AgentCDM: Enhancing Multi-Agent Collaborative Decision-Making via ACH-Inspired Structured Reasoning

Xuyang Zhao, Shiwan Zhao, Hualong Yu, Liting Zhang, Qicheng Li*

College of Computer Science, Nankai University, Tianjin, China
{xychao,2120240710,2111190}@mail.nankai.edu.cn,
zhaosw@gmail.com,
liqicheng@nankai.edu.cn

Abstract

Multi-agent systems (MAS) powered by large language models (LLMs) hold significant promise for solving complex decision-making tasks. However, the core process of collaborative decision-making (CDM) within these systems remains underexplored. Existing approaches often rely on either "dictatorial" strategies that are vulnerable to the cognitive biases of a single agent, or "voting-based" methods that fail to fully harness collective intelligence. To address these limitations, we propose **AgentCDM**, a structured framework for enhancing collaborative decision-making in LLM-based multi-agent systems. Drawing inspiration from the Analysis of Competing Hypotheses (ACH) in cognitive science, AgentCDM introduces a structured reasoning paradigm that systematically mitigates cognitive biases and shifts decision-making from passive answer selection to active hypothesis evaluation and construction. To internalize this reasoning process, we develop a two-stage training paradigm: the first stage uses explicit ACH-inspired scaffolding to guide the model through structured reasoning, while the second stage progressively removes this scaffolding to encourage autonomous generalization. Experiments on multiple benchmark datasets demonstrate that AgentCDM achieves state-of-the-art performance and exhibits strong generalization, validating its effectiveness in improving the quality and robustness of collaborative decisions in MAS.

Introduction

Large Language Models (LLMs) have achieved remarkable success across a wide range of natural language processing tasks, owing to their strong capabilities in language understanding and reasoning (Achiam et al. 2023; Guo et al. 2025; Grattafiori et al. 2024; Lei et al. 2025; Fu et al. 2025). However, despite their impressive performance, individual LLMs still exhibit several fundamental limitations, such as potential security vulnerabilities (Wolf et al. 2023), content hallucination (Min et al. 2023), and poor handling of complex, multi-step reasoning tasks (Hadi et al. 2023). These limitations have motivated increasing interest in developing multi-agent systems (MAS) built on top of LLMs, aiming to leverage agent collaboration to overcome the deficiencies of a single model.

*Corresponding author.

In LLM-based MAS, individual agents are typically assigned specialized roles (Li et al. 2023), and through interaction and collaboration, they can collectively solve complex problems that are challenging for a single agent to address alone. This paradigm has demonstrated promising potential in various domains, including code generation (Huang et al. 2023), web browsing (Chen et al. 2024), scientific exploration (Lu et al. 2024), and tool use (Shen et al. 2024). Over the past year, the field has evolved rapidly—from static, hand-crafted frameworks to more dynamic architectures in which agent roles and behaviors are adaptive and context-aware. This evolution suggests a clear trajectory toward higher levels of automation and generalization in multi-agent collaboration.

Despite this progress, most existing research on LLM-based MAS has focused primarily on designing interaction protocols—such as communication, coordination, or planning—while largely overlooking the core process of collaborative decision-making (CDM). As the component that ultimately determines the system's output, CDM is critical to the quality and robustness of MAS. Yet prevailing decision mechanisms are often simplistic, typically falling into one of two categories: "dictatorial" or "voting-based". In dictatorial schemes, a single agent serves as the final decision-maker, rendering the outcome highly vulnerable to that agent's internal limitations, such as role confusion, instruction repetition, or infinite loops (Li et al. 2023). In voting-based methods, while multiple agents contribute, the aggregation process is typically shallow, failing to synthesize partial truths or resolve contradictions.

At the heart of these limitations lies a deeper problem: **cognitive bias**. Biases such as anchoring, confirmation bias, and subjective validation affect both individual agents and collective decision-making processes. In dictatorial CDM, the final output is shaped by the subjective inclinations of a single agent. In voting-based CDM, aggregated outputs often retain unchallenged or inconsistent reasoning patterns. These biases degrade the system's ability to reason thoroughly and reliably, especially in high-stakes or ambiguous tasks. Thus, addressing cognitive bias is not only desirable but necessary to advance the effectiveness of CDM in LLM-based MAS.

In human decision-making, structured analytical frameworks are widely used to mitigate cognitive bias. One

such framework is the *Analysis of Competing Hypotheses* (ACH) (Heuer 1999), a systematic method developed in cognitive science to help analysts evaluate multiple competing explanations based on evidence, promoting disconfirmation and deliberative reasoning. Inspired by ACH, we incorporate a structured reasoning strategy into our decision-making agent, guiding it to evaluate alternative hypotheses and synthesize a superior answer through evidence-driven analysis. A conceptual comparison of this ACH-based strategy against existing methods is provided in Figure 1.

To internalize this structured reasoning process within the model, we introduce **AgentCDM**, a novel two-stage training framework. In the first stage, agents are trained with explicit ACH-inspired scaffolding that demonstrates how to conduct structured, bias-resistant reasoning. In the second stage, this scaffolding is gradually removed, encouraging agents to generalize the learned reasoning strategies and apply them autonomously in novel contexts. This design enables the agent to go beyond answer selection—toward the construction of higher-quality, collectively informed decisions.

