumes 58.36 times the oracle token budget on MATH, whereas complexity-aware Verification reduces this to 9.32 times, a 6.26× saving. Similar measurements give factors of 1.49 for Debate and 1.12 for MoA when their complexity-aware variants are used.

Multiplying the per-token and token-count reductions provides an end-to-end estimate:

Method	Token ratio	Per-token FLOP ratio	Overall reduction
MoA (Ours)	1.12	15.1	16.9
Debate (Ours)	1.49	15.1	22.5
Verification (Ours)	6.26	15.1	94.5

Table 4.6: End-to-end compute reduction relative to the FP16 baseline obtained by combining hybrid 4-bit quantization with complexity-aware collaboration. The overall reduction is the product of the token ratio and the per-token FLOP ratio.

For tasks dominated by the Verification pipeline, the combined approach therefore lowers the total FLOP budget by almost two orders of magnitude while preserving, and in some cases slightly improving, answer accuracy. Even for lighter pipelines such as MoA or Debate, the reduction remains between seventeen- and twenty-three-fold, sufficient to shift many long-context workloads from multi-GPU clusters to a single modern accelerator.

Chapter 5

Analysis of Interaction-based Methods on Small LLMs

In this chapter, we delve into an analysis into why smaller, computationally efficient language models often underperform when employing sophisticated interaction-based collaboration methods compared to simpler consistency-based aggregation strategies. Drawing upon detailed dialogues recorded from inter-model interactions during our empirical studies, we dissect the failure modes inherent to interaction-based frameworks, revealing critical insights into the limitations of small LLMs in handling complex, iterative reasoning processes.

5.1 Diagnostic Analysis of Inter-model Dialogues

To systematically investigate why interaction-based collaboration methods fail to yield performance improvements when applied to smaller, computationally efficient LLMs, we specifically focus our analysis on the widely-adopted LLM-Debate framework applied to the MATH dataset. During our primary experimental setup (Section 4.1), we comprehensively preserved detailed inter-model dialogues generated by debates among computationally efficient LLMs—specifically, Mistral-8x7B, LLaMA-3.1-8B, and Gemma-2-27B. These

models were selected for their reduced inference costs and practical advantages in resource-constrained scenarios, as detailed previously.

Due to the sheer volume of dialogues produced, manual analysis proved infeasible. To address this challenge, we leveraged a frontier model—GPT-4.5—as an automated diagnostician to systematically examine and interpret the dialogue records. The frontier model was instructed to analyze incorrect responses specifically, categorizing various reasoning errors, miscommunications, or cognitive limitations exhibited during iterative debates among computationally efficient models.

This diagnostic approach facilitated an efficient and systematic distillation of qualitative insights from extensive interaction data, providing a rigorous empirical foundation for subsequent theoretical discussions on the capability limitations of small LLMs within complex interactive frameworks.

5.2 Diagnosed Causes of Interaction-based Method Failures

Based on the diagnostic analysis of inter-model dialogues generated during the application of the LLM-Debate framework to the MATH dataset, we identify several distinct failure modes. These issues elucidate why smaller, computationally efficient models (Mistral-8x7B, LLaMA-3.1-8B, and Gemma-2-27B) often underperform when employing complex interaction-based collaboration methods compared to simpler consistency-based strategies. Specifically, we categorize the identified failure modes into four primary reasons, elaborated as follows:

5.2.1 Failure to Adhere to Prompt Instructions

A common issue identified in the recorded dialogues was the failure of computationally efficient models to strictly follow prompt instructions, particularly concerning answer format-

ting. For instance, models frequently misplaced answers or embedded redundant formatting instructions within their final responses, resulting in outputs that were structurally incorrect. A representative example from our recorded dialogues is:

> Therefore, the final answer in LaTeX, wrapped in $\boxed{\ldots}$ is: $\boxed{45}$

Such errors indicate difficulties in accurately interpreting and executing explicit formatting guidelines, contributing significantly to the interaction failures.

