

# Traceable Latent Variable Discovery Based on Multi-Agent Collaboration

Huaming Du

Southwestern University of Finance  
and Economics  
Chengdu, Sichuan, China

Tao Hu

Southwestern University of Finance  
and Economics  
Chengdu, Sichuan, China

Yijie Huang

Southwestern University of Finance  
and Economics  
Chengdu, Sichuan, China

Yu Zhao\*

Southwestern University of Finance  
and Economics  
Chengdu, Sichuan, China

Guisong Liu

Southwestern University of Finance  
and Economics  
Chengdu, Sichuan, China

Tao Gu

Southwestern University of Finance  
and Economics  
Chengdu, Sichuan, China

Gang Kou

Hunan University of Technology and  
Business  
Xiangjiang Laboratory  
Hunan, China

Carl Yang\*

Emory University  
Atlanta, Georgia, United States  
j.carlyang@emory.edu

## Abstract

Revealing the underlying causal mechanisms in the real world is crucial for scientific and technological progress. Despite notable advances in recent decades, the lack of high-quality data and the reliance of traditional causal discovery algorithms (TCDA) on the assumption of no latent confounders, as well as their tendency to overlook the precise semantics of latent variables, have long been major obstacles to the broader application of causal discovery. To address this issue, we propose a novel causal modeling framework, **TLVD**, which integrates the metadata-based reasoning capabilities of large language models (LLMs) with the data-driven modeling capabilities of TCDA for inferring latent variables and their semantics. Specifically, we first employ a data-driven approach to construct a causal graph that incorporates latent variables. Then, we employ multi-LLM collaboration for latent variable inference, modeling this process as a game with incomplete information and seeking its Bayesian Nash Equilibrium (BNE) to infer the possible specific latent variables. Finally, to validate the inferred latent variables across multiple real-world web-based data sources, we leverage LLMs for evidence exploration to ensure traceability. We comprehensively evaluate TLVD on three de-identified real patient datasets provided by a hospital and two benchmark datasets. Extensive experimental results confirm the effectiveness and reliability of TLVD, with average improvements of 32.67% in Acc, 62.21% in CAcc, and 26.72% in ECit across the five datasets.

\*Corresponding author

## CCS Concepts

• **Mathematics of computing** → **Causal networks**; • **Information systems** → **Web searching and information discovery**.

## Keywords

Latent variable discovery, Information Retrieval, Large language model, Healthcare

## ACM Reference Format:

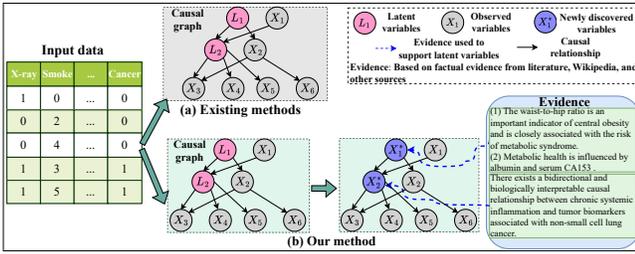
Huaming Du, Tao Hu, Yijie Huang, Yu Zhao, Guisong Liu, Tao Gu, Gang Kou, and Carl Yang. 2026. Traceable Latent Variable Discovery Based on Multi-Agent Collaboration. In *Proceedings of the ACM Web Conference 2026 (WWW '26)*, April 13–17, 2026, Dubai, United Arab Emirates. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3774904.3792244>

## 1 Introduction

Causal discovery aims to identify causal relationships from observational data and has been successfully applied in many fields [12, 37]. However, traditional methods—such as the PC algorithm [45], GES [10], and LiNGAM [44]—typically assume that no latent confounders exist in the causal graph, an assumption that often does not hold in many real-world scenarios. Therefore, extensive research has been devoted to addressing this issue from two directions to improve causal structure learning.

