

UNITED MINDS OR ISOLATED AGENTS? EXPLORING COORDINATION OF LLMS UNDER COGNITIVE LOAD THEORY

006 **Anonymous authors**

007 Paper under double-blind review

ABSTRACT

013 Large Language Models (LLMs) exhibit a notable performance ceiling on complex,
 014 multi-faceted tasks, as they often fail to integrate diverse information or adhere
 015 to multiple constraints. We posit that this limitation arises when the demands
 016 of a task exceed the LLM’s effective cognitive load capacity. This interpretation
 017 draws a strong analogy to Cognitive Load Theory (CLT) in cognitive science,
 018 which explains similar performance boundaries in the human mind, and is further
 019 supported by emerging evidence that reveals LLMs have bounded working memory
 020 characteristics. Building upon this CLT-grounded understanding, we introduce
 021 ***CoThinker***, a novel LLM-based multi-agent architecture designed to mitigate
 022 cognitive overload and enhance collaborative problem-solving abilities. ***CoThinker***
 023 operationalizes CLT principles by distributing intrinsic cognitive load through
 024 agent specialization and managing transactional load via structured communication
 025 and a collective working memory. We empirically validate ***CoThinker*** on complex
 026 problem-solving tasks and fabricated high cognitive load scenarios, demonstrating
 027 improvements over existing multi-agent baselines in solution quality and efficiency.
 028 Our analysis reveals characteristic interaction patterns, providing insights into the
 029 emergence of collective cognition and effective load management, thus offering a
 030 principled approach to overcoming LLM performance ceilings.

1 INTRODUCTION

031 The increasing capability of Large Language Models (LLMs) is transforming diverse domains, moving
 032 beyond basic text generation towards complex reasoning applications (Chang et al., 2024; Zhao et al.,
 033 2024; Li et al., 2024a). Aligning these powerful models with human intent and fostering effective
 034 thinking patterns is paramount for unlocking their full potential (Shen et al., 2023). In-Context
 035 Learning (ICL) is increasingly employed for this purpose. ICL adapts models via prompt-based
 036 guidance without parameter updates (Brown et al., 2020; Lampinen et al., 2024), shaping thinking
 037 patterns similarly to finetuning while avoiding retraining (Lin et al., 2024; Zhao et al., 2025; Yang
 038 et al., 2024). Its parameter-free flexibility has made ICL a widely adopted approach.

039 While ICL offers flexibility, it suffers from a performance ceiling when applied to multi-faceted tasks
 040 requiring the integration of diverse information sources provided in-context (He et al., 2024; Li et al.,
 041 2023b; Kirk et al., 2023). In such scenarios, when guided by extensive in-context information, LLM
 042 agents frequently exhibit degeneration of thought, lack of diversity, or an inability to follow multiple
 043 requirements (Liang et al., 2023; Huang et al., 2023; Kamoi et al., 2024; Lu et al., 2024). Despite
 044 increasing empirical studies on ICL’s limitations, the root causes remain underexplored. Concurrently,
 045 recent efforts to overcome this ceiling via agent-based solutions have yielded limited success, often
 046 relying on heuristics without a firm cognitive grounding (Liu et al., 2023; Zhang et al., 2024c).

047 To address this gap, we turn to cognitive science for explanatory insight. Similar patterns of
 048 performance degradation under high informational demands have long been studied through the
 049 framework of Cognitive Load Theory (CLT) (Sweller, 2011; 2003). To better understand this
 050 phenomenon in LLMs, we first conduct a pilot study (Section 3). This study provides theoretical
 051 grounding by establishing the cognitive load framework for LLM performance limits and empirically
 052 examines this analogy through measurable proxies for cognitive load effects. Specifically, following

CLT, we define an agent's **Working Memory (WM)** as its intrinsic, capacity-limited ability to simultaneously hold and process information active in its context, analogous to the capacity of the attention mechanism (Baddeley et al., 1986b). Correspondingly, **Cognitive Load (CL)** is the demand that a task places on an agent's WM, largely determined by the complexity and element interactivity of the information presented via ICL. When the CL imposed by a task exceeds the agent's WM capacity, a state of **cognitive overload** occurs. To examine this WM-CL analogy in LLMs, we conduct experiments measuring attention entropy and perplexity as proxies for cognitive load effects, demonstrating that LLMs exhibit patterns consistent with CLT predictions. Furthermore, recent evidence that LLMs exhibit bounded, human-like WM characteristics (Zhang et al., 2024b; Gong et al., 2024) strengthens our core posit: *The performance ceiling of ICL arises when the demands of the in-context information exceed the LLM's effective cognitive load capacity, mirroring the theoretical limits described by CLT.*

Building on the CLT-grounded understanding and the empirical evidence from our pilot study, we present *CoThinker*, a multi-agent ICL architecture designed to mitigate the cognitive overload imposed by complex tasks. *CoThinker* operationalizes CLT principles through three key functions: (i) dynamic thinking style assignment that adapts to task demands rather than fixed roles, distributing intrinsic cognitive load across specialized agents, (ii) a transactive memory system that maintains shared knowledge about agent expertise and task progress, enabling cognitive offloading and reducing redundant processing, and (iii) a communication moderator that balances cognitive similarity for efficient integration with diversity for comprehensive coverage, creating small-world network properties that minimize coordination overhead while maximizing information flow. In sum, we make the following key contributions:

- We first explain ICL failures on multi-faceted tasks as cognitive overload by formalizing a Cognitive Load Theory-based mapping from human working memory to LLM attention and context limits, and provide empirical evidence.
- We then introduce *CoThinker*, a novel multi-agent ICL architecture that operationalizes Cognitive Load Theory through agent specialization, a transactive memory system, and a moderated communication protocol to mitigate cognitive overload.
- We evaluate *CoThinker*'s effectiveness on complex benchmarks across diverse LLMs, demonstrating its superiority over existing baselines. Additionally, we provide a component analysis that underscores its success in managing cognitive load effectively.

2 RELATED WORK

2.1 MULTI-AGENT LLM COLLABORATION

The rise of LLMs has spurred research into multi-agent systems (MAS), where LLMs collaborate to tackle complex problems beyond the scope of single agents (Guo et al., 2024; Wang et al., 2024a; Qian et al., 2025). Current approaches include multi-agent debates for idea exchange and critique (Liang et al., 2023; Lu et al., 2024; Wang et al., 2024b; Du et al., 2023), iterative reflection for self-correction (Shinn et al., 2023; Madaan et al., 2023; Yao et al., 2023), and role-playing or functional specialization, where agents divide tasks in complex domains (Li et al., 2023a; Qian et al., 2023a; Hong et al., 2023). Architecturally, research explores optimal communication topologies (Li et al., 2024b), dynamic agent networks (Liu et al., 2023; Wu et al., 2023), mental set diversity (Liu et al., 2025b), and hierarchical coordination (Zhang et al., 2024a). However, these designs often rely on intuition or communication efficiency, with limited grounding in cognitive theories of collaboration and processing constraints (Pan et al., 2025). While recent work on multi-persona self-collaboration (Wang et al., 2023) and meta-prompting (Suzgun & Kalai, 2024) demonstrates empirical benefits from diversity, these approaches lack functional-level understanding of the cognitive mechanisms underlying their effectiveness, operating primarily at the behavioral level without explicit characterization of internal processing states. Our work, *CoThinker*, directly addresses this gap by operationalizing Cognitive Load Theory to enhance collective problem-solving.

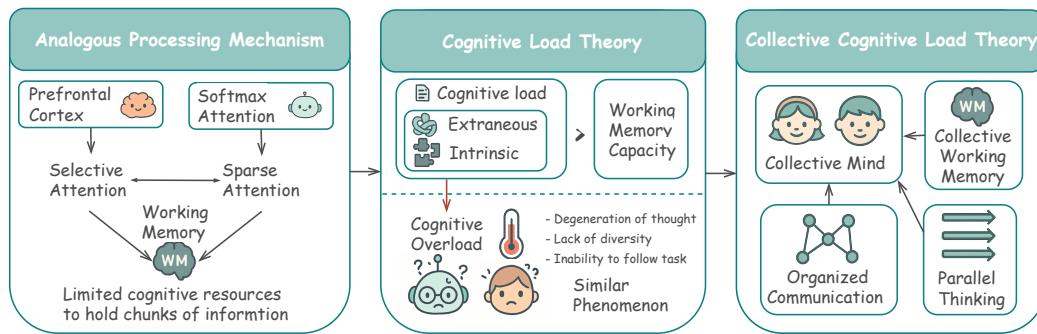
2.2 LLM FOR HUMAN SIMULATION

The capacity of LLMs to exhibit human-like intelligence (Liu et al., 2025a) and emulate nuanced social behaviors (Zhou* et al., 2024) is foundational to their use as artificial agents. Research has

108 demonstrated LLMs' ability to simulate human decision-making (Xie et al., 2024), generate believable
 109 individual and collective behaviors in social simulations (Chuang et al., 2024a), and adopt distinct
 110 personas (Chuang et al., 2024b). Critically, these parallels extend to cognitive characteristics; recent
 111 studies suggest LLMs possess bounded working memory and exhibit failure modes under cognitive
 112 overload akin to humans (Zhang et al., 2024b; Gong et al., 2024), as discussed in our introduction.
 113 Furthermore, interactions between LLM agents can mirror social psychological phenomena (Zhang
 114 et al., 2024c; Guo et al., 2024). This confluence of human-like cognitive traits, including limitations
 115 and social capabilities, provides a strong rationale for applying principles from human cognitive
 116 science—particularly theories like Cognitive Load Theory (CLT) that address cognitive limits—to
 117 the design of more effective LLM-based collaborative systems.

119 3 COGNITIVE FOUNDATIONS FOR ENHANCED LLM PERFORMANCE

121 This section presents our pilot study, which establishes the theoretical foundation for our approach by
 122 linking human cognitive limitations to performance ceilings in LLMs. We introduce a cognitive load
 123 model based on working memory (WM) analogies (Section 3.1), examine this analogy empirically
 124 (Section 3.2), and demonstrate how Cognitive Load Theory (CLT) can address individual limitations
 125 and guide LLM system design (Section 3.3).



138 Figure 1: Cognitive Load framework: Using Cognitive Load Theory (CLT), we align human and
 139 LLM cognitive load via analogy to explain ICL performance ceilings in complex tasks and to guide
 140 methods to mitigate them.

142 3.1 A COGNITIVE LOAD FRAMEWORK FOR LLM PERFORMANCE LIMITS

144 As depicted in the first and second blocks of Fig. 1, we propose a model that draws parallels between
 145 human cognitive limitations and performance ceilings in LLMs.

146 **Analogous Processing Mechanisms.** Human cognition relies fundamentally on working memory,
 147 a capacity-limited cognitive system associated with the prefrontal cortex that employs selective
 148 attention to filter and prioritize information during complex cognitive tasks (Baddeley et al., 1986a;
 149 Cowan, 2010; Miller et al., 1956). LLMs exhibit intriguing functional parallels through their softmax
 150 attention mechanisms, which perform selective focus on input data (Vaswani et al., 2017), with
 151 attention heads specializing in distinct processing patterns (Voita et al., 2019). Recent studies provide
 152 direct evidence that LLMs possess human-like WM characteristics, exhibiting clear limitations on
 153 concurrent information processing with performance degrading predictably as cognitive demands
 154 increase (Zhang et al., 2024b; Gong et al., 2024). More discussion in Appendix B.1.1, B.1.2.

155 **Cognitive Load Theory.** Building upon these working memory analogies, we apply CLT (Sweller
 156 et al., 1998; Sweller, 2011) to interpret LLM performance patterns, which are derived from human
 157 WM in cognitive science. CLT distinguishes Cognitive Load (CL) between *intrinsic load* (determined
 158 by task complexity and element interactivity) and *extraneous load* (arising from instruction presenta-
 159 tion). When the combined load exceeds working memory capacity, *cognitive overload* ensues. LLM
 160 agents demonstrate analogous performance degradation when tasked with complex problems via
 161 In-Context Learning (ICL): tasks requiring extensive multi-step reasoning or integration of numerous
 constraints can lead to degeneration of thought, lack of diversity, or inability to follow multiple

162 requirements (Liang et al., 2023; Huang et al., 2023; Kamoi et al., 2024; Lu et al., 2024). We contend
 163 that such performance ceilings represent cognitive overload, where total demands surpass the LLM’s
 164 effective processing capacity. Examples in Appendix B.1.4.
 165

166 3.2 UNDERSTANDING COGNITIVE LOAD AND WORKING MEMORY IN LLMs 167

168 We empirically examine the analogy of CL and WM in LLMs. We probe into measurable proxies for
 169 cognitive load effects by definition and examine key CLT predictions regarding task and instruction
 170 complexity effects. By definition, WM handles information processing, and cognitive load represents
 171 the *attention* required to handle information within WM, which determines the *easiness* of task
 172 completion. We identified two proxies corresponding to these key characteristics:
 173

174 **Attention Entropy** measures the diversity of the model’s attention distribution, with higher entropy
 175 indicating more distributed attention across input tokens, suggesting the model must consider multiple
 176 aspects of the input, corresponding to higher cognitive load (Zhang et al., 2025). **Perplexity** measures
 177 the model’s certainty of solutions, serving as a proxy for the *easiness* of task completion. For the
 178 *Task Complexity Effect* experiment, we construct Q&A pairs from AMPS Hendrycks et al. (2021a)
 179 with 4 difficulty levels (simple to complex arithmetics), controlling input length for fair comparison.
 180 For the *Instruction Complexity Effect* experiment, we select Q&A pairs from FLASK Ye et al. (2023)
 181 with varying instruction complexity, measuring perplexity on answers for both hard and easy tasks.
 See Appendix B.2 for details of the experimental setup, results, and discussion of these proxies.
 182

182 Experiment 1: Task Complexity Effect.		183 Experiment 2: Instruction Complexity Effect.		
184 Task	185 Attention Entropy	186 Instruction	187 PPL (Hard)	PPL (Easy)
Level 1	4.44	Level 1	120.50	3.37
Level 2	4.80	Level 2	88.97	3.42
Level 3	5.04	Level 3	85.35	3.45

188 Table 1: Pilot study results: (Left) Attention entropy increases with task complexity, indicating higher
 189 cognitive load. (Right) Perplexity patterns are aligned with CLT predictions: instructions help reduce
 190 cognitive load for hard tasks but show no benefit for easy tasks.
 191

192 The results provide strong empirical support for our CLT-LLM analogy. Attention entropy increases
 193 with task complexity (Table 1, left), indicating that harder tasks require the model to simultaneously
 194 consider more information pieces, corresponding to higher cognitive load. For perplexity (Table 1,
 195 right), hard tasks show decreasing perplexity with instruction complexity, indicating that instructions
 196 help the model focus and reduce cognitive load. Easy tasks show increasing perplexity, suggesting
 197 instructions provide no benefit and may even introduce extraneous load, validating CLT’s redundancy
 198 effect where additional information impairs performance when task demands are within capacity.
 199 This aligns with CLT predictions about cognitive load management.
 200

201 3.3 COLLECTIVE INTELLIGENCE PRINCIPLES FOR COGNITIVE LOAD MANAGEMENT

202 Having empirically validated the analogy, we can now leverage CLT principles to address cognitive
 203 overload in LLMs. As depicted in the final block of Fig. 1, when humans encounter tasks that exceed
 204 individual WM capacity, we employ two strategies: external tools or collective intelligence. For
 205 complex tasks where external tools are insufficient, humans naturally form collaborative cognitive
 206 systems that exceed individual capabilities, leading to the emergence of a *collective mind* that is
 207 more powerful than the sum of all individuals (Woolley et al., 2010; Malone et al., 2010). CLT
 208 provides principled guidance for managing cognitive load within such collective systems, particularly
 209 addressing how the introduction of new agents or coordination mechanisms can introduce extraneous
 210 load that must be carefully balanced. Appendix B.1.3 provides discussion from cognitive science.
 211

212 This collective intelligence effectively manages cognitive load through three core mechanisms
 213 guided by CLT principles: (i) **Division of Cognitive Labor** through parallel thinking, allowing
 214 individuals to focus on specialized aspects of problems, thereby reducing intrinsic load per individual
 215 (Dunbar, 2003); (ii) **Collective Working Memory**, often through Transactive Memory Systems
 where knowledge and responsibilities are distributed, enabling individuals to rely on each other for
 information sharing while managing the extraneous load of coordination (Wegner, 1987; Kirschner

et al., 2018); and (iii) **Structured Communication** that efficiently integrates diverse insights through organized information flow, carefully managing extraneous load to prevent cognitive overload from coordination overhead (Hutchins, 1995). Since LLMs face analogous cognitive limitations and our pilot study demonstrates measurable cognitive load effects, systematically operationalizing these CLT principles in multi-agent LLM systems should analogously provide benefits.

4 CoTHINKER

CoThinker is the operationalization of the CLT principles and collective intelligence mechanisms outlined in Section 3, designed as a multi-agent ICL architecture that systematically manages CL to enhance collaborative problem-solving. Simply aggregating outputs from LLM agents often proves insufficient for complex tasks, as naive collaboration can introduce transactional costs—the cognitive effort required to coordinate, communicate, and integrate—without a corresponding increase in solution quality (Pan et al., 2025). As CLT predicts, these transactional costs can quickly lead to cognitive overload, negating the benefits of parallel thinking (Kirschner et al., 2009; 2018). *CoThinker* addresses these challenges by translating the three core CLT principles into a practical multi-agent framework that creates a "collective mind" capable of distributing CL effectively.