Our contributions are summarized as follows:

- We introduce a structured reasoning strategy inspired by cognitive science (ACH) to systematically mitigate cognitive bias in LLM-based collaborative decision-making.
- We propose **AgentCDM**, a two-stage training paradigm that enables decision-making agents in MAS to internalize and generalize ACH-inspired reasoning processes.
- We conduct extensive experiments on multiple benchmark datasets, demonstrating that AgentCDM achieves state-of-the-art performance and exhibits strong generalization across domains.

Related Work

LLM-based Multi-Agent Systems

With the advancement of LLMs, multi-agent systems (MAS) leveraging LLMs have become a significant approach to augmenting task-processing capabilities. Frameworks such as CAMEL (Li et al. 2023), MetaGPT (Hong et al. 2023), and AutoGen (Wu et al. 2024) enable agents to interact and coordinate through role-playing, predefined workflows, or multi-agent conversations. SWIFTSAGE (Lin et al. 2023) further draws on dual-process cognition to enhance efficiency and robustness.

While extensive attention has been paid to agent interaction protocols, there is relatively less focus on the aggregation methods for integrating agent outputs into cohesive final decisions.

Collaborative Decision-Making in MAS

Collaborative decision-making (CDM) in MAS aims to achieve collectively superior outcomes through agent cooperation (Bose, Reina, and Marshall 2017; King and Cowlishaw 2007). Existing LLM-based MAS decision-making broadly falls into two categories: dictatorial and voting-based methods.

Dictatorial approaches appoint a predefined "decision-maker" (e.g., leader, critic, judge) to synthesize inputs and render final judgments (Hao et al. 2025; Li et al. 2023; Liang et al. 2023). However, these methods heavily depend on the reasoning capacity of the central agent, with limited research on systematically improving its robustness and interpretability, especially under conflicting inputs. This vulnerability is compounded by the fact that these decision-makers are typically activated by unstructured prompts, which simply request a final verdict without providing a systematic framework for resolving contradictions or mitigating cognitive biases.

Voting-based mechanisms aggregate agent outputs through majority or peer review paradigms (Li et al. 2024; Du et al. 2023; Xu et al. 2023), leveraging collective intelligence for more robust reasoning. Nonetheless, these methods are susceptible to individual agent unreliability and cognitive biases.

Overall, current approaches either centralize decision responsibilities or rely on simple aggregation, both with notable limitations in analytical depth and objectivity.

Rule-Based Reinforcement Learning

The use of reinforcement learning (RL) to enhance LLM reasoning has been notably advanced by DeepSeek-R1, which employs rule-based rewards and critic-free algorithms like Group Relative Policy Optimization (GRPO) (Guo et al. 2025). Subsequent analyses identified and mitigated GRPO's optimization biases (e.g., Dr.GRPO (Liu et al. 2025)), proposed simplifications such as Reinforce-Rej (Xiong et al. 2025) and DAPO (Yu et al. 2025), and ultimately removed policy constraints in GPG for further minimalism (Chu et al. 2025).

These developments highlight a progression from paradigm innovation to analytical refinement in rule-based RL for LLMs. However, the application of this paradigm to multi-agent settings remains largely unexplored.

Problem Definition and Formulation

We formalize the process of an MAS handling a user query s. The system comprises n *Execution Agents* $\{\pi_1, \pi_2, \ldots, \pi_n\}$ and a single *Decision Agent* π_D. The overall workflow is divided into two core phases:

Execution Phase In this phase, each agent π_i (where $i \in \{1, 2, \ldots, n\}$) independently receives the user query s and generates a candidate answer a_i according to its own policy $\pi_i(\cdot \mid s)$. Formally, this process can be represented as:

$$a_i \sim \pi_i(\cdot|s), \forall i \in \{1, 2, \ldots, n\}. \tag{1}$$

Decision Phase The Decision Agent π_D receives the full context, including the original query s and all candidate answers $\{a_1, a_2, \ldots, a_n\}$ from the execution agents. We define this context as the history $\mathbf{H} = \{s, a_1, a_2, \ldots, a_n\}$. Based on $\mathbf{H}$, π_D performs reasoning and produces the final, unified answer a_D for the system:

$$a_D \sim \pi_D(\cdot|H). \tag{2}$$

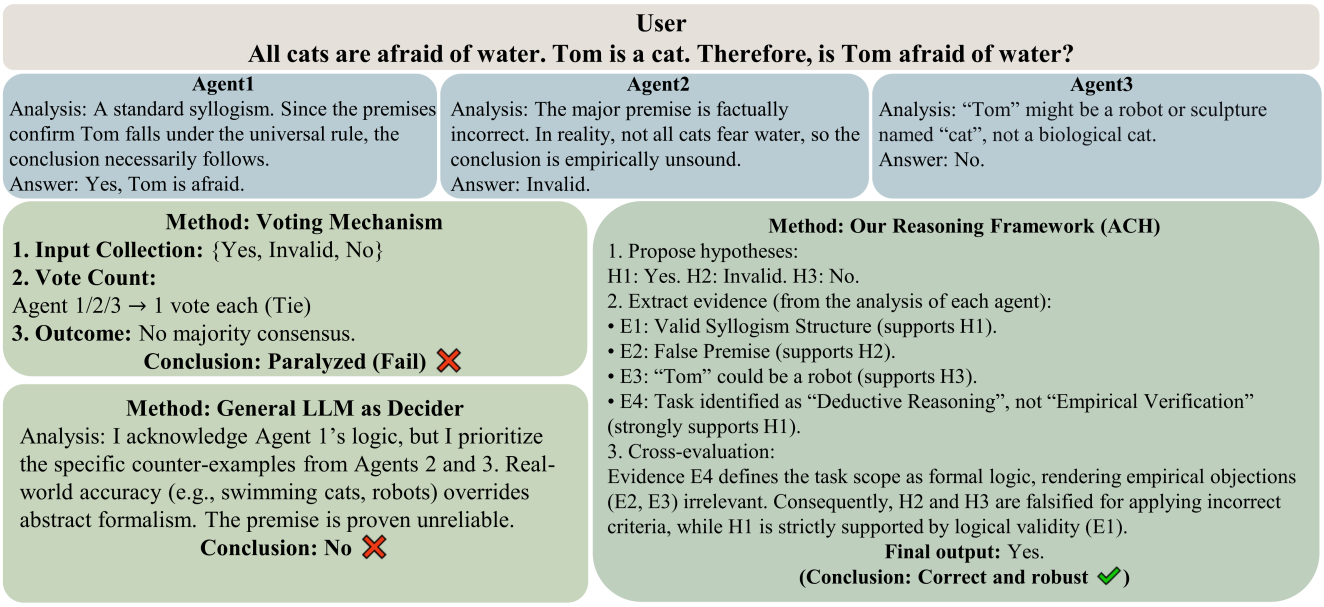

Figure 1: Comparison of collaborative decision-making strategies: (a) Voting-based methods fail due to a lack of consensus among agents, resulting in decision paralysis. (b) Dictatorial strategies rely on a single agent's judgment, which may be biased or unstable. (c) Our ACH-inspired protocol guides the decision agent through structured hypothesis evaluation, enabling more robust and bias-resistant reasoning.

Method

Although LLM-based MAS show great potential for solving complex problems through collective intelligence, how to effectively conduct the final collaborative decision remains an underexplored area. When interacting with a user, the system must provide a single, coherent output, which requires a powerful decision agent to act as the "final arbiter". Its necessity is twofold: 1) **Single Point of User Interaction**: The user needs a clear response, not multiple, potentially conflicting or redundant answers. 2) **Conflict Resolution and Information Synthesis**: When different agents provide contradictory or complementary information, a decision-making mechanism is needed to arbitrate, integrate, and construct the highest-quality answer.

However, training a decision agent with these capabilities presents significant challenges. Traditional methods like SFT struggle to internalize complex and robust reasoning capabilities into the model, and its generalization ability is limited, especially when facing conflicts or inconsistencies among agents.

To this end, we propose **AgentCDM**, a training framework designed to enhance an LLM's ability to act as a decision-maker in an MAS. Our goal is to enable the model not only to integrate responses from different agents but, more importantly, to generate a high-quality and robust final decision by applying structured reasoning, especially when faced with conflicts and contradictions. The overall architecture of this framework, illustrated in Figure 2, consists of two core operational phases (Execution and Decision) and is underpinned by a novel two-stage training paradigm, which we will describe in detail throughout this section.

Stage One: Structured Reasoning Protocol based on ACH

Recent research, such as DeepSeek-R1 (Guo et al. 2025), has demonstrated that reinforcement learning can effectively unlock the latent reasoning capabilities of pretrained language models. Building on this insight, we aim to equip the decision-making agent with more structured and cognitively aligned reasoning abilities early in training. To this end, our framework introduces explicit "scaffolding" in the first stage: a reasoning protocol inspired by the ACH. This protocol acts as an external guide, helping the agent learn to decompose decision problems, evaluate competing hypotheses, and reach bias-resistant conclusions through systematic analysis. Without such scaffolding, the model struggles to autonomously develop the desired reasoning structures, often reverting to shallow heuristics or inconsistent judgment patterns.

The ACH Protocol we proposed, illustrated in Figure 3, begins by formulating a comprehensive and mutually exclusive set of hypotheses to establish an unbiased *hypothesis space*. Concurrently, it requires the systematic collection of all relevant facts and arguments into a shared *evidence pool* before evaluation begins. The core of the framework is a *hypothesis-evidence matrix* that cross-evaluates the diagnostic value of each piece of evidence against every hypothesis (as consistent, inconsistent, or irrelevant). A crucial *meta-*

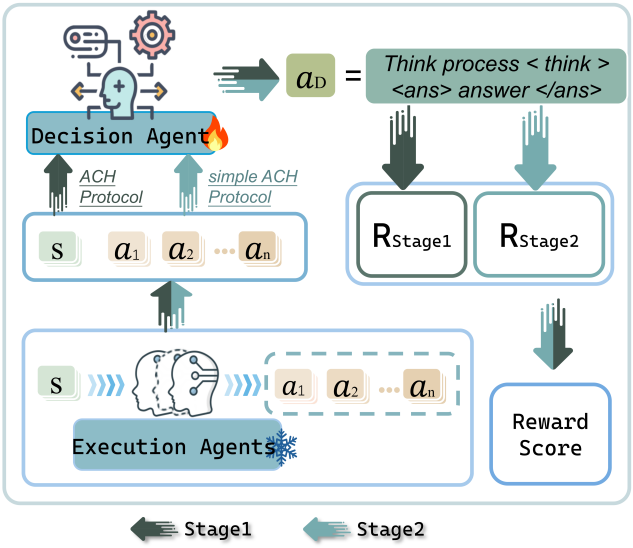

Figure 2: The AgentCDM framework. In the first stage, the framework uses a detailed ACH (Analysis of Competing Hypotheses) prompt as a structural guide or "scaffold". In the second stage, it employs a curriculum learning strategy to progressively remove this scaffold, encouraging the agent to learn and internalize the reasoning process.