5.2.2 Superficial Engagement in Debate

Another significant failure observed was the lack of substantive debate engagement by computationally efficient models. Rather than critically analyzing or addressing counterarguments from peer models, small models frequently restated their initial positions without meaningful reconsideration or adaptation. For example, in one recorded dialogue, the model simply repeated the previously given answer $\boxed{n = 5}$ without responding to peer critiques or introducing new reasoning. This superficial engagement undermines the effectiveness of iterative debate processes, leading to limited performance gains from these interactions.

5.2.3 Ineffective or Inconsistent Debate Strategies

Models occasionally employed debate strategies that were logically inconsistent or otherwise ineffective. Notably, we observed scenarios where initially correct reasoning was altered into incorrect approaches in subsequent rounds, or models introduced unrelated errors rather than refining earlier solutions. One illustrative example involved three agents debating a solution using the AM-GM inequality:

- **Round 1**: Agent 1 provided correct usage of AM-GM inequality, Agents 2 and 3 incorrectly criticized it.

- **Round 2**: Agent 1 abandoned the correct reasoning and adopted an incorrect solution, Agent 2 proposed another incorrect solution, while Agent 3 merely verified a previously stated correct solution without meaningful engagement.

Such behaviors indicate underlying limitations in the strategic reasoning capabilities of computationally efficient models during iterative interactions.

5.2.4 Error Propagation and Amplification

Finally, we identified significant error propagation effects within the debate process. Initial inaccuracies introduced by models tended to persist and magnify in subsequent dialogue rounds. For instance, in solving the inequality $\frac{|x^2-81|}{x^2-36x} < 0$, an initial incorrect assessment of critical intervals led to persistent misclassification of subsequent intervals, ultimately resulting in a compounded incorrect final solution. computationally efficient models, possessing limited capabilities in error identification and autonomous correction, were particularly vulnerable to such error propagation, resulting in cumulative degradation of interaction accuracy.

Chapter 6

Conclusion

This thesis presents a dual-perspective optimization framework for enhancing the efficiency and effectiveness of cognitive agents built upon LLMs. From the multi-model perspective, we identify inefficiencies in existing collaboration strategies due to their uniform handling of diverse queries and mismatched model capabilities. To address these issues, we propose complexity-aware collaboration and capability-aware collaboration, which respectively tailor computational strategies to query difficulty and model strengths. These methods achieve up to 6.26× reduction in token usage without sacrificing accuracy, especially under resource constraints.

From the single-model perspective, we analyze the accumulation of computational error in compressed LLMs under long-context settings. While pruning techniques remain robust, we find that low-bit quantization introduces substantial errors as context length grows. We propose a simple yet effective hybrid quantization scheme: selectively retaining 2% of high-magnitude weights at higher precision (e.g., 8-bit), while applying low-bit quantization to the remainder. This targeted approach effectively mitigates error accumulation, ensuring stable performance under long-context conditions.

While this thesis presents an effective dual-layer optimization framework for cognitive agents, several promising avenues remain open for future investigation. First, our current

methods rely on heuristic indicators like answer consistency to assess query complexity and model capabilities. Future research should explore the integration of more explicit meta-cognitive strategies, including uncertainty quantification and self-assessment mechanisms, to enhance adaptive decision-making accuracy. Second, while our evaluations concentrate primarily on reasoning and scientific question answering, extending this framework to more intricate and practical applications, such as hardware architecture design [83], is an important next step. Such domains require sophisticated capabilities like long-horizon strategic planning, multi-modal information processing, and adherence to complex programmatic constraints, thereby providing rigorous benchmarks to assess and improve the robustness and scalability of cognitive agents.

References

[1] OpenAI et al. *OpenAI o1 System Card*. 2024. arXiv: 2412.16720 [cs.AI]. URL: https://arxiv.org/abs/2412.16720.

[2] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[3] Aaron Grattafiori et al. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

[4] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL]. URL: https://arxiv.org/abs/2307.09288.