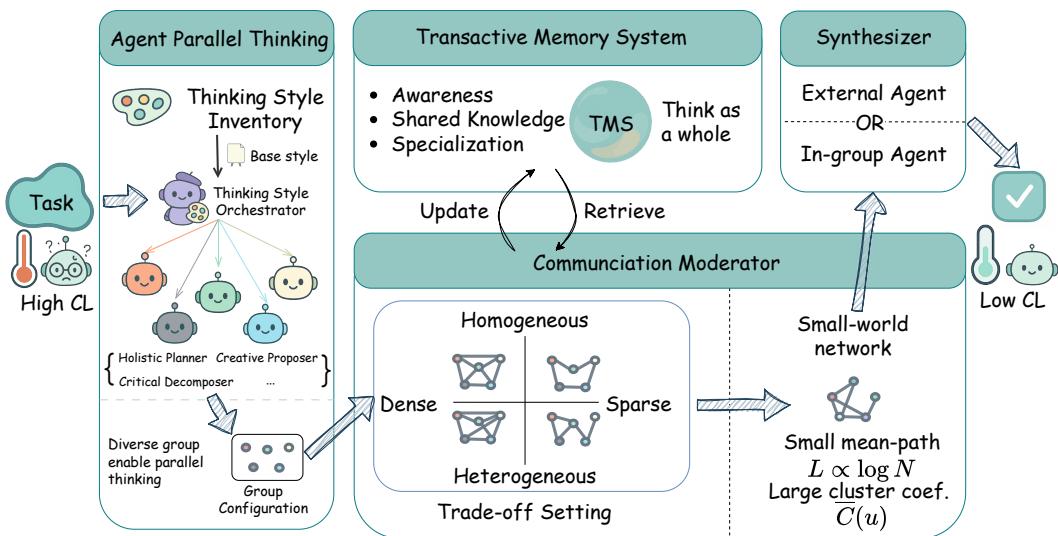


Figure 2: The *CoThinker* Architecture. A high CL task is initially processed by diverse agents via Agent Parallel Thinking. Transactive Memory System facilitates shared understanding by updating and retrieving of collective knowledge. Communication Moderator manages inter-agent information flow, leveraging a trade-off to form a cognitive small-world network, which then feeds into the Synthesizer for final solution, resulting in a lower effective CL for the overall system.

To operationalize these insights, the *CoThinker* architecture (Figure 2) comprises four main modules: Agent Parallel Thinking (Section 4.1), Transactive Memory System (Section 4.2), Communication Moderator (Section 4.3), and Synthesizer (Section 4.4). Each module is directly guided by CLT principles to emulate aspects of the human collective mind. Agent Parallel Thinking fosters initial cognitive diversity, potentially splitting the intrinsic load of the task. The Transactive Memory System boosts inter-agent understanding and tracks consensus, reducing CL from redundant processing. The Communication Moderator balances intrinsic and extraneous loads by structuring information exchange. Finally, the Synthesizer integrates refined collective insights. Let $\mathcal{A} = \{A_1, \dots, A_M\}$ be the set of M agents. Let T_{\max} be the total number of generation rounds. Agent A_i 's output at the end of round t is denoted $x_i^{(t)}$.

4.1 AGENT PARALLEL THINKING

This module promotes a *division of cognitive labor* and *parallel thinking* by assigning diverse thinking styles. Unlike assigning pre-defined roles, which require domain-specific foresight and

270 impose extraneous CL from role adherence, *CoThinker* uses an adaptive approach. A Thinking Style
 271 Orchestrator generates a task-specific style ϕ_i for each agent A_i based on a general base thinking
 272 style inventory ψ (Sternberg, 1997) and the task D :

$$\{\phi_i\}_{i=1}^M = \text{Orch}(D, \psi) \quad (1)$$

273 This yields diverse thinking styles $\{\phi_i, i = 1^M\}$, employed in subsequent stages. Unlike pre-defined
 274 roles requiring complex persona maintenance (extraneous load), thinking styles (Sternberg, 1997)
 275 represent preferred ways of applying capabilities, enabling division of labor with minimal overhead.
 276 Further details on the prompting strategy for style generation and thinking style inventory are in
 277 Appendix D.3.

281 4.2 TRANSACTIONAL MEMORY SYSTEM (TMS)

282 Human groups effectively manage complex information by developing Transactional Memory Sys-
 283 tems (TMS), which involve a shared understanding of who knows what, how to access informa-
 284 tion held by others, and a collective agreement on the information itself (Wegner, 1987), holling-
 285 shead2001cognitive. This distributed cognitive system allows individuals to specialize and rely on
 286 others, reducing individual CL and enhancing group problem-solving (Lewis, 2003). To emulate these
 287 benefits and foster a *collective working memory* in *CoThinker*, we implement a structured mechanism
 288 for maintaining and accessing shared knowledge. This enables cognitive offloading—agents rely
 289 on shared "who knows what" knowledge (Wegner, 1987; Lewis, 2003)—and reduces coordination
 290 overhead by maintaining explicit expertise awareness. Implementation details in Appendix D.4. At
 291 each round t , an evolving representation of the group's collective knowledge, denoted $\mu^{(t)}$, is updated
 292 based on contributions from all agents:

$$\mu^{(t+1)} = \text{UpdateMem}(\mu^{(t)}, \{x_j^{(t)}\}_{j=1}^M) \quad (2)$$

297 4.3 COMMUNICATION MODERATOR

298 Effective inter-agent communication is crucial, yet it incurs transactional costs, the cognitive effort for
 299 message processing and integration, imposing extra extraneous CL. To mitigate these costs, Commu-
 300 nication Moderator selects $N < M$ reference messages for each agent A_i . This process navigates the
 301 critical trade-offs between **Network Density vs. Sparsity** (high exposure vs. information loss) and
 302 **Information Homogeneity vs. Heterogeneity**. The latter involves balancing the ease of integrating
 303 cognitively similar inputs (low extraneous load but risk of echo chambers (Runkel, 1956)) against the
 304 benefits of diverse perspectives for distributing intrinsic load (Aral & Van Alstyne, 2011).

305 **Communication Topology and Algorithm:** The selection of references defines a directed commu-
 306 nication graph $G^{(t-1)} = (\mathcal{A}, E^{(t-1)})$ for each round, where an edge $(A_u, A_v) \in E^{(t-1)}$ exists if
 307 agent A_v receives a message from agent A_u generated in round $t - 1$. Motivated by how small-world
 308 networks efficiently balance local clustering with global connectivity (Watts & Strogatz, 1998), our
 309 moderator employs the following algorithm to construct this graph:

- 310 a. **Fixed In-Degree (N):** Each agent A_i (node A_v) has an in-degree of N , capping its processing
 311 load and respecting LLM WM (Zhang et al., 2024b; Gong et al., 2024).
- 312 b. **Define Cognitive Distance between Agent Outputs:** The cognitive distance $d(x_u^{(t-1)}, x_v^{(t-1)}) =$
 313 $1 - \text{sim}(x_u^{(t-1)}, x_v^{(t-1)})$ is based on the semantic similarity of previous outputs.
- 314 c. **Re-connection via Probabilistic Rewiring (β):** For each agent A_i , its N incoming edges
 315 (references $\mathcal{P}_i^{(t-1)}$) are chosen from cognitively similar peers (low distance), but with a probability
 316 β , "rewiring" some connections to randomly chosen, diverse peers.

317 **Resulting Network Properties and Cognitive Balance:** It fosters dynamic communication networks
 318 with small-world properties, with high local clustering (facilitating efficient refinement of similar
 319 ideas, reducing extraneous load locally) and short average path lengths (enabling rapid global
 320 propagation of diverse insights, aiding intrinsic load distribution). This structure offers a balance
 321 between focused collaboration and broad information access, managing CL more effectively than
 322 random or regular lattice networks. Further details are in Appendix D.5.

324 4.4 SYNTHESIZER
325326 The Synthesizer consolidate all agents answer and TMS into a final answer (details in Appendix D.6).
327**CoThinker Process Flow.** The process for task D with M agents over T rounds:328 *Initialization:*
329

330 $\{\phi_i\}_{i=1}^M = \text{Orch}(D, \psi_i), \quad x_i^{(0)} = \text{Agent}(D, \phi_i), \quad \mu^{(0)} = \text{UpdateMem}(\{x_i^{(0)}\}_{i=1}^M) \quad (3)$
331

332 *Iterative Refinement for agent A_i and round t :*
333

334 $\mathcal{P}_i^{(t)} = \text{SelectRefs}(\{x_k^{(t)}\}, N, \beta) \quad (4)$
335

336 $x_i^{(t+1)} = \text{Agent}(D, \phi_i, \mu^{(t)}, x_i^{(t)}, \mathcal{P}_i^{(t)}), \quad \mu^{(t+1)} = \text{UpdateMem}(\mu^{(t)}, \{x_k^{(t+1)}\}) \quad (5)$

337 *Final Synthesis:*
338

339 $y_{\text{final}} = \text{Synth}(\{x_i^{(T-1)}\}_{i=1}^M, \mu^{(T-1)}, D) \quad (6)$

340 5 EXPERIMENTS AND RESULTS
341342 This section details our experimental methodology and presents the empirical evaluation of *CoThinker*.
343 We first outline the experimental setup, and then present the main results on LiveBench (White et al.,
344 2025) and CommonGen-Hard (Madaan et al., 2023), followed by ablation studies and a discussion of
345 our findings through the lens of Cognitive Load Theory (CLT).
346347 5.1 EXPERIMENTAL SETUP
348349 **Models and Configuration.** For main experiments, we use three Gemini models (Team et al.,
350 2024) with varying capacities: Gemini-1.5-Flash-8B (lightweight), Gemini-1.5-Flash (mid-tier),
351 and Gemini-1.5-Pro (high-capacity). **Evaluation Benchmarks.** We evaluate on two challenging
352 benchmarks: (1) *LiveBench* (White et al., 2025), a comprehensive SOTA benchmark containing
353 *real-world tasks*, providing broad-spectrum evaluation (math, reasoning, data analysis and so on);
354 and (2) *CommonGen-Hard* (Madaan et al., 2023), a controlled *experimental challenge* designed
355 to test information interactivity, by forcing models to integrate target concepts from large pools of
356 distractors. **Baselines.** We compare *CoThinker* with both single-agent and multi-agent approaches:
357 Single Agent (IO), Single Agent (CoT) (Wei et al., 2022), Single Agent (Self-Refine) (Madaan et al.,
358 2023), Multi-Agent Debate (MAD) (Du et al., 2023; Liang et al., 2023), and Diverse MAD (DMAD)
359 (Liu et al., 2025b). Complete details are in Appendices E.3, and E.2.
360

361 5.2 MAIN RESULTS ON LIVEBENCH

362 Table 2 shows *CoThinker* performance across three Gemini models. Scores are normalized by the
363 8B model’s IO baseline. *CoThinker* achieves strong average performance, excelling in complex
364 tasks (Data Analysis, Reasoning, Math) but underperforming on Instruction Following. This pattern
365 reflects distinct tasks with different cognitive load (CL) characteristics: (1) **High intrinsic CL tasks**
366 show clear performance scaling with model capability increases, indicating cognitive bottlenecks,
367 where *CoThinker* excels by distributing intrinsic CL across agents. (2) **Low intrinsic CL tasks** show
368 minimal gains from stronger models. *CoThinker*’s communication overhead introduces extraneous
369 CL that outweighs collaboration benefits, explaining underperformance on execution-focused tasks
370 requiring straightforward adherence rather than reasoning.
371

372 5.3 MAIN RESULTS ON COMMONGEN-HARD

373 *CoThinker* demonstrates improvements on CommonGen-Hard, which tests high CL management.
374 Figure 3 shows performance across evaluation dimensions through (a) a radar plot with normalized by
375 dividing the min scores and (b) an interaction rounds plot tracking performance evolution. *CoThinker*
376 effectively handles high element interactivity by distributing CL across specialized agents and
377 leveraging transactive memory—core CLT principles in action. The radar plot (Figure 3a) reveals
378 *CoThinker*’s strengths in coherence and concept integration, with minor trade-offs in conciseness.
379

Task	Gemini-1.5-Flash-8B					Gemini-1.5-Flash					Gemini-1.5-Pro							
	IO	CoT	SR	MAD	DMAD	Ours	IO	CoT	SR	MAD	DMAD	Ours	IO	CoT	SR	MAD	DMAD	Ours
Math	1.00	1.04	0.92	1.13	1.13	1.11	1.47	1.47	1.45	1.51	1.49	1.57	2.00	1.86	1.93	2.29	2.31	2.40
Data	1.00	0.90	0.34	0.58	0.64	1.32	2.03	2.07	0.90	1.46	2.51	2.44	2.92	2.72	1.33	3.15	3.32	3.39
Reas.	1.00	1.11	0.80	1.21	0.85	1.22	1.63	1.74	1.55	1.92	1.94	1.97	1.87	1.82	1.80	1.78	1.88	1.95
Lang.	1.00	1.09	0.89	1.03	1.02	0.98	1.41	1.30	1.06	1.46	1.44	1.52	1.43	1.54	1.22	1.58	1.74	1.76
Instr.	1.00	1.02	0.81	0.87	0.89	0.80	1.10	1.10	0.87	1.01	1.06	0.99	1.03	1.02	0.72	0.77	1.02	0.95
Avg.	1.00	1.03	0.75	0.97	0.91	1.07	1.53	1.54	1.17	1.47	1.69	1.70	1.85	1.79	1.40	1.92	2.05	2.09

Table 2: LiveBench (White et al., 2025) performance normalized by Gemini-8B (IO). *Ours* refers to CoThinker. Abbreviations: Math, Data Analysis, Reasoning, Language, and Instruction Following.

The rounds plot (Figure 3b) shows sustained improvement across multiple interaction rounds: while baseline methods degrade due to accumulated coordination overhead (extraneous CL).

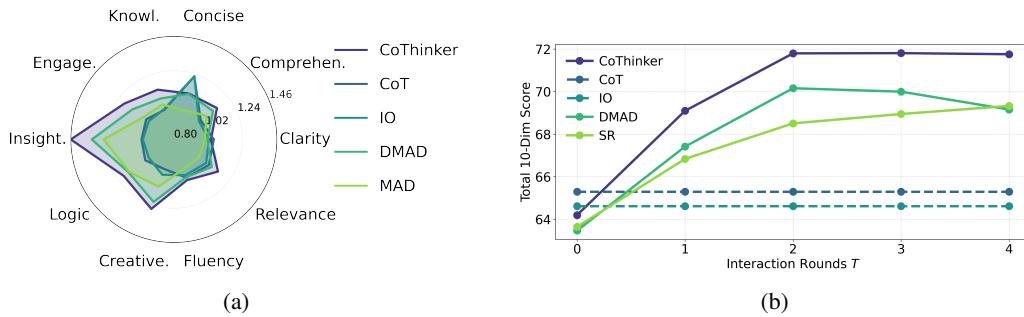


Figure 3: *CoThinker* performance on CommonGen-Hard (Madaan et al., 2023) using Gemini-1.5-Flash. (a) The radar plot illustrates a multi-dimensional performance, showing well-rounded improvement. (b) The rounds plot depicts the total score across rounds, showing stable improvement.

5.4 CROSS-MODEL GENERALIZATION

To validate *CoThinker*’s model-agnostic nature, we evaluated across multiple LLM families: GPT-5-Nano, Qwen3-30B-A3B, GPT-OSS-20B, Gemini-2.5-Flash, GPT-4.1-Mini, Qwen3-32B, and DeepSeek-R1-8B (partial results in Table 3). We examine two scenarios: (1) standard setting with IO baselines using maximum reasoning steps with temperature 0.25 and (2) constrained setting with token budget of 8192 with greedy decoding (temperature = 0). Complete results are in Appendix E.4.

Standard Setting				Constrained Setting			
Model	Method	Math	Reason.	Model	Method	Math	Reason.
GPT-5	CoThinker	88.57	81.88	Gemini-2.5	CoThinker	76.3	69.2
	IO	82.63	68.38		IO	59.3	31.0
Qwen3	CoThinker	80.62	89.50	GPT-4.1	CoThinker	40.0	70.8
	IO	77.50	76.00		IO	34.0	40.8

Table 3: Cross-model evaluation on LiveBench (White et al., 2025) Math and Reasoning (Reason.) subsets in standard setting (left) and constrained setting (right).

5.5 CO THINKER ABLATION OF COMMUNICATION MODERATOR

We ablate the Communication Moderator’s key parameters—reference set size (N), exploration rate (β), and agent count (M)—on Gemini-1.5-Flash-8B across four LiveBench categories: Math,

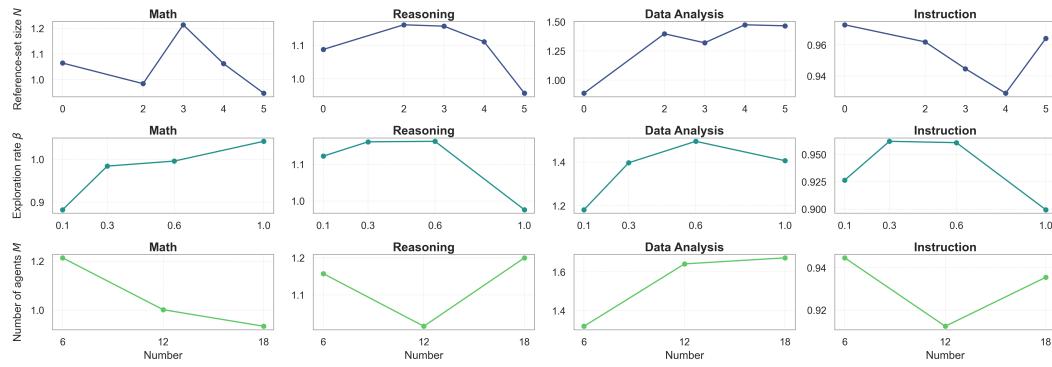


Figure 4: Communication Moderator ablation on *CoThinker* using Gemini-1.5-Flash-8B. **Top:** Reference Set Size ($N \in \{0, 2, 3, 4, 5\}$, $M = 6, \beta = 0.3$); **Middle:** Exploration Rate ($\beta \in \{0.1, 0.3, 0.6, 1.0\}$, $N = 2, M = 6$); **Bottom:** Agent Count ($M \in \{6, 12, 18\}$, $N = 3, \beta = 0.3$).