cognitive review step is integrated to scrutinize and correct potential biases in this evaluation, ensuring analytical integrity. In drawing a conclusion, the protocol deliberately shifts the focus from confirmation to *falsification*, preliminarily selecting the hypothesis with the least amount of disconfirming evidence. This tentative conclusion is then immediately subjected to rigorous *adversarial testing* to proactively probe its robustness. Finally, the framework culminates in a comprehensive analytical report that not only articulates the final decision and its key evidentiary support but also details the rationale for rejecting alternative hypotheses and assesses the relative strengths and weaknesses of the conclusion.

In RL, the reward function is the core signal that guides model optimization. To train the decision-maker agent, we have designed a composite, rule-based reward function consisting of three parts:

Format rewards We require the model to encapsulate its thinking process within `<think>` tags and its final answer within `<answer>` tags. This design not only encourages the model to develop the habit of "thinking before answering", but also establishes a standardized output format that facilitates accurate downstream parsing and evaluation.

Accuracy rewards This reward is used to evaluate the correctness of the final answer given by the model. With the guidance of the format reward, the model provides the answer in the correct format, allowing us to provide feedback on correctness in a rule-based manner.

ACH rewards The core objective of this stage is to guide the model to strictly follow the ACH Protocol in its think-

Analyze according to the Analysis of Competing Hypotheses(ACH) provided below:

Step 1: Propose Hypotheses
State the core question and list all plausible, competing explanations as distinct hypotheses.
- H1 (from Agent 1): [Conclusion of Agent 1]
- H2 (from Agent 2): [Conclusion of Agent 2]

Step 2: List Evidence
Gather and document all relevant facts, data, and key arguments from your sources as discrete evidence items.
- E1 (from Agent 1): [Fact/Argument 1]
- E2 (from Agent 2): [Fact/Argument 2]

Step 3: Construct Hypothesis-Evidence Matrix
- Create a matrix with Hypotheses (H1, H2, ...) as columns and Evidence (E1, E2, ...) as rows.
- Evaluate the relationship between each piece of evidence and each hypothesis (supporting, inconsistent, or irrelevant).

Step 4: Refine the Matrix
Critically review the matrix for logical consistency, adjust any questionable ratings, and justify the changes.

Step 5: Draw Preliminary Conclusion
Identify the hypothesis that is most consistent with the evidence (has the fewest inconsistencies) as the leading candidate.

Step 6: Challenge the Conclusion
Actively stress-test the leading hypothesis by assuming it is wrong and articulating the strongest counter-arguments against it.

Step 7: State Final Analytic Report
Synthesize the analysis into two parts: an internal justification of the reasoning process, and a clear, user-friendly final answer.

Figure 3: Decision-Making Process Based on Analysis of Competing Hypotheses

ing process. We use pattern matching to verify whether the model's output within the `<think>` tags adheres to the ACH Protocol, thereby supervising its reasoning process.

Thus, the final score is computed as:

$$score_{\text{Stage1}} = score_{\text{format}} + score_{\text{answer}} + score_{\text{ACH}}. \quad (3)$$

Stage Two: Scaffolding Removal and Autonomous Exploration

The training in the first stage equips the model with the ability to apply the ACH Protocol under explicit guidance. To encourage the model to internalize this capability and to autonomously explore superior decision-making paths without "scaffolding", we designed a second stage. Directly removing the ACH Protocol would lead to training collapse, so we introduced a smoother transition mechanism.

Soft ACH rewards In the second stage of training, we transition from strict pattern matching to a more nuanced guidance mechanism. We introduce a soft structural reward, denoted as R_{soft_ACH}, which is defined by the semantic similarity between the agent's thought process and the ACH Protocol. To quantify this, we compute the cosine similarity of their vector embeddings, generated using the 1024-dimensional BGE-M3 model with mean pooling. This approach grants the model a more flexible exploration space.

Curriculum Annealing Guidance Simultaneously, to enable the model to explore more possibilities, we created two additional prompt types: a "Full ACH Protocol" and a "Simplified ACH Protocol". During training, we use a cosine annealing schedule to dynamically adjust the probability of sampling these two prompt types:

$$p_{\text{full}} = \frac{1}{2}\left(1 + \cos(\pi t)\right), \tag{4}$$

$$p_{\text{simple}} = 1 - p_{\text{full}}, \tag{5}$$

where $t = \frac{i}{T}$, with i denoting the current training step and T the total number of training steps.

Thus, the stage two score is computed as:

$$score_{\text{Stage2}} = score_{\text{format}} + score_{\text{answer}} + score_{\text{soft\_ACH}}. \tag{6}$$

Experiment

To comprehensively evaluate the effectiveness, generalization capability, and robustness of our proposed AgentCDM framework, we designed a series of rigorous experiments. This section will detail the benchmark datasets, baseline methods for comparison, specific implementation details, and an in-depth analysis of the experimental results.

