United Minds or Isolated Agents? Exploring Coordination of LLMs under Cognitive Load Theory

Anonymous Author(s)
Affiliation
Address
email

Abstract

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

1 Introduction

The increasing prevalence and capability of Large Language Models (LLMs) are transforming diverse domains, moving beyond basic text generation towards complex reasoning and problem-solving applications [Chang et al., 2024, Zhao et al., 2024, Li et al., 2024a]. Aligning these powerful models with human intent and fostering effective thinking pattern is paramount for unlocking their full potential [Shen et al., 2023]. In-Context Learning (ICL) is increasingly employed for alignment, offering adaptation via prompts without parameter updates [Brown et al., 2020]. In this work, we adopt a broad definition of ICL, referring to the general strategy of guiding an LLM's behavior by providing any contextual information relevant to the task to perform the task [Lampinen et al., 2024]. Compared to traditional finetuning [Song et al., 2024, Lee et al., 2023], evidence suggests both methods often operate through similar mechanisms—primarily modulating the model's thinking style rather than altering core knowledge [Lin et al., 2024, Zhao et al., 2025, Yang et al., 2024]; ICL's parameter-free nature, and adaptability make it a widely adopted paradigm for this purpose.

While ICL offers flexibility, it suffers from a notable performance ceiling when applied to multi-faceted tasks requiring integration of diverse information sources [He et al., 2024, Li et al., 2023b, Kirk et al., 2023]. In such scenarios, LLM agents frequently exhibit degeneration of thought, lack of diversity, or inability to follow multiple requirements [Liang et al., 2023, Huang et al., 2023, Kamoi et al., 2024, Lu et al., 2024] when using ICL. Despite increasing empirical studies on ICL's

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

limitations, the root causes remain under-explored. Concurrently, recent efforts to overcome the ceiling via agent-based solutions have yielded limited success, often relying on heuristics without cognitive grounding [Liu et al., 2023, Zhang et al., 2024c].

To address the first challenge—the lack of theoretical understanding behind performance ceiling—we turn to cognitive science for explanatory insight. Similar patterns of performance degradation have long been studied in cognitive science, where complex tasks involving high element interactivity often induce [Sweller, 2011, 2003]. According to Cognitive Load Theory (CLT), cognitive overload happens when working memory capacity is exceeded [Baddeley et al., 1986b]. Recent work suggests LLMs also exhibit bounded working memory with human-like failure modes under overload [Zhang et al., 2024b, Gong et al., 2024]. These shared characteristics allow us to draw an analogy that explains the observed performance degradation in LLM agents: *The performance ceiling observed in LLM agents arises when their effective cognitive load capacity is exceeded, closely mirroring the theoretical limits described by CLT.*

Building on this analogical reasoning above—that the performance ceiling observed when applying In-Context Learning (ICL) to complex tasks stems from cognitive overload—we present *CoThinker*, a multi-agent ICL architecture that directly operationalizes insights from CLT to enhance the effectiveness of ICL and improve reasoning capacity through structured cooperation among LLM agents. Specifically, *CoThinker* translates the concept of collective working memory [Kirschner et al., 2018] into a practical architecture. Just as human groups distribute cognitive demands through division of labor and shared memory structures [Wilson et al., 2004, Dunbar, 1998, Tomasello, 2009], *CoThinker* employs specialized agents for parallel thinking and coordinates their outputs via a shared memory mechanism. This collaborative architecture enables the LLM agents to offload and manage high element interactivity, thereby mitigating the cognitive overload experienced by individual agents. To demonstrate the effectiveness of leveraging CLT in this manner, we test *CoThinker* on a range of complex general problem-solving tasks and specifically fabricated high cognitive load scenarios. In sum, this paper makes the following key contributions:

- First, we are the first to explain the performance ceiling of using ICL in LLM agents by drawing a strong analogy to Cognitive Load Theory, suggesting that these limitations stem from exceeding the LLM's effective cognitive load capacity.
- Second, based on these theoretical insights, we design and introduce *CoThinker*, a novel multi-agent ICL architecture. *CoThinker* operationalizes CLT principles, through agent specialization, transactive memory, and communication moderator to mitigate cognitive overload and enhance complex cooperation.
- Third, we empirically validate *CoThinker* on complex tasks, demonstrating its ability to surpass existing multi-agent baselines. Furthermore, our analysis uncovers characteristic interaction patterns among agents, providing insights into the emergence of collective cognition within the architecture.

2 Related Work

2.1 Multi-Agent LLM Collaboration

The development of LLMs has catalyzed significant research into multi-agent systems (MAS) where LLMs function as collaborative agents, aiming to tackle more complex problems than single agents can alone [Guo et al., 2024, Wang et al., 2024a, Qian et al., 2025]. Current approaches explore various interaction structures including multi-agent debate, where agents exchange and critique ideas [Liang et al., 2023, Lu et al., 2024, Wang et al., 2024b, Du et al., 2023], iterative reflection mechanisms, enabling agents to self-correct [Shinn et al., 2023, Madaan et al., 2023, Yao et al., 2023]. Role-playing and functional specialization are also prominent, assigning distinct tasks or personas to different agents to divide labor, particularly in complex, multifaceted domains [Li et al., 2023a, Qian et al., 2023, Hong et al., 2023]. Architecturally, research investigates optimal communication topologies to enhance information flow [Li et al., 2024b], the dynamic formation and adaptation of agent networks [Liu et al., 2023, Wu et al., 2023], diversity of mental set [Liu et al., 2025b], and hierarchical structures for coordination [Zhang et al., 2024a]. However, while these systems demonstrate advancing capabilities, their designs often draw from intuition or focus on communication efficiencies, with less explicit grounding in cognitive theories that explain effective collaboration and the management of processing limitations [Pan et al., 2025]. Specifically, the systematic integration

of Cognitive Load Theory (CLT) [Sweller, 2011] remains largely underexplored in the design of LLM MAS. Our work, *CoThinker*, directly addresses this gap by operationalizing CLT to mitigate cognitive overload in LLMs and enhance collective problem-solving.

2.2 LLM for Human Simulation

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

3 Cognitive Foundations for Enhanced LLM Performance

This section establishes the theoretical basis for our approach by drawing parallels between human cognitive limitations and observed performance ceilings in LLMs. We begin (Section 3.1) by discussing analogous constraints in working memory between humans and LLMs, a concept central to Cognitive Load Theory (CLT). Building on this, we then (Section 3.2) use CLT to interpret LLM performance degradation under complex task demands. Subsequently (Section 3.3), we examine how humans overcome individual cognitive limitations by naturally forming collective cognitive systems, and finally, we posit that these principles can inform the design of a more capable LLM architecture.

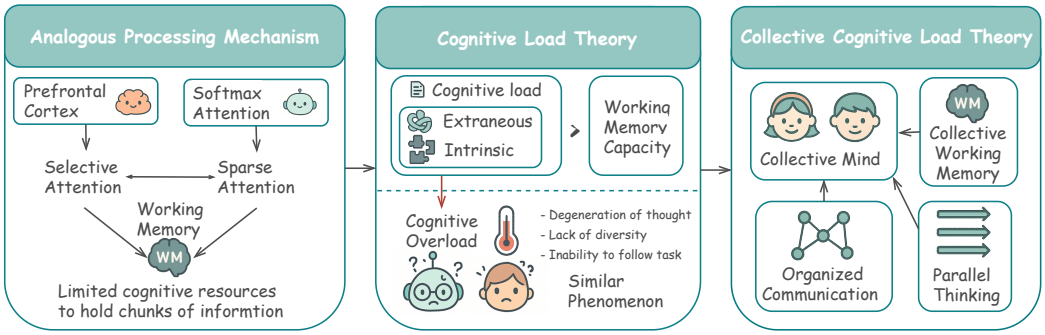


Figure 1: Analogical reasoning on how to mirror Cognitive load in human to LLM Agent to explain the performance ceiling observed when applying In-Context Learning (ICL) to LLM Agents for complex tasks, and use Cognitive Load Theory (CLT) to resolve it.