Reasoning, Data Analysis, and Instruction (See Figure 4). Default parameters: $T = 3$, with controlled variations for each parameter. Scores are normalized by IO baseline. **Analysis.** The Communication Moderator’s parameters directly control CL balance: *Reference set size* (N) manages extraneous load—optimal $N = 2–3$ balances peer input diversity against overload, respecting LLM working memory limits. *Exploration rate* (β) governs similarity-diversity trade-offs: low β (exploiting similar ideas) reduces integration load but risks echo chambers; high β (exploring diverse perspectives) aids intrinsic load distribution but increases extraneous load. Task-dependent optima (e.g., higher β for Reasoning) reflect this balance through small-world network properties. *Agent count* (M) shows non-monotonic performance—more agents distribute intrinsic load but elevate coordination costs, confirming CLT predictions for group overload. These findings validate the Communication Moderator’s role in managing CL for effective collective intelligence. See Appendix E.6 for details.

5.6 COTHINKER ABLATION ON OTHER COMPONENTS

We conducted component ablation on Transactive Memory System (TMS) and Thinking Style Orchestrator (Style) a subset of LiveBench Math tasks. We also examine the proxy, perplexity (PPL), to reflect the CL management effects of our components.

Component Ablation. Communication Moderator is fixed ON for all runs. We test configurations: $TMS \in \{\text{On}, \text{Off}\}$; $Style \in \{\text{On}, \text{Off}\}$. **Analysis.** Our components benefit most models (details in Appendix E.5). From Table 4, we find Thinking Style Orchestrator provides consistent improvements. However, TMS is less effective for new GPT models. Investigating their output, we find that they often refuse to give intermediate results, responding with “I can’t provide step-by-step reasoning.” This is counterproductive, as it discourages the detailed reasoning used to build TMS.

Configuration	Qwen3-30B-A3B	GPT5-Nano	GPT-OSS-20B
TMS: ON, Styles: ON	81.87	55.97	57.15
TMS: ON, Styles: OFF	69.79	49.02	48.21
TMS: OFF, Styles: ON	76.41	62.36	58.37

Table 4: Component ablation on subset of Math dataset, with effect of TMS and Thinking Style Orchestrator, indicating the CL management benefits of each component.

PPL proxy as Evidence. We conducted perplexity (PPL) studies on weaker models to demonstrate how our components helps weaker models reduce CL in understanding stronger models’ outputs. We choose Math and Reasoning tasks for our analysis. Lower PPL indicates higher easiness and effective CL reduction (See Table 5). For more interesting ablation with PPL proxy, see Appendix E.5.

486
487
488
489

Model	Baseline	Styles	TMS	References (N=3)
Qwen3-8B	6.56	3.58	1.69	3.10
Mistral-7B	6.63	5.04	1.58	2.86

490
491 Table 5: PPL ablation showing CL reduction effects: how components help weaker models process
492 information from better models’ answers, reducing CL (lower PPL).
493494

6 CONCLUSION

495
496 This work explains ICL performance ceilings in LLMs through Cognitive Load Theory (CLT),
497 attributing failures on complex tasks to cognitive overload. We establish both theoretical foundations
498 mapping human working memory to LLM attention mechanisms and empirical examination through
499 measurable cognitive load proxies (attention entropy and perplexity), demonstrating that the CLT-
500 LLM analogy provides a principled framework for understanding and addressing performance
501 limitations. We introduce *CoThinker*, a CLT-grounded multi-agent architecture with specialized
502 agents, a transactive memory, and communication moderator to reduce load. On benchmarks with
503 complex tasks, *CoThinker* consistently outperforms strong baselines. Analysis confirm effective
504 cognitive load management, suggesting a principled CLT-based path to more capable collaborative
505 LLM systems. Our work complements recent multi-agent advances (Wang et al., 2023; Suzgun
506 & Kalai, 2024) by providing theoretical grounding in CLT, generating falsifiable predictions and
507 deriving principled design constraints rather than relying on heuristics.
508509

ETHICS STATEMENT

510
511 This research contributes to society and human well-being by advancing scientific understanding
512 of cognitive load principles in AI systems, potentially enabling more capable and efficient collabora-
513 tive AI. We uphold high standards of scientific excellence through rigorous experimental design,
514 transparent reporting of methods and results, and honest acknowledgment of limitations. The work
515 avoids harm by focusing on computational improvements without involving human subjects, sensitive
516 personal data, or applications with direct societal risks. We maintain honesty and transparency by
517 providing complete implementation details, acknowledging all limitations, and declaring no conflicts
518 of interest. The research promotes fairness by using publicly available benchmarks and established
519 evaluation protocols accessible to the broader research community. We respect intellectual property
520 by properly citing all prior work and acknowledging the foundation provided by existing research.
521 Privacy is honored as no personal data collection or processing occurs. All experimental data uses
522 publicly available benchmarks and API-based LLM services under their respective terms of service.
523524

REPRODUCIBILITY STATEMENT

525
526 Our methods rely primarily on existing LLM APIs where we have detailed the specific models used
527 and provided sample prompts and exact agent workflow descriptions in the main paper and appendix.
528 Complete implementation details for reproducing our results are provided, including all agent prompts,
529 communication protocols, system configurations, and evaluation procedures. The benchmark datasets
530 (LiveBench, CommonGen-Hard) are publicly available with standardized evaluation metrics that
enable direct comparison.
531532

REFERENCES

533
534 Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang,
535 Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in
536 Neural Information Processing Systems*, 37:76930–76966, 2024.
537
538 Chenxin An et al. What is wrong with perplexity for long-context language modeling? *arXiv preprint
arXiv:2405.20541*, 2024.
539
540 Sinan Aral and Marshall Van Alstyne. The diversity-bandwidth trade-off. *American journal of
sociology*, 117(1):90–171, 2011.

540 Alan Baddeley, Robert Logie, Sergio Bressi, S Della Sala, and Hans Spinnler. Dementia and working
 541 memory. *The Quarterly Journal of Experimental Psychology Section A*, 38(4):603–618, 1986a.
 542

543 Alan Baddeley, Robert Logie, Sergio Bressi, S Della Sala, and Hans Spinnler. Dementia and working
 544 memory. *The Quarterly Journal of Experimental Psychology Section A*, 38(4):603–618, 1986b.
 545

546 Albert Bandura and Richard H Walters. *Social learning theory*, volume 1. Prentice hall Englewood
 547 Cliffs, NJ, 1977.
 548

549 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal,
 550 Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are
 551 few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
 552

553 Ronald S Burt. Structural holes and good ideas. *American journal of sociology*, 110(2):349–399,
 554 2004.
 555

556 Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan
 557 Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM*
 558 *transactions on intelligent systems and technology*, 15(3):1–45, 2024.
 559

560 Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference
 561 improves reasoning via consensus among diverse llms. *arXiv preprint arXiv:2309.13007*, 2023a.
 562

563 Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia
 564 Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring
 565 emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6, 2023b.
 566

567 Yun-Shiuan Chuang, Agam Goyal, Nikunj Harlalka, Siddharth Suresh, Robert Hawkins, Sijia Yang,
 568 Dhavan Shah, Junjie Hu, and Timothy Rogers. Simulating opinion dynamics with networks of
 569 LLM-based agents. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the*
 570 *Association for Computational Linguistics: NAACL 2024*, pp. 3326–3346, Mexico City, Mexico,
 571 June 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.211.
 572 URL <https://aclanthology.org/2024.findings-naacl.211/>.
 573

574 Yun-Shiuan Chuang, Krirk Nirunwiroy, Zach Studdiford, Agam Goyal, Vincent V. Frigo, Sijia Yang,
 575 Dhavan V. Shah, Junjie Hu, and Timothy T. Rogers. Beyond demographics: Aligning role-playing
 576 LLM-based agents using human belief networks. In Yaser Al-Onaizan, Mohit Bansal, and Yun-
 577 Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp.
 578 14010–14026, Miami, Florida, USA, November 2024b. Association for Computational Linguistics.
 579 doi: 10.18653/v1/2024.findings-emnlp.819. URL <https://aclanthology.org/2024.findings-emnlp.819/>.
 580

581 Nelson Cowan. The magical mystery four: How is working memory capacity limited, and why?
 582 *Current directions in psychological science*, 19(1):51–57, 2010.
 583

584 Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob
 585 Andreas. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint*
 586 *arXiv:2507.16806*, 2025.
 587

588 Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving
 589 factuality and reasoning in language models through multiagent debate. In *Forty-first International*
 590 *Conference on Machine Learning*, 2023.
 591

592 Robin IM Dunbar. The social brain: mind, language, and society in evolutionary perspective. *Annual*
 593 *review of Anthropology*, 32(1):163–181, 2003.
 594

595 Chris Frith and Uta Frith. Theory of mind. *Current biology*, 15(17):R644–R645, 2005.
 596

597 Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are
 598 key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural*
 599 *Language Processing*, pp. 5484–5495, 2021.

594 Dongyu Gong, Xingchen Wan, and Dingmin Wang. Working memory capacity of chatgpt: An
 595 empirical study. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp.
 596 10048–10056, 2024.

597

598 Mark Granovetter. The strength of weak ties: A network theory revisited. *Sociological theory*, pp.
 599 201–233, 1983.

600

601 Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf
 602 Wiest, and Xiangliang Zhang. Large language model based multi-agents: a survey of progress
 603 and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial
 Intelligence*, pp. 8048–8057, 2024.

604

605 Qianyu He, Jie Zeng, Qianxi He, Jiaqing Liang, and Yanghua Xiao. From complex to simple:
 606 Enhancing multi-constraint complex instruction following ability of large language models. In
 607 *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10864–10882, 2024.

608

609 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
 610 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv
 preprint arXiv:2103.03874*, 2021a.

611

612 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
 613 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*,
 614 2021b.

615

616 Andrea B Hollingshead. Cognitive interdependence and convergent expectations in transactive
 617 memory. *Journal of personality and social psychology*, 81(6):1080, 2001.

618

619 Andrea B Hollingshead and David P Brandon. Potential benefits of communication in transactive
 620 memory systems. *Human communication research*, 29(4):607–615, 2003.

621

622 Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao
 623 Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for
 624 a multi-agent collaborative framework. In *The Twelfth International Conference on Learning
 Representations*, 2023.

625

626 Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song,
 627 and Denny Zhou. Large language models cannot self-correct reasoning yet. *CoRR*, 2023.

628

629 Mark D Humphries and Kevin Gurney. Network 'small-world-ness': a quantitative method for
 630 determining canonical network equivalence. *PLoS ONE*, 3(4):e0002051, 2008.

631

632 Edwin Hutchins. *Cognition in the Wild*. MIT press, 1995.

633

634 Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can llms actually correct
 635 their own mistakes? a critical survey of self-correction of llms. *Transactions of the Association for
 636 Computational Linguistics*, 12:1417–1440, 2024.

637

638 Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward
 639 Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and
 640 diversity. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.

641

642 Femke Kirschner, Fred Paas, and Paul A Kirschner. A cognitive load approach to collaborative
 643 learning: United brains for complex tasks. *Educational psychology review*, 21:31–42, 2009.

644

645 Paul A Kirschner, John Sweller, Femke Kirschner, and Jimmy Zambrano R. From cognitive load
 646 theory to collaborative cognitive load theory. *International journal of computer-supported collabora-
 tive learning*, 13:213–233, 2018.

647

648 Asher Koriat. Metacognition and consciousness. In *The Cambridge handbook of consciousness*, pp.
 649 289–325. Cambridge University Press, 2007.

650

651 Andrew Kyle Lampinen, Stephanie CY Chan, Aaditya K Singh, and Murray Shanahan. The broader
 652 spectrum of in-context learning. *arXiv preprint arXiv:2412.03782*, 2024.

648 Kyle Lewis. Measuring transactive memory systems in the field: scale development and validation.
 649 *Journal of applied psychology*, 88(4):587, 2003.
 650

651 Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbulin, and Bernard Ghanem. Camel: Com-
 652 municative agents for "mind" exploration of large language model society. *Advances in Neural*
 653 *Information Processing Systems*, 36:51991–52008, 2023a.

654 Huao Li, Yu Chong, Simon Stepputtis, Joseph P Campbell, Dana Hughes, Charles Lewis, and Katia
 655 Sycara. Theory of mind for multi-agent collaboration via large language models. In *Proceedings of*
 656 *the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 180–192, 2023b.
 657

658 Yingcong Li, Kartik Sreenivasan, Angeliki Giannou, Dimitris Papailiopoulos, and Samet Oymak.
 659 Dissecting chain-of-thought: Compositionality through in-context filtering and learning. In
 660 *Advances in Neural Information Processing Systems*, volume 36, pp. 22021–22046, 2023c.

661 Yinghao Li, Rushi Qiang, Lama Moukheiber, and Chao Zhang. Language model uncertainty
 662 quantification with attention chain. *COLM*, 2025.
 663

664 Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu,
 665 Wenxing Xu, Xiang Wang, Yi Sun, et al. Personal llm agents: Insights and survey about the
 666 capability, efficiency and security. *arXiv preprint arXiv:2401.05459*, 2024a.

667 Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie.
 668 Improving multi-agent debate with sparse communication topology. In *Findings of the Association*
 669 *for Computational Linguistics: EMNLP 2024*, pp. 7281–7294, 2024b.

670 Zhongyang Li, Xiao Ding, and Ting Liu. Generating reasonable and diversified story ending using
 671 sequence to sequence model with adversarial training. In *Proceedings of the 27th International*
 672 *Conference on Computational Linguistics*, pp. 1033–1043, 2018.
 673

674 Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi,
 675 and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent
 676 debate. *arXiv preprint arXiv:2305.19118*, 2023.

677 Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and
 678 Xiang Ren. Commongen: A constrained text generation challenge for generative commonsense
 679 reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp.
 680 1823–1840, 2020.
 681

682 Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu,
 683 Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment
 684 via in-context learning. In *International Conference on Learning Representations*, 2024. URL
 685 <https://arxiv.org/abs/2312.01552>.

686 Xuan Liu, Jie Zhang, Haoyang Shang, Song Guo, Chengxu Yang, and Quanyan Zhu. Exploring
 687 prosocial irrationality for LLM agents: A social cognition view. In *The Thirteenth International*
 688 *Conference on Learning Representations*, 2025a. URL <https://openreview.net/forum?id=u8VOQVzduP>.
 689

690 Yexiang Liu, Jie Cao, Zekun Li, Ran He, and Tieniu Tan. Breaking mental set to improve reasoning
 691 through diverse multi-agent debate. In *The Thirteenth International Conference on Learning*
 692 *Representations*, 2025b.
 693

694 Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. Dynamic llm-agent network: An
 695 llm-agent collaboration framework with agent team optimization. *CoRR*, 2023.

696 Li-Chun Lu, Shou-Jen Chen, Tsung-Min Pai, Chan-Hung Yu, Hung-yi Lee, and Shao-Hua Sun.
 697 Llm discussion: Enhancing the creativity of large language models via discussion framework and
 698 role-play. In *First Conference on Language Modeling*, 2024.
 699

700 Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri
 701 Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement
 with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

702 Thomas W Malone, Robert Laubacher, and Chrysanthos Dellarocas. The collective intelligence
 703 genome. *MIT Sloan management review*, 2010.

704

705 George A Miller et al. The magical number seven, plus or minus two. *Psychological review*, 63(2):
 706 81–97, 1956.

707

708 Melissa Z Pan, Mert Cemri, Lakshya A Agrawal, Shuyi Yang, Bhavya Chopra, Rishabh Tiwari,
 709 Kurt Keutzer, Aditya Parameswaran, Kannan Ramchandran, Dan Klein, et al. Why do multiagent
 710 systems fail? In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*,
 2025.

711

712 Bradley R Postle. Working memory as an emergent property of the mind and brain. *Neuroscience*,
 713 139(1):23–38, 2006.

714

715 Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize
 716 Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. Chatdev:
 717 Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023a. URL
<https://arxiv.org/abs/2307.07924>.

718

719 Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen,
 720 Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. *arXiv
 721 preprint arXiv:2307.07924*, 2023b.

722

723 Chen Qian, Zihao Xie, YiFei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du,
 724 Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Scaling large language model-based
 725 multi-agent collaboration. In *The Thirteenth International Conference on Learning Representations*,
 2025. URL <https://openreview.net/forum?id=K3n5jPkrU6>.

726

727 Philip J Runkel. Cognitive similarity in facilitating communication. *Sociometry*, 19(3):178–191,
 1956.

728

729 Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan
 730 Liu, and Deyi Xiong. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*,
 2023.

731

732 Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion:
 733 Language agents with verbal reinforcement learning. *Advances in Neural Information Processing
 734 Systems*, 36:8634–8652, 2023.

735

736 Garriy Shteynberg, Jacob B Hirsh, Wouter Wolf, John A Bargh, Erica J Boothby, Andrew M Colman,
 737 Gerald Echterhoff, and Maya Rossignac-Milon. Theory of collective mind. *Trends in Cognitive
 Sciences*, 27(11):1019–1031, 2023.

738

739 Robert J Sternberg. *Thinking styles*. Cambridge university press, 1997.

740

741 Mirac Suzgun and Adam Tauman Kalai. Meta-prompting: Enhancing language models with task-
 agnostic scaffolding. *arXiv preprint arXiv:2401.12954*, 2024.

742

743 Mirac Suzgun, Nathan Scales, Nathanael Schärlí, Sebastian Gehrmann, Yi Tay, Hyung Won Chung,
 744 Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. Challenging big-bench tasks and
 745 whether chain-of-thought can solve them. In *Findings of the Association for Computational
 Linguistics: ACL 2023*, pp. 13003–13051, 2023.

746

747 John Sweller. Evolution of human cognitive architecture. *Psychology of learning and motivation*, 43:
 748 216–266, 2003.

749

750 John Sweller. Cognitive load theory. In *Psychology of learning and motivation*, volume 55, pp. 37–76.
 Elsevier, 2011.