Benchmarks

To ensure a comprehensive evaluation, we selected three widely-recognized Multiple-Choice Question Answering (MCQA) benchmarks that span diverse domains and difficulty levels. These include MMLU (Hendrycks et al. 2020), a large-scale benchmark covering 57 subjects to test broad knowledge; MMLU-Pro (Wang et al. 2024), a more challenging version with questions requiring complex reasoning and an expanded set of 10 options; and the most difficult subset of ARC, ARC-Challenge (Clark et al. 2018), which contains science questions demanding multi-step reasoning. This selection allows us to robustly measure the model's decision-making and reasoning performance across a spectrum of tasks.

Baselines

To evaluate the effectiveness of AgentCDM, we compare it against two major categories of baseline methods: dictatorial and voting-based strategies. We adopt the baseline configurations from GEDI (Zhao, Wang, and Peng 2024), which defines representative and standardized setups for both paradigms. This alignment ensures a fair and consistent comparison across methods.

Dictatorial Methods We evaluate an "Informed Dictatorial" method. In this approach, multiple "Executor Agents" first independently process the query to generate their own answers along with supporting analyses. Subsequently, a single "Decider Agent" evaluates all of these outputs and is solely responsible for producing the system's final decision. Crucially, this "Decider Agent" is typically activated with an unstructured prompt that simply asks for a final judgment without providing a systematic evaluation framework, leaving the quality of the outcome entirely dependent on the agent's unguided reasoning capabilities.

Voting-based Methods Consistent with GEDI, we selected a variety of voting-based methods to aggregate agent preferences. Our selection spans several mechanism types. We employ simple systems such as Plurality, which considers only first-choice votes, and the cardinal-based Range Voting. Our evaluation also includes several ordinal systems that leverage ranked preferences: Borda Count (Emerson 2013; Davies et al. 2014), which assigns points based on rank; Bucklin Voting (Erdélyi et al. 2015), which accumulates ranked choices to find a majority; and Instant-Runoff Voting (IRV) (Freeman, Brill, and Conitzer 2014), which operates through sequential elimination. Finally, we incorporate methods based on pairwise comparisons, namely Minimax (Brams, Kilgour, and Sanver 2007), which seeks to minimize the largest margin of defeat, and Ranked Pairs, which constructs a final ranking from the strongest pairwise victories.

Models and Implementation Details

To simulate a heterogeneous agent environment, we instantiated agents using representative LLMs, including open-source models such as GLM-4-9B-Chat, Mistral-7B, LLaMA-3-8B, and Qwen-2-72B, along with the closed-source GPT-4 (GLM et al. 2024; Jiang et al. 2023; Dubey et al. 2024; Team 2024; Achiam et al. 2023). Our AgentCDM model is trained from Qwen-7B-Base (Bai et al. 2023). Models are trained on A800 GPUs for 50–100 rollout–update iterations. Each batch samples P=256 prompts, with N=5 rollouts per prompt. Policy updates use GRPO, Adam optimizer. The sampling configurations are unified (temperature=0.6, top-p=0.95), and n=5 responses are generated per query. For fair comparison, we also evaluate Qwen-7B-Instruct and the reasoning-focused Qwen-7B-R1 under the ACH Protocol (Bai et al. 2023; Guo et al. 2025). Average Accuracy is used as the primary evaluation metric. Voting-based baselines follow a 5-shot in-context learning (ICL) setup, while Informed Dictatorial and ACH Protocol operate directly on Execution Agent outputs.

Main Results

Our comprehensive evaluation on the MMLU, MMLU_PRO, and ARC-Challenge benchmarks (Table 1) demonstrates the clear advantages of the proposed AgentCDM framework. Trained using our two-stage paradigm, AgentCDM consistently and significantly outperforms all baselines—including individual models, the unstructured Informed Dictatorial approach, various voting-based strategies, and models solely guided by the ACH protocol. The gains are especially pronounced on challenging tasks. For instance, on MMLU-PRO, AgentCDM improves the performance of Meta-Llama-3 and Mistral-7B by 22.5 and 29.3 percentage points over the Single Agent baseline, respectively, highlighting its ability to facilitate effective collaborative decision-making.

Table 1 also reports results for Qwen-7B-Instruct and Qwen-7B-R1 under ACH protocol guidance. Notably, Qwen-7B-R1 is a reasoning-optimized model with the same parameter scale. Yet, our AgentCDM, trained from a base model, achieves higher average accuracy when guided by