The first line of research focuses on inferring causal structures among observed variables despite the possible existence of latent confounders. Representative approaches include FCI and its variants based on conditional independence tests [3, 36], as well as over-complete ICA-based techniques that further exploit non-Gaussianity [43]. The second line of research focuses more on inferring causal structures among latent variables under the assumption that observed variables are not directly connected. This category includes Tetrad condition-based approaches [24], high-order moments-based methods [9, 49], matrix decomposition-based approaches [4], and mixture oracle-based methods [23]. Recently, Dong et al. [11] proposed a three-phase causal discovery algorithm





**Figure 1: A toy example of latent variable discovery using tabular data.**

based on rank constraints that can identify the complete causal structure involving both observed and latent variables. However, as shown in Figure 1(a), existing studies focus on recovering causal structures involving latent variables but rarely infer the specific latent variables and their semantics.

In recent years, large language models (LLMs) [1, 16] have achieved remarkable breakthroughs in natural language understanding and generation, marking a critical milestone in the pursuit of artificial general intelligence. As the capabilities of LLMs continue to advance, LLM-based agents [39, 46] are increasingly becoming core components for integrating domain expertise and tools, effectively transforming these technological advances into practical applications. Building upon this paradigm, multi-agent systems [26, 41] incorporate multiple diverse agents to coordinate and leverage their respective strengths, offering high flexibility and adaptability to provide more comprehensive solutions to complex real-world problems. The rapid development of LLMs has also provided new solutions for causal discovery [54].

This paper focuses on a more challenging problem: **traceable latent variable discovery**, i.e., *identifying specific latent variables and retrieving supporting evidence from multiple web data sources*. This setting is more general and more practical for handling many real-world problems, such as uncovering unknown factors contributing to diseases. This challenge involves two fundamental questions: (i) How can multiple LLM-based agents be coordinated to efficiently propose and define possible latent variables? (ii) How can evidence be retrieved from diverse web data sources (e.g., academic websites, Wikipedia, open databases, etc.) to consolidate the discovered latent variables?

To address this challenging problem, we propose a multi-agent collaboration framework for latent variable discovery, which simultaneously leverages real web data sources for evidence tracing in (causal) knowledge discovery. Specifically, we first employ latent variable causal discovery algorithms to recover causal graphs containing latent variables. Then, based on the recovered graph structure, we introduce game-theoretic principles, where multiple agents maintain their own belief networks to enable efficient collaboration, and utilize Bayesian Nash Equilibrium to hypothesize latent variables. Finally, we utilize LLMs to iteratively retrieve and exploit causal evidence across multiple real-world web data sources.

We conducted extensive experiments on our own real-world hospital dataset WCHSU as well as two benchmark datasets. Comprehensive evaluation results demonstrate that TLVD exhibits clear advantages over existing SOTA methods. The experimental results

and discussions, ablation studies, parameter analyses, case studies, and failure analyses, will be presented and analyzed in Section 4.

The main contributions of this paper are as follows:

- A unified latent variable discovery framework called TLVD, which integrates LLMs with TCDA and employs multi-agent collaboration to infer more plausible and semantically interpretable latent variables.
- First, we incorporate the prior knowledge of LLMs, allowing them to serve as priors for latent variable explanation. Then, we model the multi-agent collaboration process as a game of incomplete information to efficiently reason about latent variables. Meanwhile, causal evidence is retrieved from multiple web data sources, and the latent variables are iteratively updated and validated.
- Extensive experimental results on various real-world medical and generic benchmark datasets demonstrate the superiority of TLVD.

## 2 Related work

### 2.1 Causal Discovery

Traditional and LLM-based causal discovery algorithms [12, 25] usually assume that all task-relevant variables are observable [34, 38]. However, latent variables are prevalent in practice and can lead these methods to produce spurious causal relations, which has motivated extensive research on causal discovery with latent variables. Existing methods for handling latent variables in causal discovery can be broadly categorized into nine classes: Conditional independence constraints, Tetrad condition, Over-complete independent component analysis, Generalized independent noise, Mixture oracles-based, Rank deficiency, Heterogeneous data, and Score-based methods. Many methods fall within the constraint-based framework, combining conditional independence tests with algebraic constraints to infer causal relations, with representative approaches based on rank or tetrad constraints [11, 21]. While most current methods still follow the constraint-based paradigm, some recent studies have started to formalize score-based approaches for latent causal discovery [35, 55]. Although these methods have advanced the field of causal discovery, none of them attempt to identify the specific latent variables and their semantics.