751

752 John Sweller, Jeroen JG Van Merriënboer, and Fred GWC Paas. Cognitive architecture and instruc-
 753 tional design. *Educational psychology review*, 10:251–296, 1998.

754

755 Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett
 Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal
 understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

756 Zineddine Tighidet, Andrea Mogini, Hedi Ben-younes, Jiali Mei, Patrick Gallinari, and Benjamin
 757 Piwowarski. Context copying modulation: The role of entropy neurons in managing parametric
 758 and contextual knowledge conflicts. *arXiv preprint arXiv:2509.10663*, 2025.

759

760 Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne, and Henrike Moll. Understanding
 761 and sharing intentions: The origins of cultural cognition. *Behavioral and brain sciences*, 28(5):
 762 675–691, 2005.

763

764 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz
 765 Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing
 766 systems*, 30, 2017.

767

768 Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head
 769 self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint
 770 arXiv:1905.09418*, 2019.

771

772 Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai
 773 Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents.
Frontiers of Computer Science, 18(6):186345, 2024a.

774

775 Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the bounds of
 776 llm reasoning: Are multi-agent discussions the key? In *Proceedings of the 62nd Annual Meeting
 777 of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6106–6131, 2024b.

778

779 Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. Unleashing
 780 cognitive synergy in large language models: A task-solving agent through multi-persona self-
 781 collaboration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language
 782 Processing*, pp. 16816–16834, 2023.

783

784 Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’ networks. *nature*, 393
 785 (6684):440–442, 1998.

786

787 Daniel M Wegner. Transactive memory: A contemporary analysis of the group mind. In *Theories of
 788 group behavior*, pp. 185–208. Springer, 1987.

789

790 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny
 791 Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in
 792 neural information processing systems*, 35:24824–24837, 2022.

793

794 Xuezhi Wen, Deepak Ganapathy, et al. Chain-of-thought reasoning without prompting. *arXiv preprint
 795 arXiv:2025a*.

796

797 Yuxin Wen et al. From sparse dependence to sparse attention: Unveiling how chain-of-thought
 798 enhances transformer sample efficiency. In *International Conference on Learning Representations*,
 799 2025b.

800

801 Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid
 802 Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha,
 803 Siddartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and
 804 Micah Goldblum. Livebench: A challenging, contamination-free LLM benchmark. In *The
 805 Thirteenth International Conference on Learning Representations*, 2025.

806

807 Anita Williams Woolley, Christopher F Chabris, Alex Pentland, Nada Hashmi, and Thomas W
 808 Malone. Evidence for a collective intelligence factor in the performance of human groups. *science*,
 809 330(6004):686–688, 2010.

810

811 Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang,
 812 Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent
 813 conversation. *arXiv preprint arXiv:2308.08155*, 2023.

814

815 Chengxing Xie, Canyu Chen, Feiran Jia, Ziyu Ye, Shiyang Lai, Kai Shu, Jindong Gu, Adel Bibi,
 816 Ziniu Hu, David Jurgens, et al. Can large language model agents simulate human trust behavior?
 817 In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

810 Haoran Yang, Yumeng Zhang, Jiaqi Xu, Hongyuan Lu, Pheng-Ann Heng, and Wai Lam. Unveiling the
 811 generalization power of fine-tuned large language models. In *Proceedings of the 2024 Conference*
 812 *of the North American Chapter of the Association for Computational Linguistics: Human Language*
 813 *Technologies (Volume 1: Long Papers)*, pp. 884–899, 2024.

814 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan.
 815 Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural*
 816 *information processing systems*, 36:11809–11822, 2023.

817 Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo,
 818 James Thorne, Juho Kim, and Minjoon Seo. Flask: Fine-grained language model evaluation based
 819 on alignment skill sets. *arXiv preprint arXiv:2307.10928*, 2023.

820 Youngjin Yoo and Prasert Kanawattanachai. Developments of transactive memory systems and
 821 collective mind in virtual teams. *The International Journal of Organizational Analysis*, 9(2):
 822 187–208, 2001.

823 Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei
 824 Zhang, Anji Liu, Song-Chun Zhu, et al. Proagent: building proactive cooperative agents with large
 825 language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp.
 826 17591–17599, 2024a.

827 Chunhui Zhang, Yiren Jian, Zhongyu Ouyang, and Soroush Vosoughi. Working memory identifies
 828 reasoning limits in language models. In *Proceedings of the 2024 Conference on Empirical Methods*
 829 *in Natural Language Processing*, pp. 16896–16922, 2024b.

830 Jintian Zhang, Xin Xu, Ningyu Zhang, Ruibo Liu, Bryan Hooi, and Shumin Deng. Exploring
 831 collaboration mechanisms for llm agents: A social psychology view. In *Proceedings of the 62nd*
 832 *Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp.
 833 14544–14607, 2024c.

834 Zhisong Zhang, Yan Wang, Xinting Huang, Tianqing Fang, Hongming Zhang, Chenlong Deng,
 835 Shuaiyi Li, and Dong Yu. Attention entropy is a key factor: An analysis of parallel context
 836 encoding with full-attention-based pre-trained language models. In Wanxiang Che, Joyce Nabende,
 837 Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting*
 838 *of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9840–9855, Vienna,
 839 Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi:
 840 10.18653/v1/2025.acl-long.485. URL <https://aclanthology.org/2025.acl-long.485/>.

841 Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang,
 842 Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM*
 843 *Transactions on Intelligent Systems and Technology*, 15(2):1–38, 2024.

844 Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Is in-context
 845 learning sufficient for instruction following in llms? In *International Conference on Learning*
 846 *Representations*, 2025. URL <https://arxiv.org/abs/2405.19874>. To appear in ICLR
 847 2025. Preprint arXiv:2405.19874.

848 Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny
 849 Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint*
 850 *arXiv:2311.07911*, 2023.

851 Xuhui Zhou*, Hao Zhu*, Leena Mathur, Ruohong Zhang, Zhengyang Qi, Haofei Yu, Louis-
 852 Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. So-
 853 topia: Interactive evaluation for social intelligence in language agents. In *ICLR*, 2024. URL
 854 <https://openreview.net/forum?id=mM7VurbA4r>.

855

856

857

858

859

860

861

862

863

864

865

866

867

A LLM USAGE

868

869

870

871

872

873

874

Appendix

875

876

877

B THEORETICAL FOUNDATIONS AND VALIDATION

878

879

B.1 COGNITIVE FOUNDATIONS AND THE CLT-LLM ANALOGY

880

881

Here we elaborate on the cognitive science foundations underpinning our framework and the analogy to LLMs from Section 3.1.

882

883

B.1.1 HUMAN WORKING MEMORY AND ATTENTIONAL CONTROL

884

885

886

887

888

889

890

891

892

893

894

895

Human working memory (WM) is a core cognitive faculty for actively holding and manipulating a limited amount of information relevant to ongoing tasks, operating through attentional mechanisms that select and maintain internal representations, often associated with sustained neural activity in regions like the prefrontal cortex (Baddeley et al., 1986a; Cowan, 2010; Postle, 2006). Given that Large Language Models exhibit emergent sparse attention—where specific attention heads specialize in processing distinct patterns rather than diffusely attending to all input tokens (Vaswani et al., 2017; Voita et al., 2019)—it prompts an intriguing question: does this selective information processing within a finite context window imply the existence of a functional analogue to human WM in LLMs? This emergent selectivity, where not all information in the context is equally weighted or actively processed at any given step, forms a crucial part of the analogy we draw to understand potential capacity limitations and cognitive load phenomena in these models, particularly when handling tasks with high element interactivity through In-Context Learning.

896

897

B.1.2 WORKING MEMORY AND PARAMETRIC KNOWLEDGE IN LLMs

898

899

900

901

902

903

904

905

Misunderstanding 1: Working Memory as Context Window Limits A prevalent misunderstanding in current LLM research equates Working Memory (WM) with physical limitations like context window size and assumes that models only fail when they exceed their maximum token limits—that WM is simply the amount of in-context information that can fit within the context window. However, our framework distinguishes WM as the mechanism of selective attention: a limited cognitive capacity to actively hold and manipulate information simultaneously, not merely passive storage of tokens. This selective attention mechanism can become overwhelmed and lead to performance degradation well before reaching physical context limits.

906

907

908

909

910

911

912

913

914

915

916

Misunderstanding 2: High Cognitive Load as Long Input/Output Tasks Another common misconception is that high cognitive load (CL) tasks are simply those with long inputs and long outputs. This conflates task length with cognitive complexity. In our framework, a high CL task is defined by the working memory demands it places on simultaneous information processing—specifically, the amount of information that must be actively held and manipulated *simultaneously* in WM. For example, a sequential fact elicitation task (e.g., "List 100 countries and their capitals") may have long input and output but requires minimal WM since each fact can be retrieved independently without integrating interdependent information. Conversely, a complex reasoning task requiring simultaneous consideration of multiple interacting constraints, relationships, or variables creates high CL regardless of input/output length, as it demands that the model maintain and manipulate multiple information pieces concurrently in its working memory.

917

Our Framework: Working Memory and Parametric Knowledge Our framework clarifies that an LLM’s parameters can be understood as 1) *Parametric Knowledge* (analogous to human long-term

918 memory, mainly from FFN layer Geva et al. (2021)), and 2) innate generation capabilities that
 919 determine how effectively the model can select and process information for its WM (mainly attention
 920 layer Tighidet et al. (2025)). A model’s parametric knowledge affects how well it understands
 921 and integrates information during reasoning (reflected in attention patterns), while WM serves as
 922 the active processing resource that must load both contextual information and activated parametric
 923 knowledge Tighidet et al. (2025). Complex tasks demanding simultaneous consideration of multiple
 924 interacting elements can cause cognitive overload regardless of context window utilization or task
 925 length. Our pilot study (Appendix B.2) provides empirical validation of this WM framework through
 926 measurable cognitive load proxies.

928 B.1.3 DETAILED MECHANISMS OF HUMAN COLLECTIVE INTELLIGENCE

930 Here we elaborate on the cognitive mechanisms enabling human collective intelligence from Sec-
 931 tion 3.1. Human collective intelligence emerges from sophisticated social-cognitive abilities that
 932 enable groups to surpass individual cognitive limitations through several key mechanisms:

933 **Shared Intentionality and Theory of Mind:** Effective collective intelligence requires individuals to
 934 understand others’ mental states and coordinate intentions toward common goals (Tomasello et al.,
 935 2005; Frith & Frith, 2005), enabling the establishment of common ground necessary for distributed
 936 cognitive processing.

937 **Meta-cognitive Awareness:** Individuals develop meta-knowledge about "who knows what" (Holling-
 938 head, 2001), enabling efficient allocation of cognitive resources and allowing group members to rely
 939 on each other for information sharing and retrieval (Hollingshead & Brandon, 2003).

940 **Spontaneous Organization:** Effective collective intelligence often emerges spontaneously through
 941 self-organizing principles (Shteynberg et al., 2023), with groups naturally developing communication
 942 patterns and role distributions that optimize cognitive load management.

943 **Structured Communication:** Groups develop specialized communication protocols that establish
 944 common ground, minimizing extraneous cognitive load while maximizing information integration
 945 (Tomasello et al., 2005).

946 These mechanisms demonstrate that human collective intelligence results from emergent properties of
 947 social-cognitive interaction that specifically address cognitive load management challenges, providing
 948 natural solutions to cognitive overload that inform LLM multi-agent system design.

951 B.1.4 USING COGNITIVE LOAD THEORY TO EXPLAIN PHENOMENA IN LLM PERFORMANCE

953 Here we provide examples of how CLT explains LLM performance phenomena from Section 3.1.
 954 Cognitive Load Theory (CLT) offers a valuable lens to interpret puzzling LLM performance issues,
 955 positing that LLMs, like humans, have finite processing capacity. Exceeding this capacity leads to
 956 performance degradation. This section concisely analyzes several such cases through CLT.

957 1. **Degradation of Thought in Self-Reflection:** Liang et al. (2023) found LLMs may rigidly stick
 958 to incorrect initial answers during self-reflection, failing to correct meaningfully.

- 959 • *CLT Explanation:* Self-reflection (holding problem, solution, critique, and revision process
 960 concurrently) is highly demanding. If initial analysis already consumes most capacity, the
 961 LLM may lack resources for genuine re-evaluation, defaulting to superficial agreement due to
 962 cognitive overload.

963 2. **Performance Degradation with More In-Context Examples (Many-Shot ICL):** Agarwal et al.
 964 (2024) noted LLM performance can degrade with more in-context examples, especially on
 965 complex tasks (e.g., MATH).

- 966 • *CLT Explanation:* While few examples scaffold, excessive examples increase total cognitive load
 967 beyond capacity. The LLM struggles to synthesize all information, akin to CLT’s "redundancy
 968 effect" where too much information, even relevant, overwhelms working memory.

969 3. **Performance Degradation Despite Increasing "Confidence" (NLL Trends):** Agarwal et al.
 970 (2024) also found that performance degradation in many-shot ICL wasn’t always explained by
 971 NLL (confidence) trends; NLL could improve as performance worsened.

- 972 • *CLT Explanation:* Under cognitive overload, LLMs (like humans) may resort to heuristics.
 973 Overwhelmed by many examples, an LLM might latch onto superficial patterns, yielding outputs

972 that are stylistically plausible (good NLL) but incorrect. This "overconfidence" in a flawed
 973 heuristic stems from an inability to allocate resources for deeper reasoning.

974 **4. Reduced Diversity after RLHF for Instruction Following:** Kirk et al. (2023) and others
 975 observed that RLHF, while improving instruction following, can reduce output diversity.

976 • *CLT Explanation:* Intense RLHF training on narrow preferences imposes a high "germane load"
 977 for conformance. To manage this, and the extraneous load of deviating from rewarded paths,
 978 the model may operate in a constrained output space, reducing the cognitive effort of exploring
 979 diverse (potentially unrewarded) responses. The "cost" of diversity becomes too high.

980 These instances suggest CLT is a powerful analogical framework for understanding LLM limitations
 981 under demanding informational or processing conditions.

983 B.2 PILOT STUDY: EMPIRICAL VALIDATION

985 This section provides detailed implementation and analysis of our pilot study that empirically validates
 986 the Cognitive Load Theory (CLT) analogy for LLMs, as summarized in Section 3.2.

988 B.3 MATHEMATICAL DEFINITIONS OF COGNITIVE LOAD PROXIES

989 **Attention Entropy** measures the diversity of the model's attention distribution, formally defined as:

$$991 \quad 992 \quad 993 \quad H = - \sum_{i=1}^N a_i \log a_i \quad (7)$$

994 where a_i is the normalized attention weight on token i and N is the total number of tokens. Higher
 995 entropy indicates more uniform attention distribution across input tokens, suggesting the model must
 996 consider more aspects of the input simultaneously.

997 **Perplexity** measures the model's uncertainty about its predictions, formally defined as:

$$999 \quad 1000 \quad 1001 \quad \text{PPL} = \exp \left(-\frac{1}{N} \sum_{i=1}^N \log P(w_i) \right) \quad (8)$$

1002 where $P(w_i)$ is the probability assigned to token w_i and N is the sequence length. Lower perplexity
 1003 indicates higher confidence in predictions.

1005 B.4 THEORETICAL JUSTIFICATION FOR PROXIES

1007 The two proxies we employ measure **LLM-perceived cognitive load** that must be accommodated
 1008 within the model's working memory for task completion. This distinction is crucial for understanding
 1009 what we are measuring. Our analogy starts with the observation that the human brain can only attend
 1010 to a limited amount of information at once, a feature known as working memory. In cognitive science,
 1011 working memory capacity refers to how many elements one can simultaneously hold and manipulate;
 1012 these elements are often referred to as "information chunks." A high cognitive load task requires
 1013 more working memory as it demands the simultaneous use of more elements to solve. Therefore, the
 1014 CL of a task can be reflected by how many "information chunks" are needed to solve it.

1015 **Why Attention Entropy? (Measuring "Information Chunks"):** The architecture of LLMs, due
 1016 to their inherent attention mechanism, similarly restricts the amount of information they can focus
 1017 on at once. LLMs predict the next tokens through their attention mechanism, allowing the model to
 1018 selectively focus on specific, relevant parts of the input context to generate a response. By definition,
 1019 we can measure how many "information chunks" are being actively considered to complete a task.
 1020 Thus, using Attention Entropy to measure the sparsity of information integration is a direct proxy:

- 1021 • Low Entropy = Sparse/Focused attention = The model only needs few "elements" to predict
 1022 the answer (Low Load).
- 1023 • High Entropy = Uniform attention = The model is attending to many distinct "elements"
 1024 simultaneously to predict the answer (High Load).

1025 This analogy is justified through both our experiments and recent theoretical work. In our experiments,
 we find attention entropy correlates with task complexity (Table 6); also we find adding reasoning

1026 effectively reduces attention entropy even with longer context. Our analogy also matches recent work
 1027 in theoretically explaining why chain of thoughts works (Wen et al., 2025a); they also find CoT is
 1028 creating more sparse attention; in terms of our framework, it is that reasoning is creating cognitive
 1029 offloading, allowing the model to process fewer chunks during solving the tasks.

1030 **Why Perplexity? (Measuring Processing Fluency):** Perplexity serves as a proxy for processing
 1031 fluency, representing the cognitive ease of processing information. Metacognitive research establishes
 1032 that processing fluency (confidence) correlates with task difficulty and can reflect cognitive load
 1033 faithfully when the subject is not under cognitive overload (Koriat, 2007). In our pilot study, we
 1034 specifically measure perplexity on prefilled ground-truth answers and validate it. In contrast, during
 1035 open generation, cognitive overload causes models to lose metacognitive calibration, often resulting
 1036 in low-perplexity hallucinations where confidence fails to predict performance (An et al., 2024). Both
 1037 LLM phenomena align with cognitive science findings on metacognitive judgments under varying
 1038 load conditions.