3.1 Working Memory Analogies

Human cognition relies fundamentally on working memory, a capacity-limited cognitive system associated with the prefrontal cortex, essential for temporarily holding and manipulating information during complex cognitive tasks such as reasoning and learning [Baddeley et al., 1986a, Cowan, 2010]. Human working memory can only hold a limited number of information chunks simultaneously, typically around 4 to 7 [Miller et al., 1956]. This system employs selective attention to filter and prioritize information [Roussy et al., 2021]. LLMs exhibit intriguing functional parallels; their core attention mechanisms perform a form of sparse, selective focus on input data [Vaswani et al., 2017]. Recent studies have begun to characterize a functional "working memory" in LLMs, identifying capacity limits and failure modes under high informational demands that echo human working memory phenomena [Zhang et al., 2024b, Gong et al., 2024]. Thus, a key analogy emerges: both humans and LLMs operate with limited cognitive resources for the concurrent processing of

information, providing a shared foundation for understanding their processing constraints. This analogy sets the stage for applying cognitive theories developed for human reasoning to interpret performance limits in LLMs (See details in Appendix).

3.2 Cognitive Load and Performance Limits

The finite nature of working memory is central to CLT [Sweller et al., 1998, Sweller, 2011]. CLT distinguishes between *intrinsic load*, determined by the inherent complexity and element interactivity of a task, and *extraneous load*, which can arise from how a task or its accompanying instructions are presented. When the combined load exceeds working memory capacity, *cognitive overload* ensues in humans [Paas et al., 2003, Sweller, 2011]. The provided guidance, meant to help, can paradoxically hinder performance if it contributes to exceeding cognitive capacity. LLM agents demonstrate analogous performance degradation when LLM agents are tasked with complex problems and guided by In-Context Learning (ICL). This often causes agents to fail at tasks they are capable of solving. For instance, tasks requiring extensive multi-step reasoning or the integration of numerous, potentially conflicting, constraints via ICL can lead to degeneration of thought, lack of diversity, or inability to follow multiple requirements [Liang et al., 2023, Huang et al., 2023, Kamoi et al., 2024, Lu et al., 2024] (further illustrated in Appendix). This often causes agents to fail at tasks they, in principle, are capable of solving. Drawing upon the working memory analogies and these observed patterns, we contend that such performance ceilings when applying ICL in LLMs can be understood as a manifestation of cognitive overload, where total demands surpass their effective processing capacity. To identify ways to alleviate this overload, we next examine how humans naturally overcome similar limitations through collective cognition.

3.3 Human Collective Intelligence

To surmount individual cognitive limitations, humans exhibit a capacity for collaborative problem-solving, leading to the emergence of a *collective intelligence* or *collective mind* that is more powerful than the sum of its individual constituents [Woolley et al., 2010, Malone et al., 2010, Shteynberg et al., 2023]. This is not simply an aggregation of independent efforts but results from sophisticated social-cognitive abilities, including shared intentionality, theory of mind, and nuanced communication for establishing common ground [Tomasello et al., 2005, Frith and Frith, 2005]. Such collective entities effectively expand cognitive resources by distributing processing. Key aspects include the formation of a *collective working memory*, often through Transactive Memory Systems where knowledge and responsibilities are shared [Wegner, 1987, Kirschner et al., 2018] and individuals have meta-knowledge about who knows what [Hollingshead, 2001] so that they can rely on each other for information sharing and retrieval [Hollingshead and Brandon, 2003]; the engagement in *parallel thinking* through a division of cognitive labor, which reduces the intrinsic load on each individual [Dunbar, 2003]; and the use of *organized communication* to integrate diverse information and maintain a shared attentional focus [Hutchins, 1995]. These spontaneously formed group structures allow humans to manage complexities that would overwhelm an individual, demonstrating a natural solution to cognitive overload.

Inspired by these human collective cognitive strategies and human-LLM cognitive similarity discussed above, the subsequent section introduces ***CoThinker***, a multi-agent ICL architecture designed to operationalize these principles to overcome LLM performance ceilings whe using ICL.

4 CoThinker

CoThinker is a multi-agent ICL architecture designed to enhance collaborative problem-solving by systematically managing cognitive load. Simply aggregating outputs from LLM agents often proves insufficient for complex tasks, as naive collaboration can introduce significant transactional costs—the cognitive effort required to coordinate, communicate, and integrate—without a corresponding increase in solution quality [Pan et al., 2025]. As Cognitive Load Theory (CLT) suggests, these transactional costs can quickly lead to extraneous cognitive overload, negating the benefits of parallel thinking [Kirschner et al., 2009, 2018]. To overcome these challenges within the ICL paradigm, *CoThinker* operationalizes the principles of human collective intelligence discussed in Section 3, aiming to create a "collective mind" that distributes cognitive load. We leverage the insights from CLT to design an architecture that mirrors how human groups effectively solve complex problems.

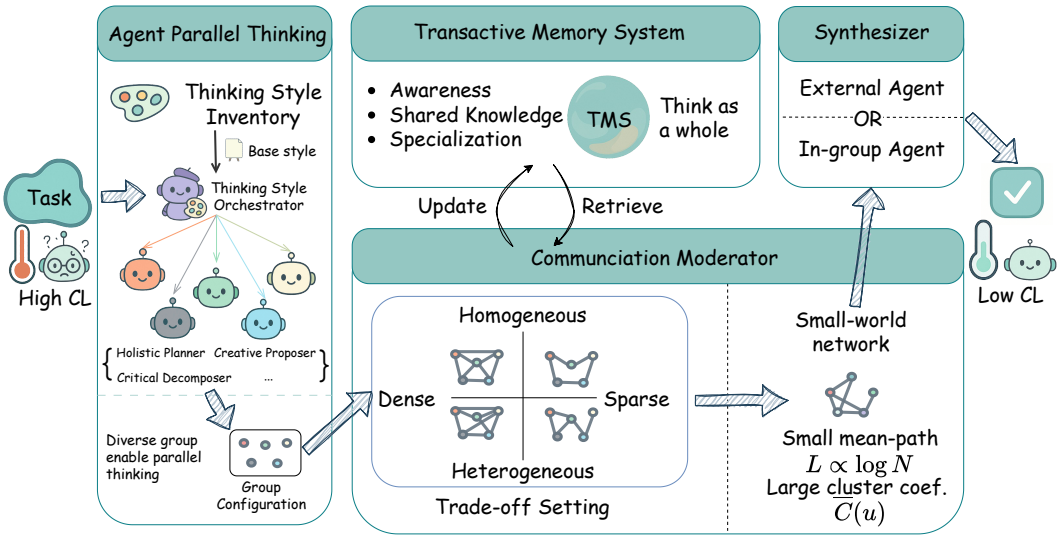


Figure 2: The *CoThinker* Architecture. A high Cognitive Load (CL) task is initially processed by diverse agents via Agent Parallel Thinking. The Transactive Memory System (TMS) facilitates shared understanding by updating and allowing retrieval of collective knowledge. The Communication Moderator manages inter-agent information flow, leveraging a trade-off to form a cognitive small-world network, which then feeds into the Synthesizer to produce a 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 cognitive load from redundant processing. The Communication Moderator balances intrinsic and extraneous loads by structuring information exchange. Finally, the Synthesizer integrates refined collective insights. Let $\mathcal{A} = \{A_1, \ldots, A_M\}$ 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 impose extraneous cognitive load from role adherence, *CoThinker* uses an adaptive approach. A Thinking Style Orchestrator generates a task-specific style ϕ_i for each agent A_i based on a general base thinking style inventory ψ [Sternberg, 1997] and the task D:

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

This yields diverse thinking styles $\{\phi_i\}_{i=1}^{M}$, employed in subsequent stages. Further details on the prompting strategy for style generation and thinking style inventory are in the Appendix.