### 2.2 LLM-based Multi-Agent Systems

LLMs have exhibited remarkable proficiency in tackling various sophisticated tasks. However, they still suffer from several inherent limitations, such as hallucination [22], their autoregressive nature (e.g., inability to engage in slow thinking [17]), and constraints imposed by scaling laws. Recent studies have extensively explored multi-agent collaboration frameworks based on LLMs, aiming to tackle complex cognitive and decision-making tasks [51, 53]. One prominent paradigm involves explicit role-playing mechanisms to simulate human collaborative dynamics, assigning different LLM agents to specialized roles within an organization [20]. Other frameworks further enhance multi-agent collaboration through voting and consensus mechanisms [28], collective reasoning or discussion-based methods [7], and structured (e.g., graph [32, 33]-based) agentic debate approaches [13, 52], aiming to improve factual accuracy and logical consistency. However, current popular multi-LLM collaboration frameworks lack solid theoretical foundations, offer no

guarantees of convergence or cooperation, and have yet to be explored in the field of causal discovery.

### 3 Traceable Latent Variable Discovery Framework

In this section, building upon game theory and reinforcement learning, we propose an evidence-traceable multi-agent collaboration framework for exploring latent variables and their semantics. As illustrated in Figure 2, the framework primarily consists of three stages: (i) Stage I: Identification of latent causal graph structures; (ii) Stage II: Utilization of multi-agent collaboration to identify latent variables; (iii) Stage III: Validation of latent variables. It is important to note that this paper treats LLMs as large-scale background knowledge providers, while the core identifiability guarantees still rely on classical latent variable discovery algorithms.

#### 3.1 Identifying Latent Causal Graph Structures

In this paper, we aim to address a more general scenario for latent variable causal discovery, where observed variables can be directly adjacent, and latent variables can flexibly be related to all other variables. That is, hidden variables can serve as confounders, mediators, or effects of latent or observed variables, and even form a hierarchical structure (see Figure 1). This setup is quite general and practically meaningful for addressing many real-world problems. RLCD [11] is currently the most advanced algorithm with theoretical guarantees, and thus, we use it to identify latent causal structures. In addition, we also explore the impact of other latent causal structure identification methods on the performance of our framework (see Appendix D). The specific formulation is as follows:

$$\mathcal{G}' = \text{RLCD}(\mathbb{X}_{\mathcal{G}}), \quad (1)$$

where  $\mathbb{X}_{\mathcal{G}}$  denotes the data samples containing  $n$  observed variables, and  $\mathcal{G}'$  represents the Markov equivalence class.

#### 3.2 Identifying Latent Variables

Previous research [11, 21] has primarily focused on revealing the positional and structural information of latent variables, while overlooking the discovery of the latent variables themselves and their semantics. However, identifying latent variables is crucial for causal inference and its practical applications. Although LLMs are capable of performing complex tasks such as creative writing, reasoning, and decision-making, with abilities that in some aspects even surpass human level, they are still constrained by hallucinations [22], their autoregressive nature (e.g., inability to perform slow thinking [17]), and scaling laws [42]. Inspired by the theory of mind [14], multiple LLM-based agents systems have been developed, enabling teamwork and specialization by integrating the strengths and perspectives of individual agents to achieve shared goals. Therefore, in this paper, we adopt LLM-based multi-agent collaboration to identify latent variables and their semantics.

$$\mathcal{X}_i = \text{MALLM}(x_i, L_i), \quad (2)$$

where *MALLM* is the multi-agent collaboration system designed in this work,  $L_i$  denotes the  $i$ -th latent variable,  $x_i$  represents the Markov blanket corresponding to the  $L_i$  in  $\mathcal{G}'$ , and  $\mathcal{X}_i$  denotes the concrete variable with explicit semantics corresponding to  $L_i$ .