1039 B.5 EXPERIMENTAL SETUP

1040 **Model:** We conducted experiments using Mistral-7B-v0.3, a mid-sized language model that exhibits
 1041 clear cognitive limitations while maintaining reasonable performance across diverse tasks.

1042 **Prefilled vs. Generated Answers:** We measure perplexity on prefilled ground-truth answers rather
 1043 than model-generated responses to avoid heuristic confounds: cognitively overloaded models may
 1044 produce confident but incorrect responses (Agarwal et al., 2024), which would not reflect the true
 1045 cognitive load of the underlying task.

1046 **Implementation Details:** For Attention Entropy, we aggregate softmax attention weights, then
 1047 average over heads and layers.

1048 B.6 EXPERIMENT 1: ATTENTION ENTROPY AND TASK COMPLEXITY

1049 **Dataset Construction:** We constructed a controlled dataset from AMPS-Hard, focusing on arithmetic
 1050 reasoning tasks. To ensure fair comparison, we:

- 1051 • Selected 4 difficulty levels (simple arithmetic to complex multi-step problems)
- 1052 • Controlled input length across difficulty levels to isolate complexity effects
- 1053 • Ensured consistent question format and domain (mathematical reasoning)

1054 Different question types are not directly comparable using attention entropy, as the metric is sensitive
 1055 to input structure and content. Therefore, we focused on a single domain with graduated difficulty.

1062 Task Complexity	1063 Attention Entropy (No Reasoning)	1064 Attention Entropy (With Reasoning)
1063 Level 1	4.442	4.439
1064 Level 2	4.796	4.726
1065 Level 3	5.043	4.937
1066 Level 4	6.101	5.920

1067 Table 6: Attention entropy increases with task complexity, validating the proxy. Reasoning steps
 1068 reduce entropy by helping the model focus on key information.

1069 **Analysis.** Attention entropy increases monotonically with task complexity, indicating that harder
 1070 tasks require the model to consider more information pieces simultaneously, corresponding to higher
 1071 cognitive load.

1072 Importantly, this finding is **mathematically non-trivial**. Standard information theory predicts that
 1073 for a probability distribution over N tokens, attention entropy $H = -\sum a_i \log a_i$ should increase
 1074 approximately as $\log N$ when N grows large. Thus, longer sequences naturally yield higher entropy.
 1075 However, Table 6 reveals that adding reasoning steps actually *decreases* attention entropy (e.g., Level
 1076 3: 5.043 → 4.937) despite creating longer context. This seemingly paradoxical result demonstrates that
 1077 reasoning induces *sparse attention patterns*—the model learns to chunk information into coherent
 1078 reasoning steps, allowing it to focus on fewer elements simultaneously even as total context grows.

1080 This finding aligns with recent theoretical work showing Chain-of-Thought creates sparse sequential
 1081 dependencies that enhance processing efficiency, and validates our working memory analogy: just
 1082 as humans chunk information to expand effective WM capacity, LLMs achieve similar cognitive
 1083 offloading through structured reasoning.
 1084

1085 B.7 EXPERIMENT 2: PERPLEXITY AND INSTRUCTION COMPLEXITY

1087 **Dataset Construction:** We used the FLASK dataset, specifically filtering for problems within the
 1088 same category to ensure comparability. We selected pairs of problems where one was marked as
 1089 requiring "expert knowledge" and another without this marking, representing hard and easy tasks
 1090 respectively within the same domain.

1091 Instruction Complexity Levels:

- 1093 • Level 1: Blank (no additional instruction)
- 1094 • Level 2: "Think step by step"
- 1095 • Level 3: "Please think step by step, focusing on factuality and logical reasoning"
- 1096 • Level 4: Extended reasoning guidance with specific cognitive strategies
- 1097 • Level 5: Comprehensive instruction with multiple reasoning frameworks

1098 **Analysis.** For hard tasks, perplexity initially decreases with instruction complexity (Levels 1-
 1099 3), indicating that structured guidance helps the model focus and reduces cognitive uncertainty.
 1100 However, perplexity increases again at higher instruction levels (Levels 4-5), suggesting that excessive
 1101 instruction becomes an additional cognitive burden—classic extraneous load as predicted by CLT.

1102 For easy tasks, perplexity remains consistently low and slightly increases with instruction complexity,
 1103 indicating that additional guidance provides no benefit and may even introduce unnecessary extra-
 1104 neous load. This pattern directly validates CLT's "**redundancy effect**": when a task is within the
 1105 model's working memory capacity, additional information—even if relevant—can impair perfor-
 1106 mance by consuming limited cognitive resources without providing commensurate benefits. The
 1107 differentiated effect across difficulty levels is particularly telling. For hard tasks, instructions re-
 1108 duce perplexity from 120.50 to 85.35, indicating they help the model focus and reduce cognitive
 1109 uncertainty. For easy tasks, the slight increase (3.37→3.45) suggests that processing unnecessary
 1110 guidance creates extraneous load that outweighs any potential benefit when the model's existing
 1111 capacity already suffices.

1112 Instruction Complexity	1113 Perplexity (Hard)	1114 Perplexity (Easy)
1114 Level 1	1115 120.50	3.37
1115 Level 2	1116 88.97	3.42
1116 Level 3	1117 85.35	3.45
1117 Level 4	1118 92.48	3.46
1118 Level 5		3.46

1119 Table 7: Perplexity patterns validate CLT predictions across instruction complexity levels.

1122 B.8 PROXY SCOPE: DIAGNOSTIC, NOT OPERATIONAL

1124 **Why not test-time use.** The proxies we study (attention entropy and perplexity) are not universally
 1125 applicable at inference time. First, when cognitively overloaded, models may fall back on *heuristics*,
 1126 yielding confident but incorrect responses (Agarwal et al., 2024); this breaks the assumption that
 1127 a model's own outputs reveal the true load of the underlying task. Second, perplexity requires
 1128 ground-truth answers, which are unavailable during inference. Together, these factors also clarify
 1129 why using PPL as a test-time assistant tends to fail: heuristic-driven outputs can appear easy for the
 1130 model while still being wrong, and ground-truth-based PPL cannot be computed online.

1131 **How we ensured clean experimental signals.** In our experiments, we treat these proxies strictly as
 1132 *diagnostic* rather than operational signals. To mitigate confounds, we (i) controlled task domain and
 1133 type within each study (addressing cross-task incomparability), (ii) controlled input length within
 each complexity level (addressing length effects on both metrics), and (iii) evaluated on pre-filled

1134 ground-truth answers rather than model-generated outputs to avoid heuristic confounds. Consequently,
 1135 the proxies serve to validate the theoretical foundations, not to drive the system at test time—hence
 1136 *CoThinker* emphasizes architectural solutions over real-time cognitive load monitoring.
 1137

1138 **B.9 POST-HOC ANALYSIS ON MAIN EXPERIMENTS USING PROXIES**

1140 We leverage perplexity as a cognitive load proxy to understand how different *CoThinker* components
 1141 help weaker models process information from stronger models’ outputs. This analysis provides direct
 1142 evidence that our components reduce cognitive burden during collaboration.
 1143

1144 This section (Table 8, Table 10, and Table 9) provides the complete set of PPL ablation studies for
 1145 component analysis referenced in Section 5.6.

1146 **Analysis.**

- 1147 • **Component effects (TMS and Style).** Table 8 shows that both Thinking Style Orchestrator and
 1148 TMS reduce PPL, indicating that each component lowers the cognitive effort required for the
 1149 model to process peer information. Consistent with our captions, TMS yields the largest PPL relief,
 1150 reflecting stronger scaffolding for integrating specialized knowledge.
- 1151 • **Number of references (k).** Table 9 shows PPL decreases as the number of peer answers increases,
 1152 with diminishing returns beyond moderate k . This suggests that models find it easier to understand
 1153 additional similar information, but overly large k risks introducing unnecessary content reducing
 1154 effective using of these answers, as explained in Section 4.3.
- 1155 • **Selection strategy at $k=3$.** Table 10 shows that selecting *similar* peers achieves the lowest PPL,
 1156 random also helps, while fully *diverse* peers increase PPL due to higher integration cost. As
 1157 discussed in Section 4.3 (and Appendix D.5), prioritizing similarity reduces extraneous load but
 1158 risks echo chambers; probabilistic rewiring (β) adds diversity to distribute intrinsic load while
 1159 preserving small-world efficiency.

Model	Base PPL	+Style PPL	+TMS PPL
Mistral-7B	6.6314	5.0422	1.5811
Qwen3-8B	6.5633	3.5814	1.6876

1160
 1161 Table 8: PPL changes from baseline when adding Style or TMS. Both components reduce PPL; TMS
 1162 shows the largest relief.
 1163

k peers	Mistral-7B PPL	Qwen3-8B PPL
1	4.1516	3.3174
2	3.1343	3.1022
3	2.8578	3.0995
4	2.6123	2.5367
5	2.1221	2.1429
6	2.0917	1.8572

1164
 1165 Table 9: PPL vs number of peer answers (k). Increasing k consistently lowers PPL with diminishing
 1166 returns beyond $k \geq 5$.
 1167

Selection ($k=3$)	Mistral-7B PPL	Qwen3-8B PPL
Random (3rand)	2.0967	2.1924
Similar (3sim)	1.8891	1.5399
Diverse (3diverse)	3.0958	2.7905

1168
 1169 Table 10: Effect of selection strategy at $k=3$. Similar peers yield the lowest PPL; random also helps;
 1170 fully diverse peers show higher PPL, consistent with higher integration cost.
 1171

1188 C SMALL-WORLD NETWORK PROPERTIES AND STATISTICAL RIGOR
1189
1190
11911192 C.1 SMALL-WORLD NETWORK ANALYSIS
1193
11941195 To validate that our Communication Moderator creates small-world network properties as predicted
1196 by CLT, we analyze the emergent communication networks using the small-world coefficient $\sigma =$
1197 $(C/C_{rand})/(L/L_{rand})$ (Humphries & Gurney, 2008), where C is clustering coefficient, L is average
1198 path length, and the denominators are random graph baselines.1199 At each round t , we construct a directed weighted graph $G^{(t)} = (\mathcal{A}, E^{(t)}, W^{(t)})$ where edge weights
1200 $w_{uv} = 1 - \text{sim}(x_u, x_v)$ represent cognitive distance. We calculate clustering coefficient, path length,
1201 and compare against 100 random graphs with the same degree distribution.1202 All configurations yield $\sigma > 1$, confirming small-world properties. For $M=6$, $N=3$, $\beta=0.3$: median
1203 $\sigma=2.75$; $M=12$: $\sigma=2.87$; $M=18$: $\sigma=3.12$. This validates that the Communication Moderator success-
1204 fully creates high local clustering (efficient refinement of similar ideas) with short average paths
1205 (rapid global propagation of diverse insights), as predicted by CLT.
1206
1207
1208
12091210 C.2 STATISTICAL SIGNIFICANCE ANALYSIS
1211
12121213 **Bootstrap Standard Error Methodology.** We use bootstrap resampling to estimate the standard
1214 error (SE) of our performance metrics across test instances. We use greedy decoding (temperature=0)
1215 for all refinement rounds to ensure deterministic behavior. Initial generation uses temperature=0.25
1216 to create diverse agent starting points, but subsequent refinement is deterministic. Multiple random
1217 seeds would yield highly similar results under greedy decoding. Therefore, we report bootstrap
1218 variance representing instance-wise variance under large samples rather than seed variance.1219 For each task and model, we: (1) collect raw scores across test instances, (2) perform bootstrap
1220 resampling with replacement ($n=1000$ iterations), (3) compute mean performance for each bootstrap
1221 sample, (4) calculate SE as standard deviation of bootstrap distribution, and (5) construct 95%
1222 confidence intervals using the percentile method.
12231224 **Bootstrap Standard Error Results.**1225 Table 11 shows complete bootstrap SE results for our main comparison. The small sampling variance
1226 across all methods and model families demonstrates high reliability and stability of our bootstrap
1227 estimates.1228 **Statistical Significance on High Cognitive Load Tasks.**1229 We computed 95% confidence intervals for key comparisons on high-CL tasks (Reasoning, Math).
1230 Non-overlapping confidence intervals indicate statistical significance at $p < 0.01$. Statistical signifi-
1231 cance analysis shows:1232
1233

- **Gemini-2.5-Flash on Reasoning:** CoThinker 75.6 (SE=4.15) vs IO 41.5 (SE=4.86). 95%
1234 CIs: [67.5, 83.7] vs [31.9, 51.1]. Non-overlapping CIs confirm $p < 0.01$.
- **GPT-4.1-Mini on Reasoning:** CoThinker 76.4 (SE=3.90) vs IO 47.4 (SE=4.82). 95% CIs:
1235 [68.8, 84.0] vs [37.9, 56.9]. Non-overlapping CIs confirm $p < 0.01$.
- **Qwen3-32B on Reasoning:** CoThinker 30.6 (SE=4.22) vs IO 11.0 (SE=3.14). 95% CIs:
1236 [22.3, 38.9] vs [4.8, 17.2]. Non-overlapping CIs confirm $p < 0.01$.
- **DeepSeek-R1-8B on Reasoning:** CoThinker 20.4 (SE=3.58) vs IO 5.4 (SE=2.00). 95%
1237 CIs: [13.4, 27.4] vs [1.5, 9.3]. Non-overlapping CIs confirm $p < 0.01$.

1238 All comparisons on high-cognitive-load tasks demonstrate statistically significant improvements,
1239 validating CoThinker’s effectiveness at managing cognitive load.
1240

	Model	Method	Math	Data	Reasoning	Language	Inst. Follow.	Overall
1242 1243 1244 1245 1246 1247 1248	Gemini-Flash-8B	IO	0.139	0.146	0.180	0.177	0.038	0.043
		CoT	0.146	0.149	0.185	0.178	0.037	0.042
		SR	0.145	0.116	0.164	0.171	0.043	0.041
		MAD	0.144	0.146	0.186	0.152	0.043	0.043
		DMAD	0.138	0.140	0.189	0.174	0.043	0.042
		CoThinker	0.156	0.140	0.209	0.288	0.040	0.045
1249 1250 1251 1252 1253 1254	Gemini-Flash	IO	0.172	0.148	0.209	0.213	0.036	0.044
		CoT	0.167	0.143	0.211	0.193	0.036	0.045
		SR	0.158	0.103	0.204	0.207	0.043	0.045
		MAD	0.165	0.136	0.215	0.204	0.041	0.045
		DMAD	0.169	0.141	0.216	0.204	0.037	0.044
		CoThinker	0.191	0.147	0.231	0.311	0.038	0.045
1255 1256 1257 1258 1259	Gemini-Pro	IO	0.264	0.136	0.240	0.311	0.034	0.045
		CoT	0.253	0.133	0.233	0.283	0.035	0.046
		SR	0.266	0.116	0.233	0.290	0.036	0.044
		MAD	0.241	0.127	0.249	0.261	0.038	0.046
		DMAD	0.243	0.141	0.239	0.285	0.036	0.046
		CoThinker	0.225	0.133	0.244	0.318	0.038	0.044

Table 11: Bootstrap Standard Error (SE) for all methods across model families, rescaled by 8B IO baseline. Values shown are rescaled SE (original SE divided by 8B IO baseline mean for each task). The small sampling variance (rescaled SE ranging from 0.034 to 0.318) demonstrates high reliability and stability of our bootstrap estimates.

D SYSTEM DESIGN AND IMPLEMENTATION DETAILS

D.1 CO THINKER ARCHITECTURE DETAILS

D.1.1 CO THINKER AS PLUG-AND-PLAY CLT PRINCIPLES

CoThinker’s primary contribution lies in providing **cognitive design principles** derived from Cognitive Load Theory (CLT) that can be integrated into existing multi-agent frameworks as “plug-and-play” enhancements. Rather than competing with established platforms, CoThinker offers a complementary cognitive lens that addresses a fundamental gap: most existing frameworks focus on engineering coordination without explicitly managing cognitive load.

The multi-agent LLM ecosystem encompasses diverse frameworks serving different purposes: **Engineering Platforms** like AutoGen (Wu et al., 2023) and CrewAI focus on conversational flows and role-based task delegation; **Domain-Specific Systems** like MetaGPT (Hong et al., 2023) and ChatDev (Qian et al., 2023b) target software development workflows; **Research Frameworks** like AgentVerse (Chen et al., 2023b) enable experimental exploration; **Workflow Systems** like LangGraph provide stateful execution; and **Consensus Systems** like Multi-Agent Debate (Du et al., 2023) and ReConcile (Chen et al., 2023a) focus on group reasoning.

Unlike these frameworks that primarily address engineering coordination, CoThinker introduces explicit cognitive load management through CLT-grounded design principles. Our approach provides three core cognitive enhancements: (1) dynamic thinking style assignment based on task demands rather than fixed functional roles, (2) communication moderation that balances cognitive similarity and diversity to prevent overload, and (3) transactive memory systems that enable cognitive offloading and specialization. These principles can augment existing frameworks—for instance, enhancing AutoGen’s group chat with load-aware communication, replacing CrewAI’s static roles with adaptive cognitive styles, or integrating TMS-based knowledge sharing into MetaGPT’s development workflows.

D.2 INFORMATION FLOW AND MESSAGE PASSING

This section provides detailed clarification of how information is generated and passed through the CoThinker system, addressing reviewer questions about system dynamics.

1296 **Multi-Channel Information Architecture:** Each agent receives input through two distinct channels
 1297 designed to manage cognitive load:
 1298

1299

 1300 • **TMS Channel:** Provides high-level, synthesized, long-term context (group progress, con-
 1301 sensus, "who knows what"), giving agents strategic awareness with low cognitive burden.
 1302 • **Communication Moderator Channel:** Provides specific, low-volume, current peer outputs
 1303 for immediate reaction and refinement.