4.2 Transactive Memory System (TMS)

Human groups effectively manage complex information by developing Transactive Memory Systems (TMS), which involve a shared understanding of who knows what, how to access information held by others, and a collective agreement on the information itself [Wegner, 1987, Hollingshead, 2001]. This distributed cognitive system allows individuals to specialize and rely on others, reducing individual cognitive load and enhancing group problem-solving [Lewis, 2003]. To emulate these benefits and foster a *collective working memory* in *CoThinker*, we implement a structured mechanism for maintaining and accessing shared knowledge. At each round t, an evolving representation of the

group's collective knowledge, denoted $\mu^{(t)}$, is updated based on contributions from all agents:

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

This aims to enhance shared awareness and efficient integration of distributed knowledge. The specific prompt-based emulation of TMS components is detailed in the Appendix.

4.3 Communication Moderator

Effective inter-agent communication is crucial, yet it incurs transactional costs—the cognitive effort for message processing and integration—which can impose extraneous cognitive load, a key concern in Collaborative Cognitive Load Theory [Kirschner et al., 2009, 2018]. To mitigate these costs, the Communication Moderator structures information exchange by selecting N reference messages $\mathcal{P}_i^{(t-1)}$ for each agent A_i. This process navigates the critical trade-offs between **Network Density vs. Sparsity** (high exposure and cost vs. low cost and potential information loss) and **Information Homogeneity vs. Heterogeneity**. The latter involves balancing the ease of integrating cognitively similar inputs (low extraneous load but risk of echo chambers [Runkel, 1956]) against the benefits of diverse perspectives for distributing intrinsic load [Aral and Van Alstyne, 2011]).

Communication Topology and Algorithm: The selection of references defines a directed communication 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 agent A_v receives a message from agent A_u generated in round $t-1$. Motivated by how small-world networks efficiently balance local clustering with global connectivity [Watts and Strogatz, 1998], our moderator employs the following algorithm to construct this graph:

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

Resulting Network Properties and Cognitive Balance: This rewiring process fosters dynamic communication networks $G^{(t-1)}$ with small-world properties. Such networks exhibit high local clustering (facilitating efficient refinement of similar ideas, reducing extraneous load locally) and short average path lengths (enabling rapid global propagation of diverse insights, aiding intrinsic load distribution). This structure offers a principled balance between focused collaboration and broad information access, managing cognitive load more effectively than purely random or regular lattice networks. Further details are in the Appendix.

4.4 Synthesizer

The Synthesizer consolidates information into a final solution after T_{max} rounds. It can be an External Agent (dedicated LLM) or an In-group Agent (team member) [Lu et al., 2024, Shinn et al., 2023]. This draws from Collaborative Cognitive Load Theory [Kirschner et al., 2018] and Observational Learning [Bandura and Walters, 1977] (See details in Appendix)

***CoThinker* Process Flow**

The process for task D with M agents over T rounds:

Initialization:

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

Iterative Refinement: For each agent A_i and round t:

$$\mathcal{P}_i^{(t)} = \text{SelectRefs}\big(\{x_k^{(t)}\}_{k \in \mathcal{A} \setminus \{A_i\}}, x_i^{(t)}, N, \beta\big) \tag{4}$$

$$x_i^{(t+1)} = \text{Agent}\big(D, \phi_i, \mu^{(t)}, x_i^{(t)}, \mathcal{P}_i^{(t)}\big) \tag{5}$$

$$\mu^{(t+1)} = \text{UpdateMem}\big(\mu^{(t)}, \{x_i^{(t+1)}\}_{i=1}^{M}\big) \tag{6}$$

Final Synthesis:

$$y_{\text{final}} = \text{Synth}\big(\{x_i^{(T-1)}\}_{i=1}^{M}, \mu^{(T-1)}, D\big) \tag{7}$$

5 Experiments and Results

This section details our experimental methodology and presents the empirical evaluation of *CoThinker*. We first outline the experimental setup, including the base LLMs, benchmarks, and baselines. We then present the main results on LiveBench and CommonGen-Hard, followed by ablation studies and a discussion of our findings through the lens of Cognitive Load Theory (CLT).

5.1 Experimental Setup

Models and Configuration. We use three Gemini models [Team et al., 2024] with varying capacities: `gemini-1.5-flash-8b` (lightweight), `gemini-1.5-flash` (mid-tier), and `gemini-1.5-pro` (high-capacity). All models run with the initial generation temperature set to 0.25 to encourage diverse outputs. In multi-agent settings, subsequent rounds use temperature 0.0 and a frequency penalty of 0.5 to reduce repetition. By default, multi-agent methods use M=6 agents interacting over T=3 rounds. For *CoThinker*, we set N=3 references and exploration parameter β=0.3.

Evaluation Benchmarks. We evaluate on two challenging benchmarks: (1) **LiveBench** [White et al., 2025], a recent diverse suite drawing from Big-Bench Hard [Suzgun et al., 2023], AMPS [Hendrycks et al., 2021], and IFEval [Zhou et al., 2023], covering domains such as mathematics, coding, language, instruction following, and data analysis; and (2) **CommonGen-Hard** [Madaan et al., 2023], a cognitively demanding variant of CommonGen [Lin et al., 2020], which evaluates multi-sentence generation under high element interactivity. We adopt a 10-dimensional metric for CommonGen-Hard evaluation [Li et al., 2018]. See full details in the Appendix.

Baselines. We compare *CoThinker* with both single-agent and multi-agent approaches. *(i) Single Agent (IO)* is a standard mode of prompting. *(ii) Single Agent (CoT)* incorporates Chain-of-Thought prompting [Wei et al., 2022]. *(iii) Single Agent (Self-Refine)* uses iterative self-critique and revision [Madaan et al., 2023]. *(iv) Multi-Agent Debate (MAD):* employs interactive agent discussion with consensus formation [Du et al., 2023, Liang et al., 2023]. *(v) Diverse MAD (DMAD):* introduces heterogeneous prompting to avoid fixed mental sets [Liu et al., 2025b]. See details in the Appendix.

5.2 Main Results on LiveBench

Table (1) presents the performance of *CoThinker* and baseline methods across the LiveBench suit for `gemini-1.5-flash-8b`, `gemini-1.5-flash`, and `gemini-1.5-pro`. Scores are reported as relative improvements over the respective `gemini-8b-flash`'s IO (Standard Prompt) baseline. An average score is calculated as the arithmetic mean of these relative scores across the main LiveBench categories (Math, Reasoning, Instruction, Data, Language).