However, achieving efficient collaboration among agents in MALLM is not straightforward and faces *three key challenges*. First, extensive inter-agent communication consumes a large number of tokens, increasing computational overhead [13]. Second, the volume of information exchanged over multiple rounds can exceed the context-window capacity of LLMs, limiting system scalability [31]. Third, without well-defined coordination protocols, these systems may underperform compared to simple ensembling or self-consistency methods [29]. To address these challenges, we innovatively propose a novel approach for efficient coordination via Bayesian Nash Equilibrium (BNE), modeling multi-LLM interactions as an incomplete-information game to identify latent variables and their semantics.

Next, we first define the process, then provide a detailed introduction to the MALLM, and finally conduct the theoretical analysis.

**3.2.1 Process Definition.** We consider a system composed of  $N - 1$  execution LLMs and a coordinator LLM, where agents coordinate and interact solely based on their beliefs. Consistent with prior studies [30], we formally model this as a decentralized partially observable Markov decision process (DEC-POMDP), defined as a Markov game  $\langle \mathcal{S}, \mathcal{A}, O, \mathcal{P}, \Omega, \mathcal{R}, \gamma \rangle$ . Here,  $\mathcal{S}$  represents the state space, including user queries and dialogue context;  $\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_N$  is the joint action space, with each  $\mathcal{A}_i$  defining agent  $i$ 's action as a prompt embedding  $a_i = [T_i, p_i]$ , which controls the LLM's output behavior via temperature and repetition penalty;  $N$  is the number of agents;  $O$  is the joint observation space;  $\mathcal{P}$  and  $\Omega$  define the state transition and observation functions;  $\mathcal{R}$  is the reward, and  $\gamma$  is the discount factor.

Our objective is to identify a policy profile  $\pi = (\pi_1, \dots, \pi_N)$  that forms a BNE through belief coordination, such that no individual agent can improve the quality of its latent-variable generation and semantic inference by unilaterally changing its policy.

**3.2.2 BNE Implementation with MALLM.** In this section, we introduce a MALLM framework compliant with DEC-POMDP. The MLLM adopts a hierarchical architecture where multiple executor LLMs operate locally under the guidance of a coordinator LLM. The framework consists of two stages: Inference (see Figure 4) and Optimization (see Figure 3). Below, we provide a detailed introduction to the core modules.

**(1) Belief Update.** Each execution LLM  $i$  maintains a belief network  $B_i(\tau_i^t, o_i^t; \theta_i^B)$ , which implements its policy  $\pi_i$  by mapping the local trajectory  $\tau_i^t$  (composed of its previous actions and observations) and current observation  $o_i^t \in O_i$  into a belief state  $b_i \in \mathbb{R}^d$ . The belief state characterizes the agent's understanding of the environment and the behaviors of other agents under partial observability, enabling strategic decision-making without direct access to others' outputs. The belief state  $b_i$  is further used to generate the prompt embedding  $e_i = [T_i, p_i]$ , which controls the LLM's output regarding the latent variables and their semantics. The specific formulation is as follows:

$$\begin{aligned} T_i &= \sigma(W_T b_i + b_T) \\ p_i &= \sigma(W_p b_i + b_p), \end{aligned} \quad (3)$$

where  $\sigma(\cdot)$  is the sigmoid function.  $W_T$  and  $b_T$  are learnable parameters. The belief network  $B_i$  produces two outputs: (1) The prompt