1303 **Detailed Information Flow Example:** Consider Agent 1 in round t :

1304 *Step 1: Information Gathering*

1305

 1306 • Communication Moderator selects peers (e.g., Agent 2 & 3) based on cognitive distance and
 1307 network topology
 1308 • Agent 1 accesses TMS to understand group consensus and peer specializations
 1309 • Agent 1 receives specific outputs from selected peers

1310 *Step 2: Cognitive Processing*

1311

 1312 • From TMS: "Agent 2 is performing fact-checking, Agent 3 is identifying mathematical
 1313 formulas"
 1314 • Agent 1 realizes these tasks are covered, enabling cognitive offloading
 1315 • Agent 1 focuses on alternative solution approaches using Agent 3's formulas

1316 *Step 3: Output Generation and System Update*

1317

 1318 • Agent 1 generates new output informed by peer insights and TMS understanding
 1319 • All agent outputs update the TMS for next round
 1320 • Communication Moderator updates network topology based on new cognitive distances

1320 **Network Dynamics:** The communication network reconstructs dynamically each round, with edges
 1321 determined by:

1322

 1323 • Fixed in-degree N (CLT working memory constraint)
 1324 • Cognitive similarity for local clustering (reduces extraneous load)
 1325 • Probabilistic rewiring β for diverse exploration (distributes intrinsic load)

1326 This dual-channel, dynamic network approach distinguishes CoThinker from existing frameworks
 1327 that typically use either broadcast communication or static network topologies.

1328 D.3 PROMPT ARCHITECTURE FOR AGENT PARALLEL THINKING

1329 The Agent Parallel Thinking module in CoThinker aims to foster a beneficial division of cognitive
 1330 labor by assigning diverse thinking styles to agents. This approach is grounded in theories of thinking
 1331 styles, such as Sternberg's Theory of Mental Self-Government (Sternberg, 1997), which posits
 1332 that styles are preferred ways of using one's abilities, not abilities themselves. This distinction is
 1333 crucial: CoThinker leverages thinking styles as preferential orientations for LLM agents, assuming
 1334 the base model possesses a broad set of underlying capabilities. The assigned style guides how these
 1335 capabilities are applied to the task, rather than attempting to imbue a new, fixed skill or enforce a rigid
 1336 behavioral script as a predefined "role" might. This aligns with findings that In-Context Learning
 1337 often modulates an LLM's thinking style rather than altering its core knowledge (Lin et al., 2024;
 1338 Zhao et al., 2025).

1339 Adherence to a flexible thinking style is hypothesized to impose less extraneous cognitive load on an
 1340 LLM agent compared to maintaining a complex, predefined role persona. This allows more of the
 1341 agent's cognitive resources to be dedicated to the primary task. Furthermore, while core thinking
 1342 styles are often seen as relatively stable, they are also understood to be somewhat malleable and can
 1343 be adapted to specific task demands (Sternberg, 1997). CoThinker operationalizes this adaptability
 1344 through a two-stage prompting strategy:

1345 **1. Style Orchestration (Orch function):** The Thinking Style Orchestrator (itself an LLM) is
 1346 provided with the overall task description D and a Thinking Style Inventory. This inventory consists
 1347 of base thinking styles derived from Sternberg's theory, encompassing dimensions such as Functions
 1348 (Legislative, Executive, Judicial), Forms (e.g., Monarchic, Hierarchic), Levels (Global, Local), Scope
 1349 (Internal, External), and Leanings (Liberal, Conservative). The Orchestrator's objective is to generate

1350 a diverse yet task-relevant set of M specific thinking styles $\{\phi_1, \dots, \phi_M\}$, one for each agent A_i .
 1351 For each agent, the Orchestrator takes one or a combination of Sternberg's dimensions as a base style
 1352 ψ_i and adapts it to the given task D . The Orchestrator is guided to ensure the resulting set of styles
 1353 $\{\phi_i\}$ promotes varied perspectives on the problem, reflecting the value of different styles for different
 1354 task facets.

1355 An example prompt for the Orchestrator, given a base combination from Sternberg (e.g., $\psi_i =$
 1356 "Legislative-Global style"):
 1357

1358 Given the primary task: "{Task D}"
 1359 And the base thinking style profile (from Sternberg's Theory of
 1360 Mental Self-Government): "{Base Style profile psi_i, e.g.,
 1361 Legislative function with a Global level preference}"

1362 Generate a concise (1-2 sentences) task-specific adaptation
 1363 of this thinking style profile that would be most beneficial
 1364 for an agent contributing to this primary task. The agent
 1365 should focus its reasoning and output according to this
 1366 adapted style.

1367 Task-Specific Style for an agent:

1368
 1369 This process results in M distinct, task-contextualized thinking styles $\{\phi_1, \dots, \phi_M\}$. By dynamically
 1370 adapting general styles to the specific task, CoThinker aims to harness the benefits of stylistic
 1371 diversity while mitigating risks such as pigeonholing or oversimplification associated with static style
 1372 assignments.

1373 **2. Agent Instruction (Agent function - style incorporation):** Each agent A_i then receives its
 1374 specific thinking style ϕ_i as part of its instruction prompt, guiding its approach throughout the
 1375 problem-solving process. An excerpt of an agent's prompt showing style incorporation:

1376 You are Agent {num}. Your assigned thinking style for this
 1377 task is: "{Style phi_i generated by Orchestrator}".
 1378 The overall task is: "{Task D}".
 1379 [Other contextual information, e.g., from TMS μ^t ,
 1380 references P_i^{t-1} , own previous thought x_i^{t-1}]
 1381

1382 Keeping your assigned thinking style in mind, please provide
 1383 your thoughts/solution:

1384
 1385 This method encourages agents to approach the problem from varied cognitive angles, promoting
 1386 comprehensive exploration of the solution space and distributing the intrinsic cognitive load of the
 1387 task, without the cognitive burden of strict role-playing.

1388 D.4 PROMPT ARCHITECTURE FOR TRANSACTIONAL MEMORY SYSTEM (TMS) EMULATION

1389 As introduced in Section 4.2, CoThinker incorporates a mechanism to emulate a human Transactional
 1390 Memory System (TMS). A TMS is a collective cognitive resource developed by groups, encompassing
 1391 a shared understanding of who knows what (metamemory or expertise directory), how to access
 1392 and integrate this distributed knowledge, and a level of trust in the information provided by different
 1393 members (Wegner, 1987; Hollingshead, 2001; Lewis, 2003). Effective TMS functioning involves
 1394 processes of knowledge *encoding* (assigning information to members or recognizing expertise), *stor-*
 1395 *age* (individuals retaining specialized knowledge), and *retrieval* (accessing and using the distributed
 1396 knowledge), facilitated by member *specialization*, perceived *credibility*, and inter-agent *coordination*
 1397 (Yoo & Kanawattanachai, 2001). This systematic division and integration of cognitive labor allows
 1398 groups to handle more complex information and solve problems more effectively than individuals or
 1399 less coordinated groups.

1400 CoThinker's emulation of TMS centers on the generation and presentation of the collective memory
 1401 state, $\mu^{(t)}$, at each round t . This is not merely an aggregation of past messages but a structured
 1402 synthesis designed to reflect key TMS components. Specifically, an auxiliary LLM agent (the
 1403 "TMS Manager") is tasked with populating a predefined "TMS Template" based on all agent outputs

1404 $\{x_j^{(t-1)}\}_{j=1}^M$ from the previous round and the existing memory state $\mu^{(t-1)}$, to produce the updated
 1405 $\mu^{(t)}$. This template explicitly guides the TMS Manager to synthesize information reflecting:
 1406

- 1407 **1. Expertise Directory ("Who Knows What"):** The template prompts the TMS Manager to list
 1408 the key contributions from each agent A_j in the previous round, often implicitly linking these
 1409 contributions back to their assigned thinking style ϕ_j or emergent problem-solving role. For
 1410 example, $\mu^{(t)}$ might state: *"Agent A (Analytical Thinker) identified three inconsistencies in the*
 1411 *data, while Agent B (Creative Ideator) proposed two novel solutions based on X."* This helps all
 1412 agents maintain an updated awareness of which peer is focusing on, or has provided significant
 1413 input regarding, specific facets of the task. This corresponds to the *encoding* of expertise and
 1414 facilitates targeted *retrieval* cues.
- 1415 **2. Shared Knowledge Store (Consensus and Artifacts):** The template requires the TMS Manager
 1416 to identify and articulate points of emerging consensus, established facts, or partial solutions that
 1417 the group has collectively built. For instance: *"Consensus: The primary bottleneck is resource*
 1418 *allocation. Established: The budget cannot exceed Y."* This component of $\mu^{(t)}$ serves as the
 1419 repository of *stored*, validated collective knowledge, reducing the need for agents to re-derive
 1420 information and providing a foundation for subsequent reasoning.
- 1421 **3. Differential Insights and Unresolved Issues (Focus for Coordination):** A crucial part of the
 1422 TMS template prompts the TMS Manager to highlight discrepancies between agent outputs,
 1423 unresolved questions, conflicting perspectives, or aspects of the problem that remain unaddressed.
 1424 Example: *"Divergence: Agent C suggests strategy Alpha, while Agent D advocates for Beta.*
 1425 *Unresolved: The feasibility of implementing X within the given timeframe."* This explicitly flags
 1426 areas requiring further discussion, debate, or focused problem-solving in the next round, thereby
 1427 guiding inter-agent *coordination* and ensuring that cognitive effort is directed towards the most
 1428 critical, unresolved aspects of the task assigned to most relevant agents.

1429 The structure of $\mu^{(t)}$, as generated by this templated process, is then presented to each agent A_i at the
 1430 beginning of round t as part of its input prompt. An excerpt illustrating this presentation is:

```
1431
1432
1433 [Agent's assigned thinking style: {Style_phi_i}]
1434 [Overall Task: {Task_D}]
1435
1436 Collective Summary from Previous Round (reflecting shared understanding mu^(t)) :
1437 "{Text of mu^(t) generated by the TMS Manager using the TMS Template}"
1438
1439 Your Previous Output (x_i^(t-1)) :
1440 "{Text of x_i^(t-1)}"
1441
1442 Reference Outputs from Peers (P_i^(t-1)) :
1443 Reference 1 (from Agent A_k): "{Text of x_k^(t-1)}"
1444 Reference 2 (from Agent A_l): "{Text of x_l^(t-1)}"
1445 ...
1446
1447 Based on all the above, and keeping your thinking style in mind,
1448 provide your refined thoughts/contribution for the current round:
```

1449
 1450
 1451 This deliberate structuring of $\mu^{(t)}$ to reflect an expertise directory, a shared knowledge store, and a
 1452 pointer to unresolved issues distinguishes CoThinker's approach from simple multi-agent cooperation
 1453 or discussion. While basic cooperation might involve information sharing, it often lacks the systematic
 1454 assignment of knowledge domains, explicit tracking of expertise, and focused mechanisms for
 1455 integrating specialized insights that a TMS provides. CoThinker's TMS emulation aims to create a
 1456 more efficient and powerful "group mind" by embedding these principles directly into the information
 1457 environment of the agents, thereby reducing redundant effort and enhancing the quality of collective
 1458 problem-solving.

1458 D.5 COMMUNICATION MODERATOR: CULTIVATING AN EFFICIENT NETWORK VIA STRONG
1459 AND WEAK TIES
1460

1461 The transactional cost induced by inter-agent communication is a key concern in Collaborative
1462 CLT (Kirschner et al., 2009; 2018). The Communication Moderator in *CoThinker* (Section 4.3)
1463 strategically structures inter-agent communication by implicitly leveraging principles from social and
1464 complex network theories. This design fosters a network optimized for managing cognitive load and
1465 enhancing collective intelligence.

1466 **Local Cohesion via Strong Cognitive Ties and High Clustering** The primary reference selection
1467 mechanism (with probability $1 - \beta$) connects agent A_i to peers whose prior outputs $x_k^{(t-1)}$ are
1468 most cognitively similar to A_i 's own $x_i^{(t-1)}$. This promotes the formation of local clusters where
1469 agents process highly related information. From a social network perspective, these connections
1470 are analogous to **strong ties** (Granovetter, 1983), fostering cohesive subgroups. In network science,
1471 this behavior inherently leads to a high **local clustering coefficient**, indicating dense intra-group
1472 connectivity.

1473 For similarity measurement, we employ cosine similarity of text embeddings (e.g., using
1474 all-MiniLM-L6-v2) to quantify cognitive proximity between agent outputs.

- 1475 • **Rationale:** Such local clustering facilitates focused refinement of shared ideas and reduces the
1476 extraneous cognitive load associated with integrating highly similar information.

1477 **Global Integration via Weak Cognitive Ties and Small-World Properties** Exclusive reliance
1478 on strong ties (i.e., $\beta = 0$) could lead to network fragmentation, where clusters become isolated
1479 "echo chambers." This corresponds to a lack of "bridging capital" across **structural holes** in social
1480 network theory (Burt, 2004), and a long **average path length** in network science, hindering the global
1481 distribution of diverse insights and the effective management of overall intrinsic cognitive load.

1482 The probabilistic "rewiring" mechanism (with probability β) counteracts this by compelling agents to
1483 also reference randomly chosen peers, irrespective of immediate cognitive similarity.

- 1484 • **Mechanism and Analogy:** These random connections function as **weak ties** (Granovetter, 1983),
1485 which are crucial for bridging disparate network segments and transmitting novel information.
- 1486 • **Network Outcome:** Introducing such weak ties into a highly clustered network is a hallmark of
1487 **small-world networks** (Watts & Strogatz, 1998). These networks advantageously combine high
1488 local clustering with short global average path lengths.
- 1489 • **Rationale:** In *CoThinker*, these β -driven connections ensure efficient propagation of diverse
1490 perspectives across cognitive clusters. This shortens the information path length, promotes the
1491 synthesis of varied knowledge, helps distribute the intrinsic cognitive load of the overall task, and
1492 prevents premature convergence.

1493 In essence, the Communication Moderator dynamically cultivates a network with small-world
1494 characteristics. By balancing the formation of strong-tie local clusters for specialized processing with
1495 weak-tie bridges for global integration, it supports both deep, focused collaboration and the broad
1496 synthesis of diverse insights, crucial for effective collective problem-solving.

1497 D.6 SYNTHESIZER MODULE: CONSOLIDATION AND COGNITIVE GROUNDING

1498 The Synthesizer module (Section 4.4) consolidates outputs from all agents ($\{x_i^{(T-1)}\}_{i=1}^M$) and the
1499 final Transactive Memory System state ($\mu^{(T-1)}$) into a single solution for the task D after T_{max}
1500 rounds. It can be implemented as an External Agent (dedicated LLM) or an In-group Agent (team
1501 member) (Lu et al., 2024; Shinn et al., 2023). The design choice for the Synthesizer can vary, with
1502 different cognitive implications drawing from Collaborative Cognitive Load Theory (Kirschner et al.,
1503 2018) and Observational Learning (Bandura & Walters, 1977):

- 1504 1. **External Agent Synthesizer (Observational Learning):** This involves a dedicated LLM instance,
1505 distinct from the collaborating agents, to produce the final output. This agent receives all final
1506 individual perspectives and the collective memory summary.
 - 1507 • *Cognitive Analogy:* This setup mirrors **Observational Learning** (Bandura & Walters, 1977).
1508 The External Synthesizer observes the diverse problem-solving behaviors and refined outputs

1512 of the specialist agents. By analyzing these varied "models" of thought and their collective
 1513 synthesis ($\mu^{(T-1)}$), it can construct a comprehensive solution, potentially integrating insights in
 1514 a novel way without having been part of the iterative load distribution.

1515 **2. In-group Agent Synthesizer (Collaborative Leading/Shared Regulation):** One of the existing
 1516 collaborating agents (e.g., an agent identified as a leader or one with a consistently high-quality
 1517 output, or a randomly chosen one) can be tasked with synthesizing the final solution. This agent
 1518 uses its own understanding, the collective memory $\mu^{(T-1)}$, and the final outputs of its peers. align
 1519 • *Cognitive Analogy*: This aligns with principles from **Collaborative Cognitive Load Theory**
 1520 (**CCLT**) (Kirschner et al., 2018), specifically aspects of shared regulation and distributed
 1521 leadership. The synthesizing agent, having participated in the collaborative process, leverages
 1522 its deep contextual understanding and the established collective working memory ($\mu^{(T-1)}$)
 1523 to guide the final integration. Its synthesis is an act of "collaborative leading" by taking
 1524 responsibility for the final product based on the group's efforts.

1525 **Sample Prompt for an External Agent Synthesizer (Synth):**

1527 Original Task: "[Task Description D]"
 1528 After collaborative thinking, the final individual
 1529 perspectives from $M=[Number of Agents]$ agents are:
 1530 Agent 1: " $[x_1^{(T-1)}]$ "
 1531 ...
 1532 Agent M : " $[x_M^{(T-1)}]$ "
 1533 The final collective understanding synthesized during
 1534 their collaboration is:
 1535 " $[\mu^{(T-1)}]$ "
 1536 Based on all this information, please generate a
 1537 comprehensive, high-quality, and coherent final
 1538 solution to the original task.

1539 This prompt structure ensures the Synthesizer has all the necessary context to perform its role
 1540 effectively.

1541
 1542 **E SUPPLEMENTARY EXPERIMENTAL DATA AND BENCHMARK DETAILS**

1543 **E.1 EXPERIMENTAL CONFIGURATION DETAILS**

1544
 1545 Here we give detailed configuration of experiments from Section 5.1.

1546
 1547 **LLM API Parameters:** For all baseline methods (IO, CoT, SR) and the initial generation round
 1548 ($t = 0$) of multi-agent methods (MAD, DMAD, *CoThinker*), the API temperature was set to "0.25" to
 1549 encourage some diversity. For subsequent iterative rounds ($t > 0$) in *CoThinker*, MAD, and DMAD,
 1550 the temperature was set to "0.0" and "frequency_penalty" to "0.5" to promote focused refinement
 1551 and reduce repetition. Other API parameters (e.g., "top_p", "top_k") were left at their default values.
 1552 Maximum output tokens were set to be large enough for each task, otherwise specified in the task
 1553 description.