Base	Single	Dictatorial	Voting-Based						ACH Protocol			
		Informed	Plur.	Buck.	Borda	IRV	Minimax	RP	Homo.	Qwen-7B-Inst.	Qwen-7B-R1	AgentCDM(Ours)
MMLU												
glm-4-9b-chat	67.0	70.6	67.7	67.9	67.5	67.7	68.0	68.0	74.2	75.1	78.9	**79.6**
Llama-3-8B	66.0	65.2	66.5	66.7	66.6	66.5	66.7	66.7	68.4	70.1	**76.8**	76.4
Mistral-7B	59.4	50.9	59.9	59.9	61.0	60.9	60.9	60.9	61.2	64.0	72.7	**74.3**
Qwen-2-72b	68.7	72.1	68.6	68.6	68.5	68.6	68.6	68.6	76.2	74.8	77.2	**78.1**
GPT-4	84.7	**85.7**	84.5	84.6	84.4	84.5	84.6	84.6	83.8	74.7	84.5	85.6
Average	69.2	69.8	68.9	69.4	69.5	69.6	69.6	69.8	72.8	71.7	78.0	**78.8**
MMLU_PRO												
glm-4-9b-chat	47.4	48.3	47.5	47.7	47.4	47.5	47.9	47.7	48.8	49.5	57.5	**63.9**
Llama-3-8B	41.2	40.8	41.4	41.2	41.6	41.4	41.6	41.4	42.3	44.5	54.4	**63.7**
Mistral-7B	32.5	31.7	32.0	32.0	31.9	32.0	32.1	32.0	31.5	37.2	48.6	**61.8**
Qwen-2-72b	49.2	51.1	49.0	49.2	49.5	49.0	49.3	49.3	53.4	53.1	60.8	**65.7**
GPT-4	69.0	**71.2**	69.8	69.6	69.4	69.8	69.6	69.5	64.2	58.1	68.5	71.0
Average	47.9	48.0	48.6	47.9	47.9	48.0	48.0	48.1	48.0	48.5	58.0	**65.2**
ARC-Challenge												
glm-4-9b-chat	84.6	88.8	85.0	85.0	85.0	85.0	84.0	84.0	89.4	90.8	**92.6**	91.8
Llama-3-8B	77.2	72.4	77.0	77.0	77.0	77.0	77.0	77.0	77.4	76.6	**81.8**	81.0
Mistral-7B	63.2	61.2	62.0	62.0	63.0	62.0	62.0	62.0	66.8	70.2	81.6	**84.0**
Qwen-2-72b	86.7	85.6	85.0	85.0	86.0	85.0	85.0	85.0	88.8	87.6	90.0	**91.6**
GPT-4	91.3	91.5	92.9	91.9	91.9	92.9	91.9	92.0	92.0	86.0	92.0	**93.4**
Average	80.6	80.0	79.9	80.4	80.2	80.6	80.4	80.0	82.9	82.2	87.6	**88.4**
All Average	65.9	66.0	65.8	65.9	65.9	66.0	66.0	66.0	67.9	67.5	74.5	**77.5**

Table 1: **Comprehensive performance comparison of AgentCDM against various baseline methods on the MMLU, MMLU_PRO, and ARC-Challenge datasets.** All values represent accuracy (%). The results show that AgentCDM consistently outperforms other approaches. Baselines include the model's standalone performance (Single Agent), six Voting-Based methods, and an unstructured Informed Dictatorial method. The ACH Protocol group evaluates different decision agents: in the Homogeneous (Homo.) setting, the base model itself is used as the decision agent, whereas Qwen-7B-Instruct (Inst.) and Qwen-7B-R1 serve as fixed external decision agents. For each row, the best-performing method is **highlighted in bold**, and the second-best is underlined. *Note:* Single=Single Agent, Plur.=Plurality, Buck.=Bucklin, Borda=Borda Count, RP=Ranked Pairs, Homo.=Homogeneous, Inst.=Instruct.

the same protocol. For example, on the challenging MMLU-PRO benchmark, AgentCDM achieves 65.2% accuracy, outperforming Qwen-7B-R1's already strong 58.0%. This result highlights the effectiveness of our two-stage training framework in enhancing structured reasoning, even compared to specialized reasoning models.

Further analysis reveals the inherent power of structured reasoning. Remarkably, even without parameter tuning, introducing the ACH Protocol as a zero-shot prompting strategy significantly enhances performance. For example, the Homogeneous variant—which functions as both executor and decision-maker—generally outperforms the unstructured Informed Dictatorial method when guided by the ACH Protocol, surpassing it by 1.9 percentage points on the All Average metric and consistently outperforming all Voting-Based methods. These results show that a systematic, evidence-driven evaluation of competing hypotheses is, in itself, an effective approach to promoting rigorous, bias-resistant reasoning.

Cross-Dataset Generalization Evaluation

To further assess the generalization capability of our AgentCDM framework, we conducted cross-dataset transfer evaluations. Specifically, a decision-making agent trained on a particular dataset (e.g., MMLU-PRO) was directly evaluated on previously unseen test sets (e.g., MMLU and ARC-Challenge).

	Testing Dataset		
Training Dataset	**MMLU**	**MMLU_PRO**	**ARC-Challenge**
MMLU	**78.5**	55.0	91.0
MMLU_PRO	80.8	**65.2**	94.0
ARC-Challenge	74.9	52.0	**89.5**

Table 2: Cross-dataset generalization performance of the AgentCDM framework. Each row specifies the training dataset, and each column specifies the evaluation dataset. Values indicate accuracy(%). Diagonal results (in bold) show in-domain performance.

As shown in the table 2, models trained on the more challenging MMLU-PRO dataset exhibited remarkable generalization performance. Notably, the model achieved a score of 80.84 on the MMLU test set and 94.0 on the ARC-Challenge test set. These results not only demonstrate strong performance but even surpass the scores of models trained directly on MMLU and ARC-Challenge (which achieved 78.52 and 89.5, respectively).