[5] Gemini Team et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2024. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

[6] Gemini Team et al. *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context*. 2024. arXiv: 2403.05530 [cs.CL]. URL: https://arxiv.org/abs/2403.05530.

[7] Jinze Bai et al. *Qwen Technical Report*. 2023. arXiv: 2309.16609 [cs.CL]. URL: https://arxiv.org/abs/2309.16609.

[8] An Yang et al. *Qwen2 Technical Report*. 2024. arXiv: 2407.10671 [cs.CL]. URL: https://arxiv.org/abs/2407.10671.

[9] Qwen et al. *Qwen2.5 Technical Report*. 2025. arXiv: 2412.15115 [cs.CL]. URL: https://arxiv.org/abs/2412.15115.

[10] An Yang et al. *Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement*. 2024. arXiv: 2409.12122 [cs.CL]. URL: https://arxiv.org/abs/2409.12122.

[11] Theodore R. Sumers et al. *Cognitive Architectures for Language Agents*. 2024. arXiv: 2309.02427 [cs.AI]. URL: https://arxiv.org/abs/2309.02427.

[12] Mieczyslaw M. Kokar, David Brady, and Kenneth Baclawski. "Chapter 13 - The Role of Ontologies in Cognitive Radios". In: *Cognitive Radio Technology (Second Edition)*. Ed. by Bruce A. Fette. Second Edition. Oxford: Academic Press, 2009, pp. 401–428. ISBN: 978-0-12-374535-4. URL: https://www.sciencedirect.com/science/article/pii/B9780123745354000138.

[13] Junlin Wang et al. *Mixture-of-Agents Enhances Large Language Model Capabilities*. 2024. arXiv: 2406.04692 [cs.CL]. URL: https://arxiv.org/abs/2406.04692.

[14] Andries Smit et al. *Should we be going MAD? A Look at Multi-Agent Debate Strategies for LLMs*. 2024. arXiv: 2311.17371 [cs.CL]. URL: https://arxiv.org/abs/2311.17371.

[15] Yilun Du et al. *Improving Factuality and Reasoning in Language Models through Multiagent Debate*. 2023. arXiv: 2305.14325 [cs.CL]. URL: https://arxiv.org/abs/2305.14325.

[16] Shalev Lifshitz, Sheila A. McIlraith, and Yilun Du. *Multi-Agent Verification: Scaling Test-Time Compute with Multiple Verifiers*. 2025. arXiv: 2502.20379 [cs.AI]. URL: https://arxiv.org/abs/2502.20379.

[17] Tri Dao et al. *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness*. 2022. arXiv: 2205.14135 [cs.LG]. URL: https://arxiv.org/abs/2205.14135.

[18] Chejian Xu et al. *From 128K to 4M: Efficient Training of Ultra-Long Context Large Language Models*. 2025. arXiv: 2504.06214 [cs.CL]. URL: https://arxiv.org/abs/2504.06214.

[19] Coleman Hooper et al. *KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization*. 2024. arXiv: 2401.18079 [cs.LG]. URL: https://arxiv.org/abs/2401.18079.

[20] Wei Tao et al. *Cocktail: Chunk-Adaptive Mixed-Precision Quantization for Long-Context LLM Inference*. 2025. arXiv: 2503.23294 [cs.CL]. URL: https://arxiv.org/abs/2503.23294.

[21] Hailin Zhang et al. *PQCache: Product Quantization-based KVCache for Long Context LLM Inference*. 2025. arXiv: 2407.12820 [cs.CL]. URL: https://arxiv.org/abs/2407.12820.

[22] Chenyu Wang, Yihan Wang, and Kai Li. *Evaluating Zero-Shot Long-Context LLM Compression*. 2025. arXiv: 2406.06773 [cs.CL]. URL: https://arxiv.org/abs/2406.06773.

[23] Shervin Minaee et al. *Large Language Models: A Survey*. 2025. arXiv: 2402.06196 [cs.CL]. URL: https://arxiv.org/abs/2402.06196.