1554
 1555 ***CoThinker* Default Configuration:** Unless specified otherwise in ablation studies, *CoThinker* used
 1556 $M = 6$ agents, $T_{max} = 3$ interaction rounds (initial generation + 2 refinement rounds), a reference
 1557 set size $N = 3$ (each agent receives messages from 3 peers), and an exploration rate $\beta = 0.3$.

1558
 1559 All models run with the initial generation temperature set to 0.25 to encourage diverse outputs. In
 1560 multi-agent settings, subsequent rounds use temperature 0.0 and a frequency penalty of 0.5 to reduce
 1561 repetition. By default, multi-agent methods use $M=6$ agents interacting over $T=3$ rounds. For
 1562 *CoThinker*, we set $N=3$ references and exploration parameter $\beta=0.3$.

1563
 1564 **E.2 DETAILED BASELINE METHOD DESCRIPTIONS**

1565 The baseline methods used for comparison from Section 5.1 are implemented as follows:

- **Single Agent (Standard Prompt - IO):** The base LLM is given the task instruction directly, without any specialized prompting techniques, serving as a fundamental measure of its raw capability.
- **Single Agent (CoT):** Chain-of-Thought prompting (Wei et al., 2022) is employed, where the LLM is prompted to "think step by step" or provided with few-shot examples demonstrating a reasoning process before arriving at the final answer.
- **Single Agent (Self-Refine - SR) (Madaan et al., 2023):** This method involves an iterative process ($T = 3$ iterations). The LLM first generates an initial solution. Subsequently, it is prompted to critique its previous output and then to generate an improved version based on that critique.
- **Multi-Agent Debate (MAD) (Liang et al., 2023; Du et al., 2023):** Multiple LLM agents ($M = 6$) initially generate individual solutions. In subsequent iterative rounds ($T = 3$ total generations), each agent receives the solutions from all other agents from the previous round and is prompted to consider these peer solutions, critique them if necessary, and refine its own solution. The final answer is typically derived from the best-performing agent's output after the debate rounds.
- **Diverse Multi-Agent Debate (DMAD) (Liu et al., 2025b):** DMAD extends MAD by promoting diverse reasoning methods from the outset. Each agent is assigned a distinct prompting strategy (e.g., standard IO, Chain-of-Thought, Step-Back Prompting) to generate its initial solution, aiming to break "fixed mental sets." These diverse initial solutions are then shared and refined through iterative debate rounds, similar to MAD.

Summary. (i) *Single Agent (IO)* is a standard mode of prompting without additional techniques. (ii) *Single Agent (CoT)* incorporates Chain-of-Thought prompting to elicit step-by-step reasoning. (iii) *Single Agent (Self-Refine)* uses iterative self-critique and revision processes. (iv) *Multi-Agent Debate (MAD)* employs interactive agent discussion with consensus formation. (v) *Diverse MAD (DMAD)* introduces heterogeneous prompting to avoid fixed mental sets and encourage diverse perspectives.

E.3 BENCHMARK DETAILS

LiveBench (SOTA, comprehensive real-world tasks). LiveBench (White et al., 2025) is a widely used 2025 SOTA benchmark featuring challenging, objectively scorables tasks (e.g., high school math competitions, zebra and word puzzles) that demand high cognitive load for humans. It integrates content from established benchmarks including Big-Bench Hard (Suzgun et al., 2023), AMPS (Hendrycks et al., 2021b), and IFEval (Zhou et al., 2023). Frequent updates minimize test data contamination so results reflect genuine reasoning rather than memorization. Covered domains include:

- *Mathematics:* competitive programming, olympiad-level math, algebraic simplification
- *Reasoning:* logical deduction and spatial reasoning
- *Language:* nuanced understanding and manipulation
- *Instruction Following:* adherence to complex instructions
- *Data Analysis:* structured data manipulation

The tasks are intentionally difficult—even strong models struggle—making LiveBench an ideal proxy for high cognitive load scenarios.

CommonGen-Hard (controlled information interactivity). CommonGen-Hard (Madaan et al., 2023), derived from CommonGen (Lin et al., 2020), systematically stresses element interactivity: the model must generate a coherent multi-sentence paragraph using 3–5 target concepts selected from a large pool (30) of distractors while maintaining narrative coherence and commonsense plausibility. We adopt a 10-dimensional evaluation metric (Li et al., 2018) and employ an LLM-based evaluator (Gemini-1.5-Pro), following the protocol in Ye et al. (2023) to mitigate LLM-as-judge bias. The evaluator uses a detailed rubric with per-dimension Likert (1–10) scoring and aggregates to a total score. The ten rubric dimensions are:

1. **Relevance to Query** (appropriateness and focus; highest weight)
2. **Conciseness** (brevity without loss of essentials)
3. **Clarity & Understandability** (ease of comprehension)
4. **Readability & Fluency** (natural flow; grammar)
5. **Comprehensiveness & Completeness** (covers all prompt aspects)

1620 6. **Demonstrated Knowledge** (accurate commonsense/domain knowledge)
 1621 7. **Logic & Coherence** (internal consistency and structure)
 1622 8. **Originality & Creativity** (novel framing/ideas)
 1623 9. **Engagement & Interest** (compelling responses)
 1624 10. **Insightfulness & Depth** (analytical richness; lowest weight)

1625 This setup provides a controlled environment to validate that our architecture manages the information
 1626 interactivity predicted by CLT to cause performance degradation, while keeping evaluation fair and
 1627 systematic.

1629 E.4 CROSS-MODEL DETAILED RESULTS

1631 This section provides comprehensive cross-model evaluation results referenced in Section 5.4, includ-
 1632 ing detailed performance breakdowns across multiple LLM families and task categories.

1634 Table 12 presents the full cross-model evaluation results across all tested models and evaluation
 1635 scenarios. Table 13 provides detailed task-specific results for the standard setting cross-model
 1636 evaluation, showing CoThinker’s performance across individual reasoning and mathematical tasks.

1638 Model	1639 Method	1639 Avg (SE)	1639 Math (SE)	1639 Reasoning (SE)
1640 Gemini-2.5-Flash	1641 CoThinker	72.8 (2.67)	76.3 (2.57)	69.2 (4.15)
	1641 DMAD	59.7 (3.32)	56.7 (4.48)	62.8 (4.70)
	1641 IO (Baseline)	45.1 (3.40)	59.3 (3.78)	31.0 (4.86)
1643 GPT-4.1-Mini	1644 CoThinker	55.4 (2.92)	40.0 (3.72)	70.8 (3.90)
	1644 DMAD	39.1 (3.21)	34.2 (3.84)	44.0 (4.82)
	1644 IO (Baseline)	37.4 (3.27)	34.0 (3.91)	40.8 (4.82)
1646 Qwen3-32B	1647 CoThinker	22.1 (2.89)	18.9 (3.52)	25.2 (4.22)
	1647 DMAD	11.5 (2.38)	8.8 (3.53)	14.2 (3.25)
	1647 IO (Baseline)	11.7 (2.33)	3.4 (3.53)	20.0 (3.14)
1649 DeepSeek-R1-8B	1650 CoThinker	5.8 (2.28)	2.9 (1.42)	8.8 (3.58)
	1650 DMAD	5.2 (1.71)	3.8 (2.00)	6.5 (2.58)
	1650 IO (Baseline)	2.3 (1.46)	1.9 (1.94)	2.8 (2.00)

1652 Table 12: Complete cross-model evaluation results on LiveBench under constrained setting (8192
 1653 tokens, greedy decoding). Values in parentheses are Bootstrap Standard Errors (SE) in percentage
 1654 points. CoThinker demonstrates consistent and statistically significant improvements across diverse
 1655 model families and capabilities.

1659 Model Configuration	1660 Zebra Puzzle	1660 Spatial	1660 Math Comp	1660 AMPS Hard
1661 GPT-5-Nano (CoThinker)	75.75	88.00	89.13	88.00
1661 GPT-5-Nano (IO)	56.75	80.00	78.26	87.00
1662 Qwen3-30B-A3B (CoThinker)	83.50	84.00	84.78	88.00
1663 Qwen3-30B-A3B (IO)	79.00	76.00	73.91	77.00
1664 GPT-OSS-20B (CoThinker)	55.00	80.00	78.26	82.00
1664 GPT-OSS-20B (IO)	54.00	78.00	71.74	82.00

1666 Table 13: Task-specific performance breakdown across models and tasks. Full CoThinker configu-
 1667 ration consistently outperforms baseline single-agent approach across reasoning and mathematical
 1668 tasks, demonstrating the effectiveness of the complete multi-agent architecture.

1670 The detailed results reveal several key insights about CoThinker’s generalization capabilities:

1671 **Model Family Independence:** CoThinker demonstrates consistent improvements across architec-
 1672 turally diverse families (GPT, Gemini, Qwen, DeepSeek), indicating that the CLT-based approach
 1673 generalizes beyond specific model architectures.

1674
 1675 **Performance Scaling:** Stronger base models (e.g., GPT-5-Nano, Qwen3-30B-A3B) show larger
 1676 absolute gains from CoThinker, suggesting that the framework effectively leverages higher baseline
 1677 reasoning capabilities while providing substantial improvements for weaker models.

1678 **Task-Dependent Benefits:** Mathematical reasoning tasks (math_comp, AMPS_Hard) and logical
 1679 reasoning tasks (zebra_puzzle, spatial) consistently benefit from CoThinker’s collaborative approach,
 1680 validating the cognitive load distribution hypothesis for high-intrinsic-load tasks.

1681 **Configuration Sensitivity:** The comparison between full CoThinker configuration and baseline
 1682 single-agent approach shows that the complete multi-agent architecture (including TMS, Com-
 1683 munication Moderator, and Style Generation) provides meaningful improvements over simplified
 1684 configurations.

1685 E.5 OTHER COMPONENT ABLATION DETAILS

1686 This section provides the complete component ablation results referenced in Section 5.6, showing
 1687 detailed performance breakdowns for other model configurations tested. All configurations use M=6
 1688 agents and N=3 references with greedy decoding.

Task	TMS: ON	TMS: OFF
Math: Math Comp	35.6	7.7

1689 Table 14: TMS ablation for Gemini-1.5-Flash-8B.

Task	TMS: ON	TMS: OFF
Math: Olympiad	47.1	44.2

1700 Table 15: TMS ablation for Gemini-1.5-Flash.

Task	TMS: ON	TMS: OFF
Data Analysis: CTA	58.0	58.0
Instruction Following: Paraphrase	69.8	70.2
Instruction Following: Simplify	68.0	68.9
Instruction Following: Summarize	68.4	66.8
Language: Connections	53.5	52.6
Reasoning: Zebra Puzzle	48.2	39.5

1710 Table 16: TMS ablation for Gemini-1.5-Pro.

1711 **Analysis:** The TMS component shows variable effectiveness across model variants and task types.
 1712 For the weaker Flash-8B model, TMS provides dramatic improvements on mathematical computation
 1713 tasks, suggesting it helps manage cognitive load for complex reasoning. The standard Flash model
 1714 demonstrates modest gains on advanced mathematical problems. The Pro model shows mixed
 1715 results, with notable benefits for logical reasoning (Zebra Puzzle) but minimal impact on instruction-
 1716 following tasks. This pattern indicates that TMS effectiveness depends on both model capability and
 1717 task complexity, with greater benefits for cognitively demanding tasks on less capable models.

1718 E.6 TASK-WISE PERFORMANCE DATA

1719 *This subsection provides comprehensive raw scores for all subtasks across various model families and*
 1720 *prompting methodologies, along with ablation studies investigating sensitivity to key hyperparameters.*

1721 E.7 RAW SUBTASK PERFORMANCE SCORES

1722 The subsequent tables (Table 17 through Table 19) itemize the raw performance scores achieved
 1723 on each subtask. Scores are reported to two decimal places. A hyphen (-) signifies missing or
 1724 non-numeric data. Each table is dedicated to a distinct base model family.

1728

1729

1730

Subtask	IO	CoT	SR	MAD	DMAD	CoThinker
Connections	13.50	18.17	17.33	17.67	17.00	19.33
CTA	54.00	50.00	30.00	48.00	52.00	54.00
Math Comp.	26.09	23.91	21.74	28.26	30.43	26.09
Olympiad	23.82	27.64	23.84	28.25	25.87	29.00
Paraphrase	74.27	72.82	38.42	65.22	66.55	46.02
Simplify	70.33	70.70	62.78	63.88	61.08	70.25
Spatial	34.00	28.00	18.00	34.00	22.00	28.00
Story Gen.	73.08	68.75	62.92	66.75	67.00	65.08
Summarize	69.35	71.27	50.43	58.32	62.62	42.32
Table Join	5.44	4.10	0.00	2.00	1.78	12.02
Table Reformat	80.00	82.00	36.00	38.00	50.00	60.00
Zebra Puzzle	16.00	22.25	17.25	22.75	17.00	25.75

1742

Table 17: Raw scores for each subtask for Gemini-1.5-Flash-8B models across different prompting methods.

1745

1746

1747

Subtask	IO	CoT	SR	MAD	DMAD	CoThinker
Connections	28.17	24.00	22.83	33.17	28.50	33.67
CTA	56.00	56.00	36.00	56.00	54.00	52.00
Math Comp.	41.30	39.13	39.13	41.30	41.30	41.30
Olympiad	32.20	34.37	33.35	34.41	33.27	36.89
Paraphrase	80.70	78.17	52.22	80.58	82.22	72.35
Simplify	75.83	77.68	67.57	72.07	74.40	69.00
Spatial	50.00	50.00	36.00	58.00	52.00	52.00
Story Gen.	76.25	77.50	57.92	60.75	80.75	79.50
Summarize	77.55	75.92	54.05	68.47	74.33	68.97
Table Join	21.64	22.78	8.12	15.00	32.60	31.20
Table Reformat	86.00	80.00	44.00	48.00	44.00	50.00
Zebra Puzzle	28.50	32.00	32.50	34.25	37.50	38.50

1761

1762

1763

1764

1765

1766

1767

Subtask	IO	CoT	SR	MAD	DMAD	CoThinker
Connections	31.17	36.50	35.17	44.67	44.50	46.00
CTA	56.00	58.00	36.00	56.00	60.00	58.00
Math Comp.	47.83	36.96	45.65	54.35	56.52	56.52
Olympiad	51.79	54.77	50.16	59.63	58.46	62.72
Paraphrase	75.37	73.78	34.18	48.50	73.88	65.17
Simplify	74.77	75.72	54.48	55.43	72.88	66.37
Spatial	44.00	48.00	36.00	34.00	38.00	38.00
Story Gen.	69.72	68.05	42.55	56.85	67.30	73.05
Summarize	68.92	67.17	46.23	52.83	69.05	65.72
Table Join	35.98	32.56	16.16	43.82	42.32	44.18
Table Reformat	88.00	88.00	28.00	28.00	86.00	78.00
Zebra Puzzle	39.00	35.75	40.75	41.00	42.25	44.50

1779

Table 19: Raw scores for each subtask for Gemini-1.5-Pro models across different prompting methods.

1782 E.8 SUBTASK DESCRIPTIONS
1783

1784 The evaluation benchmark comprises a diverse array of subtasks, each designed to assess specific
1785 reasoning and generation capabilities of the models. Concise descriptions for each subtask category
1786 are provided below:

1787 **Connections:** Assesses the model’s aptitude for identifying and comprehending relationships (e.g.,
1788 logical, causal, shared attributes) between disparate textual elements or conceptual ideas.
1789 **CTA (Call to Action):** Evaluates the model’s effectiveness in generating or interpreting persuasive or
1790 directive language aimed at eliciting a targeted response or action.
1791 **Math Comp. (Mathematical Computation):** Measures the model’s proficiency in executing mathe-
1792 matical calculations and resolving problems necessitating computational procedures.
1793 **Olympiad:** Challenges the model with highly complex mathematical problems, characteristic of
1794 mathematics Olympiads, which demand profound reasoning and multi-step solution strategies.
1795 **Paraphrase:** Tests the model’s ability to accurately rephrase given text while preserving its original
1796 semantic content, thereby demonstrating linguistic understanding and versatility.
1797 **Simplify:** Assesses the model’s capacity to transform complex textual information into a more readily
1798 understandable format, typically by employing simpler vocabulary and sentence structures without
1799 loss of core meaning.
1800 **Spatial:** Evaluates the model’s spatial reasoning faculties, including its ability to understand and
1801 reason about objects in two or three-dimensional space, their interrelations, positions, and transfor-
1802 mations.
1803 **Story Generation:** Measures the model’s creative ability to produce coherent, engaging, and context-
1804 ually relevant narratives derived from specified prompts or constraints.
1805 **Summarize:** Assesses the model’s proficiency in condensing extended passages of text into succinct
1806 summaries that encapsulate the principal points and essential information.
1807 **Table Join:** Evaluates the model’s comprehension of relational data structures by requiring it to
1808 identify appropriate mechanisms for combining or linking multiple data tables based on common
1809 columns or keys.
1810 **Table Reformat:** Tests the model’s capability to manipulate tabular data by converting a table from
1811 one structural or data representation format to another, adhering to provided instructions.
1812 **Zebra Puzzle:** Assesses the model’s deductive reasoning and constraint satisfaction abilities through
1813 logic puzzles (such as Einstein’s Puzzle) that necessitate deriving a solution from a given set of clues.

1814 E.9 ABLATION STUDY: IMPACT OF REFERENCE SET SIZE (N)
1815

1816 This study investigates the influence of varying the reference set size (hyperparameter N) on model
1817 performance across selected subtasks. N dictates the number of prior examples or “thoughts”
1818 considered by the model during generation. Values of N from 0 (representing a baseline, e.g.,
1819 standard CoT where N/A) to 5 were evaluated using the Gemini-1.5-Flash-8B model. The results are
1820 illustrated in Figure 5.