These findings provide strong evidence that training on more complex and demanding datasets enables AgentCDM

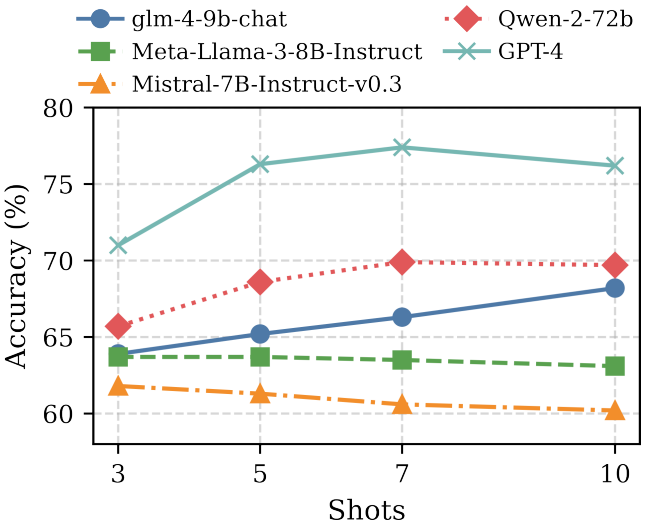

Figure 4: Effect of agent quantity on MMLU-PRO performance. Values indicate accuracy (%).

Dataset	Qwen-7B-Instruct(ACH)	AgentCDM
MMLU	75.2	**78.9**
MMLU_PRO	55.9	**67.9**
ARC-Challenge	87.2	**93.0**

Table 3: Performance evaluation in a heterogeneous agent pool. Values indicate accuracy (%).

to acquire deeper, more generalizable structured reasoning capabilities. Such capabilities transcend specific knowledge domains and can be effectively transferred to diverse tasks, leading to high-quality decision-making across domains. In contrast, models trained on relatively simpler datasets (e.g., ARC-Challenge) perform poorly when transferred to more challenging tasks (e.g., MMLU-PRO), further highlighting the critical role of high-quality and high-difficulty training data in cultivating robust general decision-making abilities.

Robustness and Scalability Analysis

To evaluate AgentCDM's performance in settings that resemble real-world applications, we conducted experiments on its scalability and robustness.

First, we tested scalability by varying the number of agents on the MMLU-PRO dataset (Figure 4), uncovering a stark dual phenomenon. For capable models (e.g., GPT-4), performance scaled positively with more agents, demonstrating effective "collective intelligence". Conversely, for less capable models (e.g., Mistral-7B), performance degraded as the number of agents increased. We conclude that this is because aggregating multiple weaker inputs amplifies noise more than signal, overwhelming the decision-making process.

Second, to assess the robustness of decision-makers in more realistic settings, we build a heterogeneous agent pool comprising outputs from five distinct models. For each query, the decision-making agent receives as input a combination of three outputs randomly selected from this pool. As shown in Table 3, the fully trained AgentCDM significantly outperformed Qwen-7B-Instruct with ACH Protocol, achieving a score of 67.9 versus the baseline's 55.9 on MMLU-PRO. This result, combined with the scalability findings, strongly validates our two-stage training paradigm. It successfully endows the agent with internalized analytical skills to critically evaluate diverse and conflicting information, demonstrating the robustness crucial for complex, dynamic environments.

Method	MMLU	MMLU_PRO	ARC-Challenge
Full	**78.5**	**65.2**	**96.0**
Vanilla prompt	72.6	52.5	89.0
Scaffolding-Only	68.1	49.4	84.0
Exploration-Only	71.5	52.1	89.7

Table 4: Ablation study results on three benchmarks. "Full" is our complete two-stage AgentCDM framework. "Scaffolding-Only" and "Exploration-Only" refer to using only Stage 1 or Stage 2 of our training, respectively. Values indicate accuracy (%).

Ablation Study

To validate the necessity of our proposed two-stage training paradigm, we conducted a series of ablation studies (Table 4) to isolate the contributions of Stage 1 (Structured Scaffolding) and Stage 2 (Autonomous Exploration). The results clearly reveal their synergistic effect. The Full model (AgentCDM), incorporating both stages, consistently achieved the best performance across all datasets. In contrast, removing the second stage (Scaffolding-Only) led to a drastic performance drop, indicating that rigid adherence to the structured protocol impairs generalization. Conversely, removing the first stage (Exploration-Only) resulted in mediocre performance comparable to a simple baseline, demonstrating that unguided exploration is inefficient without a solid foundation in structured reasoning. These experiments provide compelling evidence that the two stages are both indispensable and complementary: Stage 1 endows the model with a robust reasoning core, while Stage 2 fosters the generalization and adaptability crucial for superior performance.

Conclusion and Limitations

We presented AgentCDM, an ACH-inspired framework that enhances multi-agent decision-making through structured scaffolding and autonomous exploration. Empirical results demonstrate that AgentCDM significantly outperforms voting-based and dictatorial baselines in accuracy and robustness. Limitations include reliance on the quality of initial hypotheses and the assumption of cooperative agents. Future work will extend the framework to adversarial environments and explore adaptive scaffolding strategies.

Acknowledgments

The work was supported by the National Key R&D Program of China (No.2022ZD0116307) and the National Nat-

ural Science Foundation of China (No.62271270).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Bose, T.; Reina, A.; and Marshall, J. A. 2017. Collective decision-making. *Current opinion in behavioral sciences*, 16: 30–34.

Brams, S. J.; Kilgour, D. M.; and Sanver, M. R. 2007. A minimax procedure for electing committees. *Public Choice*, 132(3): 401–420.

Chen, Z.; Liu, K.; Wang, Q.; Liu, J.; Zhang, W.; Chen, K.; and Zhao, F. 2024. Mindsearch: Mimicking human minds elicits deep ai searcher. *arXiv preprint arXiv:2407.20183*.