[24] Wayne Xin Zhao et al. *A Survey of Large Language Models*. 2025. arXiv: 2303.18223 [cs.CL]. URL: https://arxiv.org/abs/2303.18223.

[25] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[26] Y. Bengio, P. Simard, and P. Frasconi. "Learning Long-term Dependencies with Gradient Descent is Difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166.

[27] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[28] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. *On the difficulty of training Recurrent Neural Networks*. 2013. arXiv: 1211.5063 [cs.LG]. URL: https://arxiv.org/abs/1211.5063.

[29] Jared Kaplan et al. *Scaling Laws for Neural Language Models*. 2020. arXiv: 2001.08361 [cs.LG]. URL: https://arxiv.org/abs/2001.08361.

[30] Stan Franklin and Art Graesser. "Is It an agent, or just a program?: A taxonomy for autonomous agents". In: *Intelligent Agents III Agent Theories, Architectures, and Languages*. Ed. by Jörg P. Müller, Michael J. Wooldridge, and Nicholas R. Jennings. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 21–35. ISBN: 978-3-540-68057-4.

[31] Finale Doshi-Velez and Been Kim. *Towards A Rigorous Science of Interpretable Machine Learning*. 2017. arXiv: 1702.08608 [stat.ML]. URL: https://arxiv.org/abs/1702.08608.

[32] Guanzhi Wang et al. *Voyager: An Open-Ended Embodied Agent with Large Language Models*. 2023. arXiv: 2305.16291 [cs.AI]. URL: https://arxiv.org/abs/2305.16291.

[33] Joon Sung Park et al. *Generative Agents: Interactive Simulacra of Human Behavior*. 2023. arXiv: 2304.03442 [cs.HC]. URL: https://arxiv.org/abs/2304.03442.

[34] Ehud Karpas et al. *MRKL Systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning*. 2022. arXiv: 2205.00445 [cs.CL]. URL: https://arxiv.org/abs/2205.00445.

[35] Shunyu Yao et al. *ReAct: Synergizing Reasoning and Acting in Language Models*. 2023. arXiv: 2210.03629 [cs.CL]. URL: https://arxiv.org/abs/2210.03629.

[36] Zhilin Yang et al. *HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering*. 2018. arXiv: 1809.09600 [cs.CL]. URL: https://arxiv.org/abs/1809.09600.

[37] Mohit Shridhar et al. *ALFWorld: Aligning Text and Embodied Environments for Interactive Learning*. 2021. arXiv: 2010.03768 [cs.CL]. URL: https://arxiv.org/abs/2010.03768.

[38] Noah Shinn et al. *Reflexion: Language Agents with Verbal Reinforcement Learning*. 2023. arXiv: 2303.11366 [cs.AI]. URL: https://arxiv.org/abs/2303.11366.

[39] Mark Chen et al. *Evaluating Large Language Models Trained on Code*. 2021. arXiv: 2107.03374 [cs.LG]. URL: https://arxiv.org/abs/2107.03374.

[40] Shunyu Yao et al. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. 2023. arXiv: 2305.10601 [cs.CL]. URL: https://arxiv.org/abs/2305.10601.

[41] Sirui Hong et al. *MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework*. 2024. arXiv: 2308.00352 [cs.AI]. URL: https://arxiv.org/abs/2308.00352.

[42] Aman Madaan et al. *Self-Refine: Iterative Refinement with Self-Feedback*. 2023. arXiv: 2303.17651 [cs.CL]. URL: https://arxiv.org/abs/2303.17651.

[43] Yongliang Shen et al. *HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face*. 2023. arXiv: 2303.17580 [cs.CL]. URL: https://arxiv.org/abs/2303.17580.

[44] Xuezhi Wang et al. *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. 2023. arXiv: 2203.11171 [cs.CL]. URL: https://arxiv.org/abs/2203.11171.

[45] Xun Liang et al. *Internal Consistency and Self-Feedback in Large Language Models: A Survey*. 2024. arXiv: 2407.14507 [cs.CL]. URL: https://arxiv.org/abs/2407.14507.