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

Table 1: LiveBench[White et al., 2025] performance with all scores normalized by `gemini-1.5-flash-8b-io` baseline. The abbreviations corresponded to Math, Data Analysis, Reasoning Language, and Instruction Following

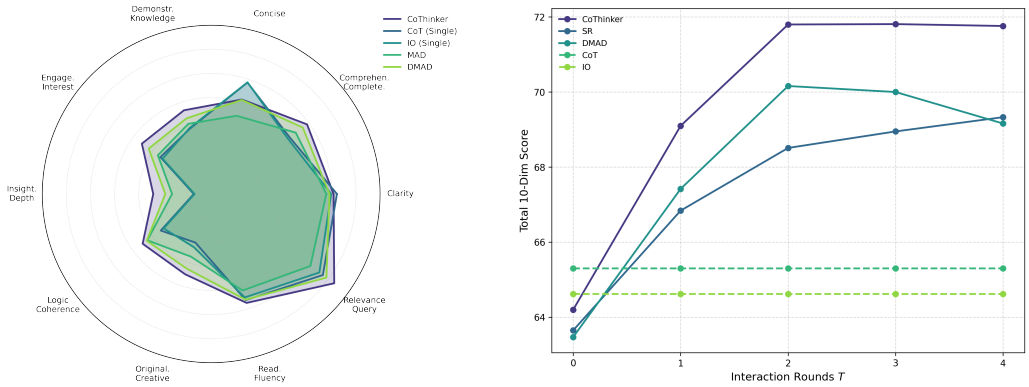
Analysis of LiveBench Results. *CoThinker* consistently achieves strong average performance across all base model families, with particularly notable gains in complex categories like Data Analysis, Reasoning, and often Math, but low performance on Instruction Following. We posit this performance pattern reflects two distinct task categories: those with high intrinsic cognitive load and those with low intrinsic load. The former, characterized by tasks like Data Analysis and Reasoning, demonstrates a clear scaling in baseline performance as model capability increases (e.g., from

`gemini-1.5-flash-8b` to `gemini-1.5-pro`), indicating that greater raw cognitive power inherently improves outcomes. For these high-load tasks, *CoThinker* excels by effectively decomposing complex problems and orchestrating collaborative agent contributions, therefore, splitting the intrinsic load to enhance performance.

Conversely, tasks with low intrinsic load, such as instruction following (Instr.), show minimal or inconsistent performance gains when moving from weaker to stronger base models; for instance, the `gemini-1.5-pro` IO baseline on Instruction Following does not substantially outperform that of `gemini-1.5-flash-8b`. This suggests the primary bottleneck is not cognitive load. In such scenarios, the added communication overhead inherent in *CoThinker* can outweigh the benefits of collaboration. For tasks demanding straightforward adherence rather than sophisticated reasoning, this introduced more extraneous cognitive load, explaining why *CoThinker* may not show an advantage or can even underperform on these low-load, execution-focused tasks.

5.3 Main Results on CommonGen-Hard

In CommonGen-Hard, which emphasizes managing high element interactivity, *CoThinker* demonstrates notable performance improvements. Figure 3 presents these results, with Figure 3a illustrating its balanced strengths across evaluation dimensions and Figure 3b showing performance trends over interaction rounds.



(a) (b)

Figure 3: *CoThinker* performance on CommonGen-Hard using `gemini-1.5-flash`. (a) The radar plot illustrates a multi-dimensional performance profile, where *CoThinker* typically shows well-rounded and superior strengths compared to baselines. (b) The rounds plot depicts the total score trend across interaction rounds (T), often indicating an optimal number of rounds before performance plateaus or declines.

Analysis of CommonGen-Hard Results.
CoThinker demonstrates strong overall performance on CommonGen-Hard (Figure 3), effectively managing its high element interactivity. The multi-dimensional profile (Figure 3a) typically shows *CoThinker* excelling in key areas like coherence and concept integration, albeit with occasional trade-offs in aspects such as conciseness. This aligns with Cognitive Load Theory (CLT); the high intrinsic load of the task is managed by *CoThinker*'s distributed processing and transactive memory. Notably, its performance trajectory versus interaction rounds (Figure 3b) highlights a key advantage: *CoThinker* achieves sustained constructive refinement over several rounds, effectively managing cognitive load. This contrasts with the multi-agent baseline where performance degrades due to rapidly accumulating extraneous load from inefficient coordination or information overload. *CoThinker*'s architecture appears more adept at balancing these loads, delaying diminishing returns.

5.4 Ablation Studies on LiveBench Subsets

Ablation studies were conducted on `gemini-1.5-flash-8b` using averaged scores from selected LiveBench subtask categories (Math, Reasoning, Data Analysis, and Instruction). These studies investigated the impact of *CoThinker*'s reference set size (N), exploration rate (β), and the number of agents (M). Unless otherwise specified, default parameters were $T = 3$. For N ablation,

$M = 6, \beta = 0.3$. For β ablation, $N = 2, M = 6$. For M ablation, $N = 3, \beta = 0.3$. All scores are normalized by the I/O baseline performance for each subtask before category averaging.

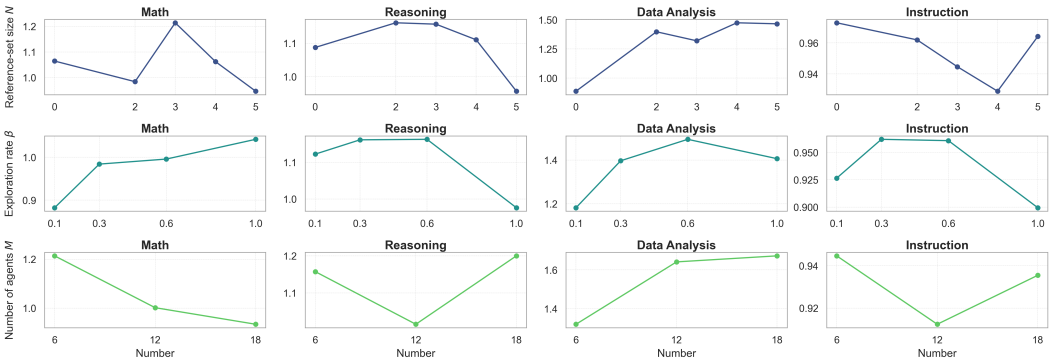


Figure 4: Ablation studies on *CoThinker* parameters (N, β, M) using `gemini-1.5-flash-8b`. Performance is shown for four LiveBench task categories (Math, Reasoning, Data Analysis, Instruction), normalized by IO baseline performance (1.0). **Top row**: Effect of Reference Set Size (N), varying $N \in \{0, 2, 3, 4, 5\}$ with $M = 6, \beta = 0.3, T = 3$. **Middle row**: Effect of Exploration Rate (β), varying $\beta \in \{0.1, 0.3, 0.6, 1.0\}$ with $N = 2, M = 6, T = 3$. **Bottom row**: Effect of Number of Agents (M), varying $M \in \{6, 12, 18\}$ with $N = 3, \beta = 0.3, T = 3$. Optimal parameter settings are task-dependent, indicating varying sensitivities to peer input diversity and information overload.

Analysis of Ablation Studies.
Figure 4 demonstrates *CoThinker*'s hyperparameter sensitivity, offering insights into cognitive load management as theorized in Section 3. The reference set size (N, top row) directly impacts extraneous cognitive load. An optimal N (e.g., 2-3) balances diverse peer input against cognitive overload, respecting LLM working memory limits. Too few references limit collaboration; too many overwhelm. The exploration rate (β, middle row) governs the trade-off between exploiting similar ideas (low β, lower extraneous load for integration) and exploring diverse ones (high β, high extraneous load). Task-dependent optima, like higher β for Reasoning, reflect this balance, managed by the Communication Moderator's cognitive small-world network. The number of agents (M, bottom row) shows that while more agents can distribute intrinsic load, increasing M also elevates transactional (extraneous) load from coordination. Non-monotonic performance indicates that beyond a point, these transactional costs negate the benefits of parallelism, aligning with CLT's predictions for group overload. These findings affirm that *CoThinker*'s parameters are crucial for managing cognitive load, enabling the emergence of an effective "collective mind" by mitigating overload.