1821 **Analysis of Figure 5:**
1822

- 1823 • The general trend in performance on these reasoning-intensive (‘olympiad’, ‘spatial’, ‘ze-
1824 bra_puzzle’) and language-based (‘connections’) tasks is examined to determine if it im-
1825 proves, plateaus, or reveals an optimal N value.
- 1826 • Performance at N=0 (baseline) is contrasted with N>0 configurations to ascertain whether
1827 the introduction of a reference set confers a tangible advantage for these specific tasks.
- 1828 • The differential sensitivity of subtasks to variations in N is analyzed, particularly for compu-
1829 tionally demanding tasks like ‘olympiad’ (Math) or ‘zebra_puzzle’ (Reasoning) relative to
1830 ‘connections’ or ‘spatial’.
- 1831 • The investigation seeks to identify if a particular N value (e.g., N=2 or N=3) consistently
1832 yields superior scores or an advantageous performance-cost balance across these subtasks.
- 1833 • Evidence for diminishing returns is sought, where increasing N beyond a certain point might
1834 lead to marginal gains or even performance degradation, potentially due to the introduction
1835 of noise or distracting elements from an overly large reference set.

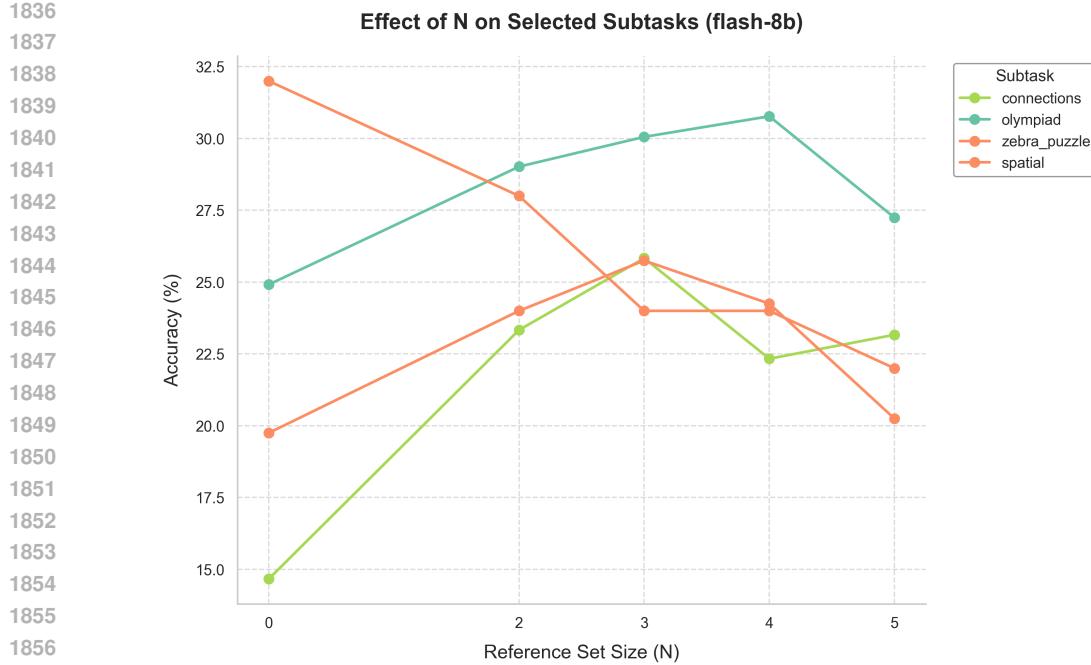


Figure 5: Effect of Reference Set Size (N) on performance for selected subtasks ('connections', 'olympiad', 'zebra_puzzle', 'spatial') using the Gemini-1.5-Flash-8B model. Subtasks are color-coded by their primary category.

Contextual Note: Reasoning and mathematical tasks are often hypothesized to benefit from a moderately sized, diverse reference set. While N=0 or N=1 might provide insufficient context, excessively large N values could introduce irrelevant information.

E.10 ABLATION STUDY: IMPACT OF EXPLORATION RATE (BETA)

This ablation study explores the effect of the exploration rate (hyperparameter Beta) on model performance for selected subtasks, maintaining a fixed reference set size of N=2. Beta influences the diversity of thoughts or solutions generated by the model. The Gemini-1.5-Flash-8B model was employed for this analysis (Figure 6).

Analysis of Figure 6:

- The analysis aims to identify an optimal or effective range for Beta where performance peaks for the selected subtasks, which include data analysis ('tablejoin'), instruction following ('story_generation', 'simplify'), and mathematical computation ('math_comp').
- The impact of extreme Beta values (both very low, indicating minimal exploration, and very high, indicating extensive exploration) on performance is examined for potential suboptimality.
- Differential responses to Beta across subtasks are investigated, for instance, whether creative tasks like 'story_generation' benefit from a different Beta regime compared to more structured tasks such as 'math_comp' or 'tablejoin'.
- The stability of performance across the spectrum of Beta values is assessed, noting any significant fluctuations versus relatively consistent scores within particular ranges.

Contextual Note: A moderate Beta value (e.g., 0.3-0.6 in analogous systems) often represents a balance. Excessively low Beta values might risk premature convergence on suboptimal solutions, while overly high values could lead to an excessively diverse, and potentially lower-quality, set of outputs.

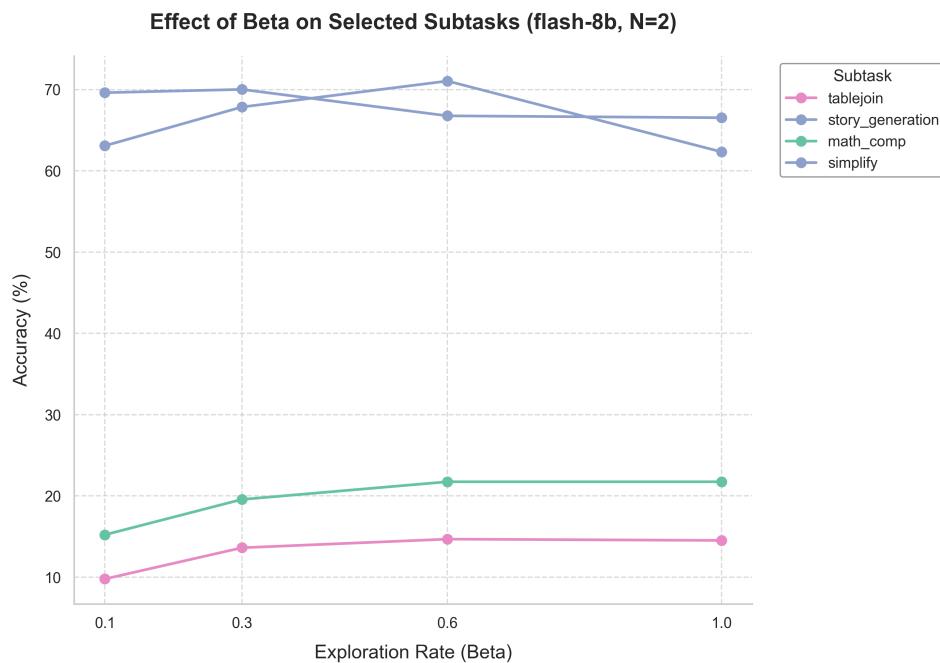


Figure 6: Effect of Exploration Rate (Beta) on performance for selected subtasks ('tablejoin', 'story_generation', 'math_comp', 'simplify') using Gemini-1.5-Flash-8B with N=2. Subtasks are color-coded by their primary category.

E.11 ABLATION STUDY: IMPACT OF NUMBER OF AGENTS (M)

This study assesses the influence of the number of agents (hyperparameter M) on performance across all subtasks, with the reference set size fixed at N=3. M denotes the number of independent reasoning paths or "thinkers" utilized by the model. The Gemini-1.5-Flash-8B model was used for this evaluation (Figure 7).

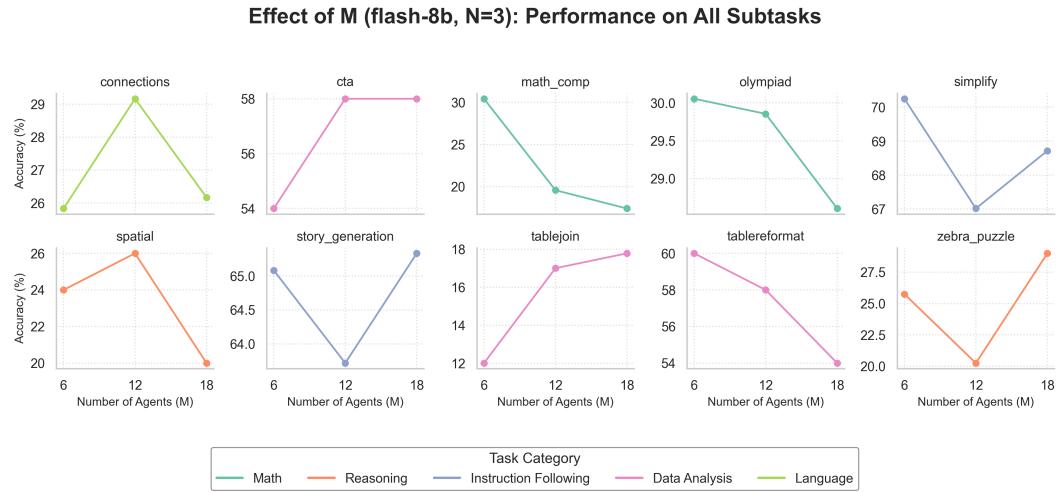


Figure 7: Effect of Number of Agents (M) on performance across all subtasks for Gemini-1.5-Flash-8B with N=3. Each facet corresponds to a subtask, color-coded by its primary category.

Analysis of Figure 7:

1944

- The overall impact of increasing M on performance is analyzed to determine if it generally leads to improvements across most subtasks or if the effects are heterogeneous.

1945

- A cost-benefit perspective is considered, as higher M values, while potentially enhancing performance, also incur increased computational overhead. The study seeks an M value that offers a good trade-off.

1946

- Subtasks that derive particular benefit from a larger number of agents are identified; for example, complex reasoning tasks or those requiring diverse perspectives might exhibit more substantial gains.

1947

- The analysis looks for a saturation point where the benefits of increasing M diminish or where performance might even degrade for some (or all) tasks.

1948

1949

1950

1951

1952

1953

1954 *Contextual Note:* Employing a greater number of agents can enhance the robustness and breadth of exploration. However, an excessive number might not yield significant incremental value or could potentially introduce noise if the aggregation of outputs from multiple agents is not optimally managed.

1955

1956

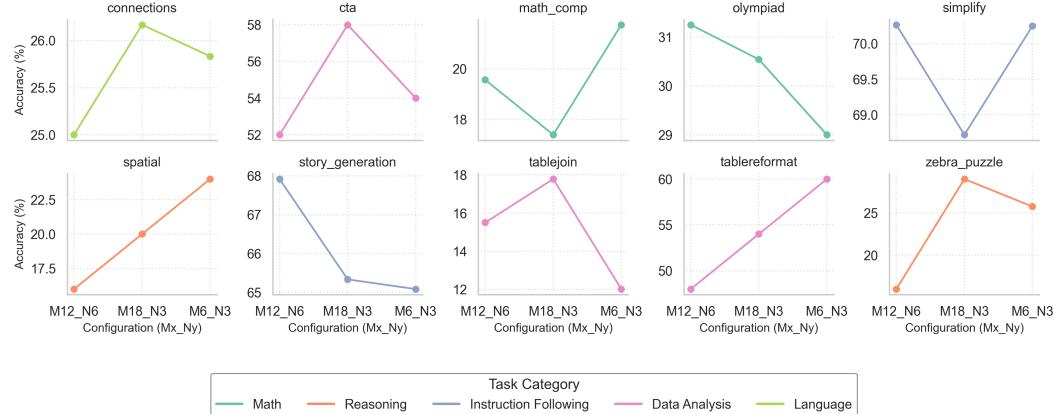
1957

E.12 ABLATION STUDY: PERFORMANCE FOR SPECIFIC M/N

1960 This analysis evaluates performance across three distinct (M, N) configurations for the Gemini-1.5-
1961 Flash-8B model: M6_N3, M12_N6, and M18_N3. These evaluations are conducted under the "With
1962 Style" configuration, with Beta fixed at 0.3 and T (temperature or trials) at 3. Results are presented in
1963 Figure 8.

1964

Subtask Perf. M/N Configs (With Style, B=0.3, T=3, gemini-1.5-flash-8b): Performance on All Subtasks



1981 Figure 8: Subtask performance for specific M/N configurations (M6_N3, M12_N6, M18_N3) using
1982 Gemini-1.5-Flash-8B under the configuration (Beta=0.3, T=3). Faceted by subtask.
1983

Analysis of Figure 8:

1984

- The investigation aims to identify which of the tested (M, N) pairs yields the most favorable performance, either broadly across subtasks or for specific, critical subtasks.

1985

- The trade-off between computational cost and performance gain is considered, as the configurations (M6_N3, M12_N6, M18_N3) entail different computational demands.

1986

- The interaction between M and N is observed by comparing configurations; for instance, whether simultaneous increases in M and N (e.g., M6_N3 to M12_N6) lead to consistent improvements. The M18_N3 configuration provides insight into a different scaling strategy (higher M, moderate N).

1987

- Consistency in the ranking of these (M, N) configurations across different subtasks is examined.

1988

1989

1990

1991

1992

1993

1994

1995

1996

1997 *Contextual Note:* This study assists in identifying potentially effective fixed configurations by exploring varied scaling strategies for the hyperparameters M and N within the "With Style" framework.

1998 F LIMITATIONS AND FUTURE WORK

1999
 2000 Our work successfully demonstrates the utility of Cognitive Load Theory (CLT) as a generative
 2001 framework for designing multi-agent LLM systems. The limitations of this initial study are best
 2002 understood as defining the current boundaries of our methods and charting a course for developing
 2003 more robust tools for this new area of research.

2004
 2005 **Toward Runtime Cognitive Load Measurement and Adaptive Systems.** A central challenge
 2006 is developing better **runtime measurement** capabilities for cognitive states in LLMs. Our current
 2007 proxies (attention entropy and perplexity) require ground-truth answers, making them valuable for
 2008 design-time validation but unsuitable for runtime adaptation. To enable truly adaptive systems that
 2009 detect cognitive overload during inference, three key quantities require measurement:

2010 (1) *Working Memory with Chunking*: Understanding how LLMs organize and chunk information into
 2011 representational units—what constitutes an "element" in their working memory beyond simple token
 2012 counts.

2013 (2) *Dynamic Cognitive Load During Inference*: Tracking how cognitive load transforms through
 2014 reasoning chains, as recent work shows attention patterns evolve with reasoning steps (Wen et al.,
 2015 2025b; Li et al., 2023c). Attention-based runtime monitoring offers potential solutions.

2016 (3) *Metacognitive Sensitivity*: Calibrating uncertainty signals to distinguish processing difficulty from
 2017 knowledge gaps. Recent advances in attention-based uncertainty quantification (Li et al., 2025) and
 2018 RL-based uncertainty training (Damani et al., 2025) provide promising directions.

2019 Once these components are quantified, our validated proxy framework provides the foundation for
 2020 runtime prediction systems that can adaptively decide when to invoke multi-agent collaboration—a
 2021 critical step toward truly intelligent cognitive load management.

2022
 2023 **Developing Universal Proxies for Cognitive Load.** Beyond runtime measurement, we need more
 2024 **universal proxies** for design-time analysis. Our current measures can be influenced by model
 2025 architecture and tokenization. A critical direction is developing standardized "cognitive toolkits" that
 2026 work reliably across diverse model families, potentially including gradient-based sensitivity analysis
 2027 or direct elicitation methods as suggested by reviewers.

2028
 2029 **Boundary Conditions and Task-Adaptive Collaboration.** Our findings help delineate the **boundary**-
 2030 **conditions** under which CLT-based collaboration helps. Benefits are most pronounced for
 2031 high intrinsic cognitive load tasks. Future work should develop principled methods to predict *a*
 2032 *priori* which tasks require collaborative architectures, potentially through cognitive load estimation
 2033 before execution. This connects to the adaptive system vision above—determining not just *when* to
 2034 collaborate during a task, but *which tasks* benefit from collaboration at all.

2035
 2036 **Emergent Collective Dynamics.** While *CoThinker* operationalizes mechanisms for collective
 2037 cognition, the rich internal social dynamics remain underexplored. Applying methods from **com-**
 2038 **putational social science** to analyze interaction patterns—network evolution, transient leadership,
 2039 consensus formation—could reveal deeper insights into artificial collective intelligence and potentially
 2040 inform improved architectures.

2041
 2042 **Human-AI Collaboration.** A particularly exciting direction is extending the framework to include
 2043 human users as specialized agents, creating **human-AI cognitive systems** where the architecture
 2044 actively manages cognitive load for both human and AI participants. This could enable solving
 2045 problems that neither could tackle alone, fostering true hybrid intelligence.

2046
 2047 **Bidirectional Benefits: LLMs as Cognitive Science Research Tools.** Finally, we note the potential
 2048 for **bidirectional knowledge transfer**. While we apply cognitive science to improve LLMs, studying
 2049 LLMs offers more controllable experimental paradigms than human studies. Investigating how
 2050 LLMs chunk information, manage cognitive load, and develop collective intelligence could yield new
 2051 insights about cognitive mechanisms that inform human cognitive science itself.

2052 Addressing these future directions will advance our understanding of how to build truly collaborative
2053 and cognitively capable LLM-based systems while potentially contributing back to cognitive science
2054 through computational modeling.

2055

2056

2057

2058

2059

2060

2061

2062

2063

2064

2065

2066

2067

2068

2069

2070

2071

2072

2073

2074

2075

2076

2077

2078

2079

2080

2081

2082

2083

2084

2085

2086

2087

2088

2089

2090

2091

2092

2093

2094

2095

2096

2097

2098

2099

2100

2101

2102

2103

2104

2105