[46] Chaithanya Bandi and Abir Harrasse. *Adversarial Multi-Agent Evaluation of Large Language Models through Iterative Debates*. 2024. arXiv: 2410.04663 [cs.CL]. URL: https://arxiv.org/abs/2410.04663.

[47] Tian Liang et al. *Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate*. 2024. arXiv: 2305.19118 [cs.CL]. URL: https://arxiv.org/abs/2305.19118.

[48] Chau Pham et al. *Let Models Speak Ciphers: Multiagent Debate through Embeddings*. 2024. arXiv: 2310.06272 [cs.CL]. URL: https://arxiv.org/abs/2310.06272.

[49] Pranjal Aggarwal et al. *Let's Sample Step by Step: Adaptive-Consistency for Efficient Reasoning and Coding with LLMs*. 2023. arXiv: 2305.11860 [cs.CL]. URL: https://arxiv.org/abs/2305.11860.

[50] Xinglin Wang et al. *Make Every Penny Count: Difficulty-Adaptive Self-Consistency for Cost-Efficient Reasoning*. 2025. arXiv: 2408.13457 [cs.CL]. URL: https://arxiv.org/abs/2408.13457.

[51] Han Wang et al. *Soft Self-Consistency Improves Language Model Agents*. 2024. arXiv: 2402.13212 [cs.CL]. URL: https://arxiv.org/abs/2402.13212.

[52] Yann Dubois et al. *Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators*. 2025. arXiv: 2404.04475 [cs.LG]. URL: https://arxiv.org/abs/2404.04475.

[53] Ge Bai et al. "MT-Bench-101: A Fine-Grained Benchmark for Evaluating Large Language Models in Multi-Turn Dialogues". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2024, pp. 7421–7454. URL: http://dx.doi.org/10.18653/v1/2024.acl-long.401.

[54] Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. *ReConcile: Round-Table Conference Improves Reasoning via Consensus among Diverse LLMs*. 2024. arXiv: 2309.13007 [cs.CL]. URL: https://arxiv.org/abs/2309.13007.

[55] Jinyu Xiang et al. *Self-Supervised Prompt Optimization*. 2025. arXiv: 2502.06855 [cs.CL]. URL: https://arxiv.org/abs/2502.06855.

[56] Jon Saad-Falcon et al. *Archon: An Architecture Search Framework for Inference-Time Techniques*. 2024. arXiv: 2409.15254 [cs.LG]. URL: https://arxiv.org/abs/2409.15254.

[57] Fan Nie et al. *Weak-for-Strong: Training Weak Meta-Agent to Harness Strong Executors*. 2025. arXiv: 2504.04785 [cs.AI]. URL: https://arxiv.org/abs/2504.04785.

[58] Song Han, Huizi Mao, and William J. Dally. *Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding*. 2016. arXiv: 1510.00149 [cs.CV]. URL: https://arxiv.org/abs/1510.00149.

[59] Xunyu Zhu et al. *A Survey on Model Compression for Large Language Models*. 2024. arXiv: 2308.07633 [cs.CL]. URL: https://arxiv.org/abs/2308.07633.

[60] Chenyu Wang et al. *EPIM: Efficient Processing-In-Memory Accelerators based on Epitome*. 2024. arXiv: 2311.07620 [cs.AR]. URL: https://arxiv.org/abs/2311.07620.

[61] Guangxuan Xiao et al. *SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models*. 2024. arXiv: 2211.10438 [cs.CL]. URL: https://arxiv.org/abs/2211.10438.

[62] Xinyin Ma, Gongfan Fang, and Xinchao Wang. *LLM-Pruner: On the Structural Pruning of Large Language Models*. 2023. arXiv: 2305.11627 [cs.CL]. URL: https://arxiv.org/abs/2305.11627.

[63] Elias Frantar et al. *GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers*. 2023. arXiv: 2210.17323 [cs.LG]. URL: https://arxiv.org/abs/2210.17323.