6 Conclusion

This work addresses the performance limitations of LLMs on complex tasks, particularly when employing In-Context Learning (ICL), by drawing an analogy to Cognitive Load Theory (CLT). We posit that observed performance ceilings arise from exceeding an LLM's effective cognitive load capacity when processing intricate task details and extensive in-context guidance. We introduced *CoThinker*, a multi-agent architecture that operationalizes CLT principles. Through agent specialization, a transactive memory system, and moderated communication, *CoThinker* mitigates overload and enhances collaborative problem-solving, especially for tasks that challenge single agents using ICL. Empirical evaluations on benchmarks like LiveBench and CommonGen-Hard demonstrated *CoThinker*'s superior performance over existing baselines on high-load tasks. Analyses validated *CoThinker*'s effective management of cognitive load, fostering a more robust "collective mind." By grounding multi-agent LLM design in CLT, this research offers a principled path towards overcoming performance bottlenecks encountered when applying ICL to demanding problems, contributing to more powerful collaborative AI systems through the lens of cognitive science.

References

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966, 2024.

Sinan Aral and Marshall Van Alstyne. The diversity-bandwidth trade-off. *American journal of sociology*, 117(1):90–171, 2011.

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

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

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

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

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

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

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

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

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

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

Robin IM Dunbar. The social brain hypothesis. *Evolutionary Anthropology: Issues, News, and Reviews: Issues, News, and Reviews*, 6(5):178–190, 1998.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. arXiv preprint arXiv:2309.00267, 2023.

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

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

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

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

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

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

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

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

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

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

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

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

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

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri 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.

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

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

Fred Paas, Alexander Renkl, and John Sweller. Cognitive load theory and instructional design: Recent developments. *Educational psychologist*, 38(1):1–4, 2003.

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

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

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

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

Megan Roussy, Diego Mendoza-Halliday, and Julio C Martinez-Trujillo. Neural substrates of visual perception and working memory: two sides of the same coin or two different coins? *Frontiers in neural circuits*, 15:764177, 2021.

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

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

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

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

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998, 2024.

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

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

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

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

John Sweller, Jeroen JG Van Merrienboer, and Fred GWC Paas. Cognitive architecture and instructional design. *Educational psychology review*, 10:251–296, 1998.

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.

Michael Tomasello. *Why we cooperate*. MIT press, 2009.

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

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

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

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

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

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

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

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

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

David Sloan Wilson, John J Timmel, and Ralph R Miller. Cognitive cooperation: when the going gets tough, think as a group. *Human Nature*, 15:225–250, 2004.

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

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

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

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

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

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

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

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

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

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

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

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

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

Appendix

A Cognitive Foundations: Elaborations

A.1 Human Working Memory and Attentional Control

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.

A.2 Using Cognitive Load Theory to Explain Phenomena in LLM Performance

Cognitive Load Theory (CLT) offers a valuable lens to interpret puzzling LLM performance issues, positing that LLMs, like humans, have finite processing capacity. Exceeding this capacity leads to performance degradation. This section concisely analyzes several such cases through CLT.

1. **Degradation of Thought in Self-Reflection:** Liang et al. [2023] found LLMs may rigidly stick to incorrect initial answers during self-reflection, failing to correct meaningfully.
   - *CLT Explanation:* Self-reflection (holding problem, solution, critique, and revision process concurrently) is highly demanding. If initial analysis already consumes most capacity, the LLM may lack resources for genuine re-evaluation, defaulting to superficial agreement due to cognitive overload.
2. **Performance Degradation with More In-Context Examples (Many-Shot ICL):** Agarwal et al. [2024] noted LLM performance can degrade with more in-context examples, especially on complex tasks (e.g., MATH).
   - *CLT Explanation:* While few examples scaffold, excessive examples increase total cognitive load beyond capacity. The LLM struggles to synthesize all information, akin to CLT's "redundancy effect" where too much information, even relevant, overwhelms working memory.
3. **Performance Degradation Despite Increasing "Confidence" (NLL Trends):** Agarwal et al. [2024] also found that performance degradation in many-shot ICL wasn't always explained by NLL (confidence) trends; NLL could improve as performance worsened.
   - *CLT Explanation:* Under cognitive overload, LLMs (like humans) may resort to heuristics. Overwhelmed by many examples, an LLM might latch onto superficial patterns, yielding outputs that are stylistically plausible (good NLL) but incorrect. This "overconfidence" in a flawed heuristic stems from an inability to allocate resources for deeper reasoning.
4. **Reduced Diversity after RLHF for Instruction Following:** Kirk et al. [2023] and others observed that RLHF, while improving instruction following, can reduce output diversity.
   - *CLT Explanation:* Intense RLHF training on narrow preferences imposes a high "germane load" for conformance. To manage this, and the extraneous load of deviating from rewarded paths, the model may operate in a constrained output space, reducing the cognitive effort of exploring diverse (potentially unrewarded) responses. The "cost" of diversity becomes too high.

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

B CoThinker Architecture: Implementation and Prompting

B.1 Prompt Architecture for Agent Parallel Thinking

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

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

1. Style Orchestration (Orch **function):** The Thinking Style Orchestrator (itself an LLM) is provided with the overall task description D and a Thinking Style Inventory. This inventory consists of base thinking styles derived from Sternberg's theory, encompassing dimensions such as Functions (Legislative, Executive, Judicial), Forms (e.g., Monarchic, Hierarchic), Levels (Global, Local), Scope (Internal, External), and Leanings (Liberal, Conservative). The Orchestrator's objective is to generate a diverse yet task-relevant set of M specific thinking styles $\{\phi_1, \ldots, \phi_M\}$, one for each agent A_i. For each agent, the Orchestrator takes one or a combination of Sternberg's dimensions as a base style ψ_i and adapts it to the given task D. The Orchestrator is guided to ensure the resulting set of styles $\{\phi_i\}$ promotes varied perspectives on the problem, reflecting the value of different styles for different task facets.

An example prompt for the Orchestrator, given a base combination from Sternberg (e.g., ψ_i = "Legislative-Global style"):

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

Generate a concise (1-2 sentences) task-specific adaptation
of this thinking style profile that would be most beneficial
for an agent contributing to this primary task. The agent
should focus its reasoning and output according to this
adapted style.
Task-Specific Style for an agent:
```

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

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

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

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

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

B.2 Prompt Architecture for Transactive Memory System (TMS) Emulation

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

CoThinker's emulation of TMS centers on the generation and presentation of the collective memory state, $\mu^{(t)}$, at each round t. This is not merely an aggregation of past messages but a structured synthesis designed to reflect key TMS components. Specifically, an auxiliary LLM agent (the "TMS Manager") is tasked with populating a predefined "TMS Template" based on all agent outputs $\{x_j^{(t-1)}\}_{j=1}^{M}$ from the previous round and the existing memory state $\mu^{(t-1)}$, to produce the updated $\mu^{(t)}$. This template explicitly guides the TMS Manager to synthesize information reflecting:

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

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

```
[Agent's assigned thinking style: {Style_phi_i}]
[Overall Task: {Task_D}]

Collective Summary from Previous Round (reflecting shared understanding mu^(t)):
"{Text of mu^(t) generated by the TMS Manager using the TMS Template}"
```

```
Your Previous Output (x_i^(t-1)):
"{Text of x_i^(t-1)}"

Reference Outputs from Peers (P_i^(t-1)):
Reference 1 (from Agent A_k): "{Text of x_k^(t-1)}"
Reference 2 (from Agent A_l): "{Text of x_l^(t-1)}"
...