Chu, X.; Huang, H.; Zhang, X.; Wei, F.; and Wang, Y. 2025. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Davies, J.; Katsirelos, G.; Narodytska, N.; Walsh, T.; and Xia, L. 2014. Complexity of and algorithms for the manipulation of Borda, Nanson's and Baldwin's voting rules. *Artificial Intelligence*, 217: 20–42.

Du, Y.; Li, S.; Torralba, A.; Tenenbaum, J. B.; and Mordatch, I. 2023. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv e-prints*, arXiv–2407.

Emerson, P. 2013. The original Borda count and partial voting. *Social Choice and Welfare*, 40(2): 353–358.

Erdélyi, G.; Fellows, M. R.; Rothe, J.; and Schend, L. 2015. Control complexity in Bucklin and fallback voting: A theoretical analysis. *Journal of Computer and System Sciences*, 81(4): 632–660.

Freeman, R.; Brill, M.; and Conitzer, V. 2014. On the axiomatic characterization of runoff voting rules. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.

Fu, T.; Xu, X.; Xu, W.; Chen, J.; Ren, R.; Deng, B.; Zhao, X.; Cao, J.; and Cao, X. 2025. Two Heads are Better than One: Distilling Large Language Model Features Into Small Models with Feature Decomposition and Mixture. arXiv:2511.07110.

GLM, T.; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; Yin, D.; Zhang, D.; Rojas, D.; Feng, G.; Zhao, H.; et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hadi, M. U.; Qureshi, R.; Shah, A.; Irfan, M.; Zafar, A.; Shaikh, M. B.; Akhtar, N.; Wu, J.; Mirjalili, S.; et al. 2023. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints*, 1: 1–26.

Hao, R.; Hu, L.; Qi, W.; Wu, Q.; Zhang, Y.; and Nie, L. 2025. Chatllm network: More brains, more intelligence. *AI Open*, 6: 45–52.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Heuer, R. J. 1999. *Psychology of intelligence analysis*. Center for the Study of Intelligence.

Hong, S.; Zheng, X.; Chen, J.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S. K. S.; Lin, Z.; Zhou, L.; et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 3(4): 6.

Huang, D.; Zhang, J. M.; Luck, M.; Bu, Q.; Qing, Y.; and Cui, H. 2023. Agentcoder: Multi-agent-based code generation with iterative testing and optimisation. *arXiv preprint arXiv:2312.13010*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. arXiv:2310.06825.

King, A. J.; and Cowlishaw, G. 2007. When to use social information: the advantage of large group size in individual decision making. *Biology letters*, 3(2): 137–139.

Lei, Y.; Ge, X.; Zhang, Y.; Yang, Y.; and Ma, B. 2025. Do Large Language Models Think Like the Brain? Sentence-Level Evidence from fMRI and Hierarchical Embeddings. arXiv:2505.22563.

Li, G.; Hammoud, H.; Itani, H.; Khizbullin, D.; and Ghanem, B. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36: 51991–52008.

Li, J.; Zhang, Q.; Yu, Y.; Fu, Q.; and Ye, D. 2024. More agents is all you need. *arXiv preprint arXiv:2402.05120*.

Liang, T.; He, Z.; Jiao, W.; Wang, X.; Wang, Y.; Wang, R.; Yang, Y.; Shi, S.; and Tu, Z. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Lin, B. Y.; Fu, Y.; Yang, K.; Brahman, F.; Huang, S.; Bhagavatula, C.; Ammanabrolu, P.; Choi, Y.; and Ren, X. 2023. Swiftsage: A generative agent with fast and slow thinking

for complex interactive tasks. *Advances in Neural Information Processing Systems*, 36: 23813–23825.

Liu, Z.; Chen, C.; Li, W.; Qi, P.; Pang, T.; Du, C.; Lee, W. S.; and Lin, M. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Lu, C.; Lu, C.; Lange, R. T.; Foerster, J.; Clune, J.; and Ha, D. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Min, S.; Krishna, K.; Lyu, X.; Lewis, M.; Yih, W.-t.; Koh, P. W.; Iyyer, M.; Zettlemoyer, L.; and Hajishirzi, H. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Shen, W.; Li, C.; Chen, H.; Yan, M.; Quan, X.; Chen, H.; Zhang, J.; and Huang, F. 2024. Small llms are weak tool learners: A multi-llm agent. *arXiv preprint arXiv:2401.07324*.

Team, Q. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Wang, Y.; Ma, X.; Zhang, G.; Ni, Y.; Chandra, A.; Guo, S.; Ren, W.; Arulraj, A.; He, X.; Jiang, Z.; et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290.

Wolf, Y.; Wies, N.; Avnery, O.; Levine, Y.; and Shashua, A. 2023. Fundamental limitations of alignment in large language models. *arXiv preprint arXiv:2304.11082*.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.

Xiong, W.; Yao, J.; Xu, Y.; Pang, B.; Wang, L.; Sahoo, D.; Li, J.; Jiang, N.; Zhang, T.; Xiong, C.; et al. 2025. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*.

Xu, Z.; Shi, S.; Hu, B.; Yu, J.; Li, D.; Zhang, M.; and Wu, Y. 2023. Towards reasoning in large language models via multi-agent peer review collaboration. *arXiv preprint arXiv:2311.08152*.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Dai, W.; Fan, T.; Liu, G.; Liu, L.; et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Zhao, X.; Wang, K.; and Peng, W. 2024. An electoral approach to diversify llm-based multi-agent collective decision-making. *arXiv preprint arXiv:2410.15168*.