[64] Sehoon Kim et al. *SqueezeLLM: Dense-and-Sparse Quantization*. 2024. arXiv: 2306.07629 [cs.CL]. URL: https://arxiv.org/abs/2306.07629.

[65] Ji Lin et al. *AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration*. 2024. arXiv: 2306.00978 [cs.CL]. URL: https://arxiv.org/abs/2306.00978.

[66] Wei Wen et al. *Learning Structured Sparsity in Deep Neural Networks*. 2016. arXiv: 1608.03665 [cs.NE]. URL: https://arxiv.org/abs/1608.03665.

[67] Yihui He, Xiangyu Zhang, and Jian Sun. *Channel Pruning for Accelerating Very Deep Neural Networks*. 2017. arXiv: 1707.06168 [cs.CV]. URL: https://arxiv.org/abs/1707.06168.

[68] Elias Frantar and Dan Alistarh. *SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot*. 2023. arXiv: 2301.00774 [cs.LG]. URL: https://arxiv.org/abs/2301.00774.

[69] Susan Zhang et al. *OPT: Open Pre-trained Transformer Language Models*. 2022. arXiv: 2205.01068 [cs.CL]. URL: https://arxiv.org/abs/2205.01068.

[70] BigScience Workshop et al. *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. 2023. arXiv: 2211.05100 [cs.CL]. URL: https://arxiv.org/abs/2211.05100.

[71] Mingjie Sun et al. *A Simple and Effective Pruning Approach for Large Language Models*. 2024. arXiv: 2306.11695 [cs.CL]. URL: https://arxiv.org/abs/2306.11695.

[72] Anthropic. *Claude 3.5 Sonnet*. https://www.anthropic.com/news/claude-3-5-sonnet. Accessed 2025-04-30. June 2024.

[73] Together Computer. *LLaMA-2-7B-32K Model Card*. https://huggingface.co/togethercomputer/LLaMA-2-7B-32K. Accessed 2025-04-30. 2024.

[74] DeepSeek-AI et al. *DeepSeek-V3 Technical Report*. 2025. arXiv: 2412.19437 [cs.CL]. URL: https://arxiv.org/abs/2412.19437.

[75] Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. *Introducing Gemini 2.0: Our New AI Model for the Agentic Era*. https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/. Accessed 2025-04-30. Dec. 2024.

[76] OpenAI et al. *GPT-4o System Card*. 2024. arXiv: 2410.21276 [cs.CL]. URL: https://arxiv.org/abs/2410.21276.

[77] Albert Q. Jiang et al. *Mixtral of Experts*. 2024. arXiv: 2401.04088 [cs.LG]. URL: https://arxiv.org/abs/2401.04088.

[78] Meta AI. *Introducing Llama 3.1: Our Most Capable Models to Date*. https://ai.meta.com/blog/meta-llama-3-1/. Accessed 2025-04-30. July 2024.

[79] Gemma Team et al. *Gemma 2: Improving Open Language Models at a Practical Size*. 2024. arXiv: 2408.00118 [cs.CL]. URL: https://arxiv.org/abs/2408.00118.

[80] Dan Hendrycks et al. *Measuring Mathematical Problem Solving With the MATH Dataset*. 2021. arXiv: 2103.03874 [cs.LG]. URL: https://arxiv.org/abs/2103.03874.

[81] David Rein et al. *GPQA: A Graduate-Level Google-Proof Q&A Benchmark*. 2023. arXiv: 2311.12022 [cs.AI]. URL: https://arxiv.org/abs/2311.12022.

[82] Stephen Merity et al. *Pointer Sentinel Mixture Models*. 2016. arXiv: 1609.07843 [cs.CL]. URL: https://arxiv.org/abs/1609.07843.

[83] Vijay Janapa Reddi and Amir Yazdanbakhsh. "Architecture 2.0: Challenges and Opportunities". In: *2023 60th ACM/IEEE Design Automation Conference (DAC)*. 2023, pp. 1–2.