Based on all the above, and keeping your thinking style in mind,
provide your refined thoughts/contribution for the current round:
```

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

B.3 Communication Moderator: Cultivating an Efficient Network via Strong and Weak Ties

The Communication Moderator in *CoThinker* (Section 4.3) strategically structures inter-agent communication by implicitly leveraging principles from social and complex network theories. This design fosters a network optimized for managing cognitive load and enhancing collective intelligence.

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

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

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

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

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

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

B.4 Synthesizer Module: Consolidation and Cognitive Grounding

The Synthesizer module (Section 4.4) consolidates outputs from all agents ($\{x_i^{(T-1)}\}_{i=1}^{M}$) and the final Transactive Memory System state ($\mu^{(T-1)}$) into a single solution for the task D. The design choice for the Synthesizer can vary, with different cognitive implications:

1. **External Agent Synthesizer (Observational Learning):** This involves a dedicated LLM instance, distinct from the collaborating agents, to produce the final output. This agent receives all final individual perspectives and the collective memory summary.
   - *Cognitive Analogy:* This setup mirrors **Observational Learning** [Bandura and Walters, 1977]. The External Synthesizer observes the diverse problem-solving behaviors and refined outputs of the specialist agents. By analyzing these varied "models" of thought and their collective synthesis ($\mu^{(T-1)}$), it can construct a comprehensive solution, potentially integrating insights in a novel way without having been part of the iterative load distribution.
2. **In-group Agent Synthesizer (Collaborative Leading/Shared Regulation):** One of the existing collaborating agents (e.g., an agent identified as a leader or one with a consistently high-quality output, or a randomly chosen one) can be tasked with synthesizing the final solution. This agent uses its own understanding, the collective memory $\mu^{(T-1)}$, and the final outputs of its peers. align
   - *Cognitive Analogy:* This aligns with principles from **Collaborative Cognitive Load Theory (CCLT)** [Kirschner et al., 2018], specifically aspects of shared regulation and distributed leadership. The synthesizing agent, having participated in the collaborative process, leverages its deep contextual understanding and the established collective working memory ($\mu^{(T-1)}$) to guide the final integration. Its synthesis is an act of "collaborative leading" by taking responsibility for the final product based on the group's efforts.

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

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

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

C Experimental Setup: In-Depth Information

C.1 Detailed Benchmark Descriptions

LiveBench [White et al., 2025] LiveBench serves as a dynamic and robust benchmark for evaluating LLM capabilities, characterized by its frequent updates (monthly) to minimize test data contamination and its focus on objectively scorable, challenging tasks. It draws from established hard benchmarks like Big-Bench Hard and AMPS, as well as introducing novel problems. The tasks span a broad range of domains, including:

- *Mathematics:* Encompassing competitive programming problems, olympiad-level mathematics, and algebraic simplification.
- *Reasoning:* Covering logical deduction and spatial reasoning.
- *Language:* Focusing on nuanced understanding and manipulation.
- *Instruction Following:* Testing adherence to complex instructions
- *Data Analysis:* Requiring structured data manipulation

CommonGen-Hard [Madaan et al., 2023] CommonGen-Hard, an extension of the CommonGen dataset [Lin et al., 2020], is specifically designed to impose high cognitive load by increasing element interactivity. The core task is to generate a coherent, multi-sentence paragraph incorporating a small set of 3-5 target concepts. The difficulty is amplified by including a large number (approximately 30) of irrelevant distractor concepts from which the model must select and use only the targets, while maintaining narrative coherence and commonsense plausibility. Given its generative nature, evaluation employs an LLM-based evaluator (`gemini-1.5-pro`) guided by a detailed rubric assessing ten dimensions. These dimensions are: (1) **Relevance to Query** (appropriateness and focus, highest weight); (2) **Conciseness** (brevity without losing essential content); (3) **Clarity & Understandability** (ease of comprehension); (4) **Readability & Fluency** (natural language flow, grammatical correctness); (5) **Comprehensiveness & Completeness** (addressing all prompt aspects); (6) **Demonstrated Knowledge** (accurate commonsense or domain knowledge); (7) **Logic & Coherence** (internal consistency and logical structure); (8) **Originality & Creativity** (novelty in ideas or framing); (9) **Engagement & Interest** (compelling nature of the response); (10) **Insightfulness & Depth** (analytical richness beyond surface content, lowest weight). Each dimension is scored (e.g., 1-10), and an aggregated total score is used. This setup directly tests the model's ability to manage high element interactivity and filter relevant information, key aspects related to cognitive load.

C.2 Detailed Baseline Method Descriptions

The baseline methods used for comparison 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.

C.3 General Implementation Details

Experiments were conducted using Python and Google's Generative AI SDK. **LLM API Parameters:** For all baseline methods (IO, CoT, SR) and the initial generation round ($t = 0$) of multi-agent methods (MAD, DMAD, *CoThinker*), the API temperature was set to "0.25" to encourage some diversity. For subsequent iterative rounds ($t > 0$) in *CoThinker*, MAD, and DMAD, the temperature was set to "0.0" and "frequency_penalty" to "0.5" to promote focused refinement and reduce repetition. Other API parameters (e.g., "top_p", "top_k") were left at their default values. Maximum output tokens were set appropriately for each task.

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

D Detailed Experimental Results and Ablation Studies

This appendix provides supplementary experimental results, including comprehensive raw scores for all subtasks across various model families and prompting methodologies. Furthermore, it details ablation studies conducted to investigate the sensitivity of model performance to key hyperparameters.

D.1 Raw Subtask Performance Scores

The subsequent tables (Table 2 through Table 4) itemize the raw performance scores achieved on each subtask. Scores are reported to two decimal places. A hyphen (-) signifies missing or non-numeric data. Each table is dedicated to a distinct base model family.

Table 2: Raw scores for each subtask for `gemini-1.5-flash-8b` models across different prompting methods.

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

Table 3: Raw scores for each subtask for `gemini-1.5-flash` models across different prompting methods.

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

D.2 Subtask Descriptions

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

Connections: Assesses the model's aptitude for identifying and comprehending relationships (e.g., logical, causal, shared attributes) between disparate textual elements or conceptual ideas.
CTA (Call to Action): Evaluates the model's effectiveness in generating or interpreting persuasive or directive language aimed at eliciting a targeted response or action.

Table 4: Raw scores for each subtask for `gemini-1.5-pro` models across different prompting methods.

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

Math Comp. (Mathematical Computation): Measures the model's proficiency in executing mathematical calculations and resolving problems necessitating computational procedures.
Olympiad: Challenges the model with highly complex mathematical problems, characteristic of mathematics Olympiads, which demand profound reasoning and multi-step solution strategies.
Paraphrase: Tests the model's ability to accurately rephrase given text while preserving its original semantic content, thereby demonstrating linguistic understanding and versatility.
Simplify: Assesses the model's capacity to transform complex textual information into a more readily understandable format, typically by employing simpler vocabulary and sentence structures without loss of core meaning.
Spatial: Evaluates the model's spatial reasoning faculties, including its ability to understand and reason about objects in two or three-dimensional space, their interrelations, positions, and transformations.
Story Generation: Measures the model's creative ability to produce coherent, engaging, and contextually relevant narratives derived from specified prompts or constraints.
Summarize: Assesses the model's proficiency in condensing extended passages of text into succinct summaries that encapsulate the principal points and essential information.
Table Join: Evaluates the model's comprehension of relational data structures by requiring it to identify appropriate mechanisms for combining or linking multiple data tables based on common columns or keys.
Table Reformat: Tests the model's capability to manipulate tabular data by converting a table from one structural or data representation format to another, adhering to provided instructions.
Zebra Puzzle: Assesses the model's deductive reasoning and constraint satisfaction abilities through logic puzzles (such as Einstein's Puzzle) that necessitate deriving a solution from a given set of clues.

D.3 Ablation Study: Impact of Reference Set Size (N)

This study investigates the influence of varying the reference set size (hyperparameter N) on model performance across selected subtasks. N dictates the number of prior examples or "thoughts" considered by the model during generation. Values of N from 0 (representing a baseline, e.g., standard CoT where N/A) to 5 were evaluated using the `gemini-1.5-flash-8b` model. The results are illustrated in Figure 5.

Analysis of Figure 5:

- The general trend in performance on these reasoning-intensive ('olympiad', 'spatial', 'zebra_puzzle') and language-based ('connections') tasks is examined to determine if it improves, plateaus, or reveals an optimal N value.
- Performance at N=0 (baseline) is contrasted with N>0 configurations to ascertain whether the introduction of a reference set confers a tangible advantage for these specific tasks.

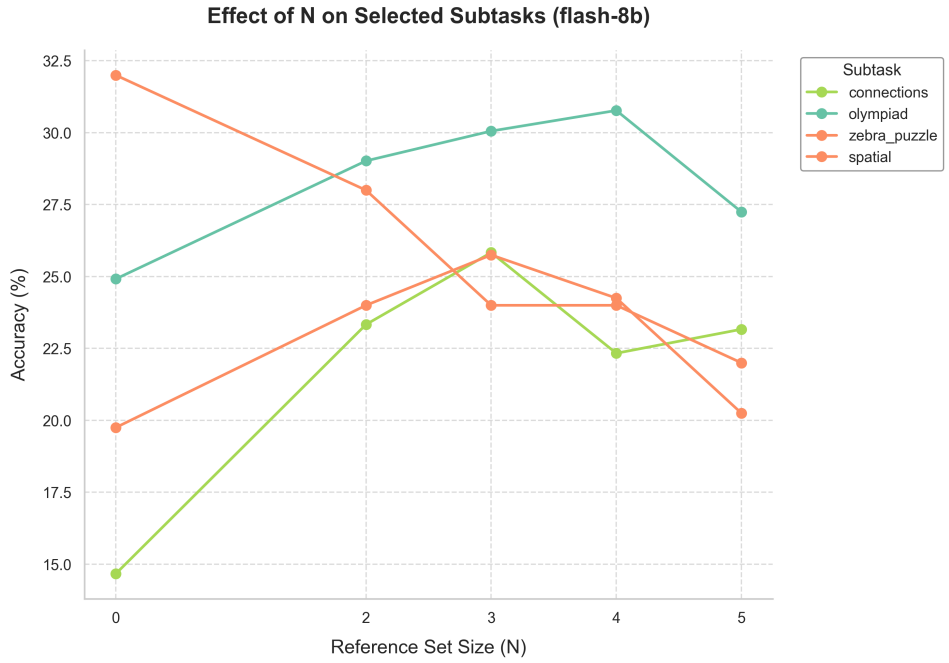


Figure 5: Effect of Reference Set Size (N) on performance for selected subtasks ('connections', 'olympiad', 'zebra_puzzle', 'spatial') using the `gemini-1.5-flash-8b` model. Subtasks are color-coded by their primary category.

- The differential sensitivity of subtasks to variations in N is analyzed, particularly for computationally demanding tasks like 'olympiad' (Math) or 'zebra_puzzle' (Reasoning) relative to 'connections' or 'spatial'.
- The investigation seeks to identify if a particular N value (e.g., N=2 or N=3) consistently yields superior scores or an advantageous performance-cost balance across these subtasks.
- Evidence for diminishing returns is sought, where increasing N beyond a certain point might lead to marginal gains or even performance degradation, potentially due to the introduction of noise or distracting elements from an overly large reference set.

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.

D.4 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'.

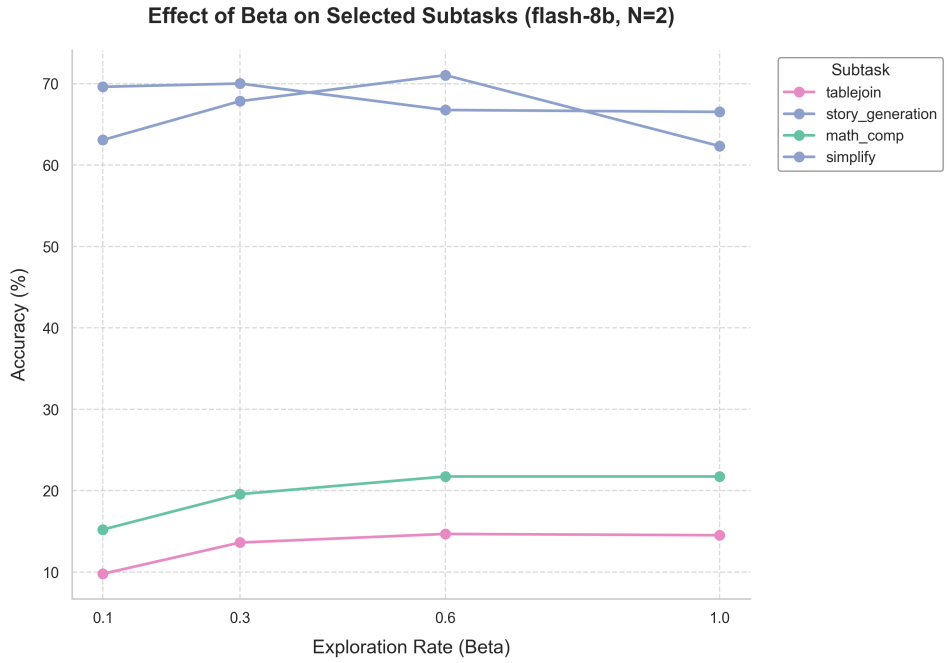


Figure 6: Effect of Exploration Rate (Beta) on performance for selected subtasks ('tablejoin', 'story_generation', 'math_comp', 'simplify') using `gemini-1.5-flash-8b` with N=2. Subtasks are color-coded by their primary category.

- 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.

D.5 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).

Analysis of Figure 7:

- 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.
- 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.
- 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.
- 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.

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.

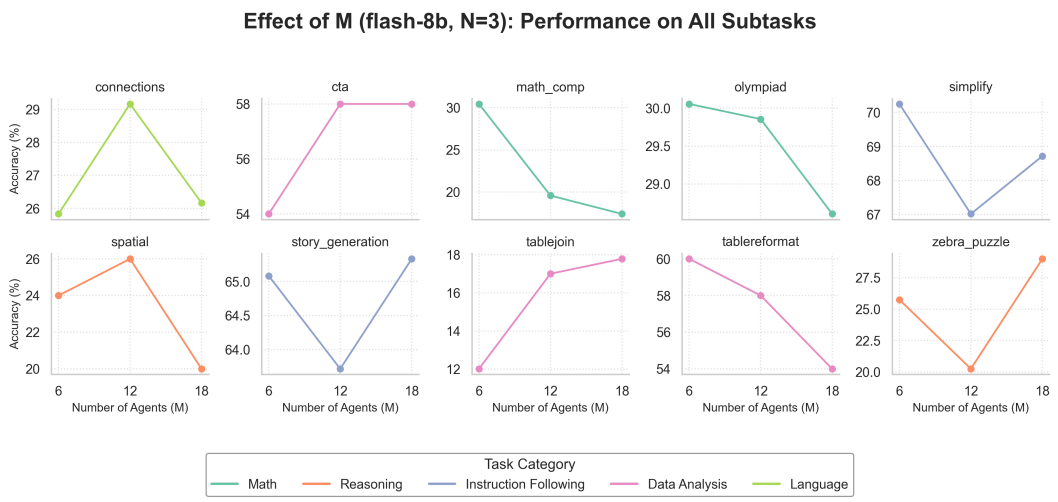


Figure 7: Effect of Number of Agents (M) on performance across all subtasks for `gemini-1.5-flash-8b` with N=3. Each facet corresponds to a subtask, color-coded by its primary category.

D.6 Ablation Study: Performance for Specific M/N

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

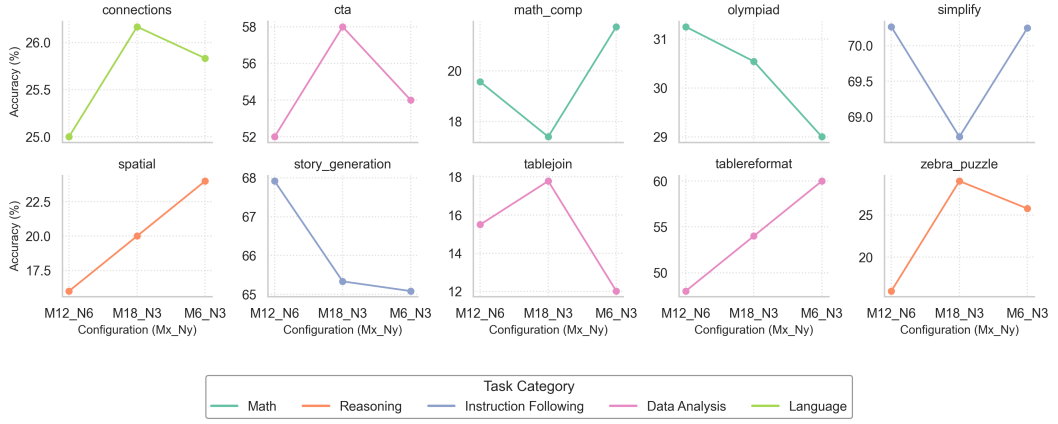


Figure 8: Subtask performance for specific M/N configurations (M6_N3, M12_N6, M18_N3) using `gemini-1.5-flash-8b` under the configuration (Beta=0.3, T=3). Faceted by subtask.

Analysis of Figure 8:

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

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

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.

E Limitations and Future Work

While *CoThinker* demonstrates promising results in managing cognitive load and enhancing collaborative LLM performance, this work has several limitations that also point towards avenues for future research.

Limitations include the scope of LLM evaluation, which primarily utilized models from the Gemini family. The generalizability of specific performance benefits and optimal hyperparameter settings across a wider range of LLM architectures requires further exploration. Additionally, while we argue that *CoThinker* manages transactional costs associated with multi-agent collaboration, a more fine-grained quantitative analysis of these costs versus the gains in solution quality would offer a more complete efficiency profile. The "thinking styles" currently rely on an LLM orchestrator and base styles; the true emergent specialization and their direct impact on distributing intrinsic load warrant deeper investigation.

Future Work could explore several promising directions. Developing **adaptive *CoThinker* architectures** that dynamically adjust parameters (number of agents, communication topology) based on real-time task assessment is a key area. **Deeper integration of CLT principles**, such as explicitly modeling and minimizing extraneous load from prompt design or fostering germane load via sophisticated scaffolding, could further enhance performance. Creating methods for **explainability of collective cognition** within *CoThinker*—tracing information flow, identifying critical contributions, and characterizing shared understanding evolution—would improve transparency. Extending the framework for **human-AI collaboration**, incorporating human users as specialized agents, could lead to powerful human-LLM group cognition. Finally, the prospect of such fused intelligence necessitates proactive examination of its **societal implications**, including equity, potential for misuse, accountability, and ethical considerations, demanding robust frameworks for responsible development and governance. Addressing these limitations and pursuing these future directions will further advance our understanding of how to build truly collaborative and cognitively capable LLM-based systems.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer:[Yes]

   Justification: The abstract and introduction clearly state the main contributions: (1) explaining LLM performance ceilings via an analogy to CLT, (2) introducing *CoThinker* as a CLT-operationalizing multi-agent architecture, and (3) empirically validating *CoThinker*. These claims are reflected in the theoretical discussions (Section 3, 4) and experimental results (Section 5).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have include a section talking about our limitation and future work in the appendix E

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper's theoretical contribution is primarily an analogical framework (CLT applied to LLMs) and a conceptual architecture (*CoThinker*) rather than formal mathematical theorems or proofs. The justification for the architecture's design is rooted in established cognitive science principles (Section 3).

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5.1 and section C.3 in appendix details the base LLMs, benchmarks, baselines, and all configuration parameters (temperature, agent count, rounds, *CoThinker* parameters N, β). The *CoThinker* architecture and process flow are described in Section 4. Further details on detailed prompts will be included in the Supplementary Material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: At the time of submission, the code for *CoThinker* and specific experimental scripts are not publicly released. However, the paper provides detailed descriptions of the architecture (Section 4) and experimental setup (Section 5.1) to facilitate conceptual replication. The datasets used (LiveBench, CommonGen-Hard) are publicly available.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: ection 5.1 specifies the base LLMs, temperature settings, agent count ($M = 6$), interaction rounds ($T = 3$), and *CoThinker*-specific hyperparameters ($N = 3, \beta = 0.3$) for the main experiments. Ablation studies (Section 5.4) explore variations of these. Details on data (benchmarks used) are also provided, with further specifics referenced to the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The current version of the paper reports point estimates for performance on benchmarks. Error bars or statistical significance tests are not included, primarily due to the deterministic nature of the current experimental setup with fixed temperatures (after initial generation) and the focus on demonstrating architectural efficacy across diverse tasks rather than fine-grained statistical variations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The paper specifies the LLMs used (Gemini family via API access) but does not detail exact execution times per task, as these can vary based on API load and are less directly controlled. Specific compute hardware and memory on the user's side are minimal as computation is offloaded. Token usage, which is a key cost factor, will be detailed in the supplementary material/Appendix for transparency regarding resource consumption.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research focuses on developing a multi-agent LLM architecture for improved problem-solving and does not involve human subjects, direct data collection from individuals, or applications with immediate high-risk ethical concerns. We have reviewed the NeurIPS Code of Ethics and believe our work conforms to it.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Justification: Section E (Future Work subsection) discusses the broader societal impacts. It outlines the potential for "Human-LLM Fused Group Cognition" to dramatically enhance problem-solving for societal grand challenges (positive impact). It also explicitly addresses potential negative impacts and ethical considerations, including equity of access, new forms of manipulation, amplified biases, accountability in distributed decision-making, and the ethics of deeply integrating AI into human deliberative processes, calling for responsible development and governance.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper introduces an architecture (*CoThinker*) that utilizes existing pre-trained LLMs (Gemini family). It does not release new pre-trained models or large-scale datasets that would pose a high risk for misuse requiring specific safeguards beyond those implemented by the original LLM providers.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring

that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The base LLMs used (Gemini family) are products of Google and are used via their API, respecting their terms of service. The benchmarks LiveBench [White et al., 2025] and CommonGen-Hard [Madaan et al., 2023] are publicly available datasets and are cited appropriately (Section 5.1). Specific licenses for these benchmarks could be detailed further in an Appendix.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper introduces a new architecture (*CoThinker*) but does not release new datasets, code, or pre-trained models as standalone assets at this time. The architecture itself is documented within the paper (Section 4).

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The research presented does not involve crowdsourcing experiments or direct research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The research presented does not involve crowdsourcing experiments or direct research with human subjects, so IRB approval was not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are a central component of the core methods investigated; specifically, the *CoThinker* architecture is an LLM-based multi-agent system where LLMs (Gemini family) function as the agents (Sections 4, 5.1). The research studies these LLM agents within our novel framework. The conceptualization of this framework, the CLT analogy, and the research design itself are human-derived contributions, with LLMs being the subject and operational components of the proposed methodology.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.