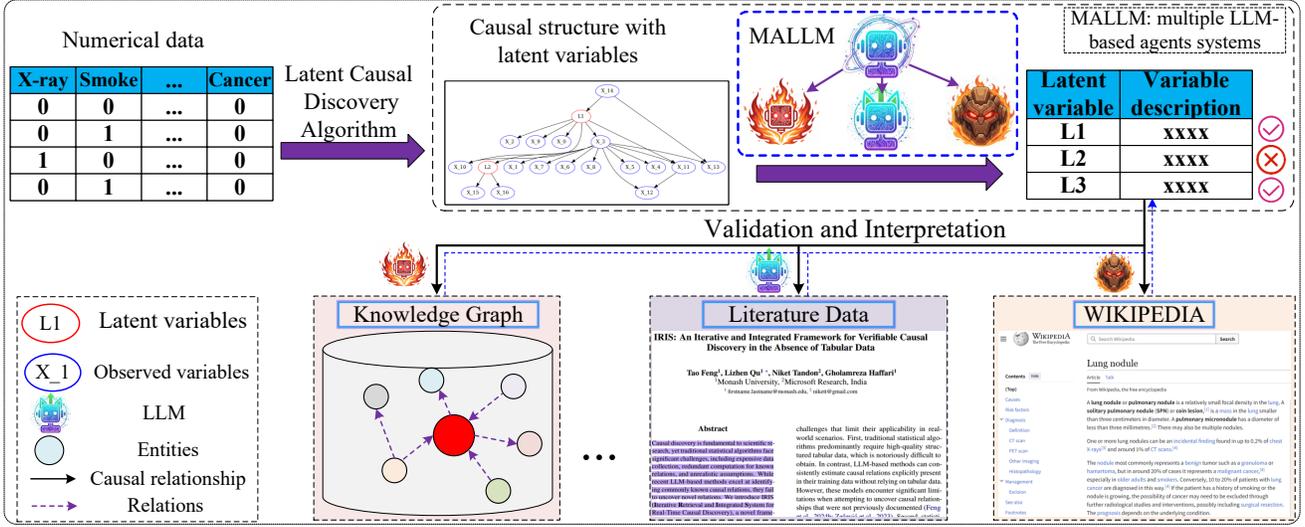


Figure 2: The overview of TLVD framework.

embedding  $e_i$ , which serves as the action in the DEC-POMDP framework; (2) The local Q-value  $Q_t^i(\tau_t^i, e_t^i; \varphi_i)$ , which estimates the expected return from the current belief state. The belief state  $b_i$  is also passed to a belief encoder for group-level processing.

To optimize the belief network, we apply a Temporal Difference (TD) loss to update the parameters  $\theta_i^B = \{\varphi_i, W_T, b_T, W_p, b_p\}$ , which include the Q-value function parameters  $\varphi_i$  and the prompt embedding parameters, as illustrated in Figure 3. The TD loss is defined as follows:

$$\mathcal{L}_{TD}^i(\theta_i^B) = \mathbb{E}_{\mathcal{D}} \left[ \left( r_t^i + \gamma \max_{e_{t+1}^i} Q_{t+1}^i(\tau_{t+1}^i, e_{t+1}^i; \varphi_i') - Q_t^i(\tau_t^i, e_t^i; \varphi_i) \right)^2 \right], \quad (4)$$

where  $r_t^i = \mathcal{R}(s^t, a^t)$  denotes the local reward signal, and  $\varphi_i'$  represents the target network parameters updated via soft update mechanism. By minimizing  $\mathcal{L}_{TD}^i$ , execution LLM  $i$  refines its belief state to improve local decision-making. To enable agents to implicitly exchange beliefs and thereby facilitate system convergence to BNE, we employ an attention mechanism within the belief encoder to capture inter-agent dependencies in the belief states:

$$\mathcal{B} = \text{Attention}(W^Q \mathbf{b}, W^K \mathbf{b}, W^V \mathbf{b}; \theta_e), \quad (5)$$

where  $\mathbf{b} = [b_1; \dots; b_N] \in \mathbb{R}^{Nd}$  is the concatenated vector of the individual belief states  $\{b_i\}_{i=1}^N$ . The final output is:  $E = \mathcal{B}W^O$ , where  $\{W^Q, W^K, W^V\}$  are learnable parameters and  $W^O$  is the output projection. The belief encoder captures high-level interactions among execution LLMs, ensuring coherence in group behavior.

**(2) Mixing Network.** Inspired by Qmax [40], the mixing network coordinates the integrated belief information from all execution LLMs, thereby driving the overall system toward optimization with respect to the BNE. Specifically, each agent's prompt embedding  $\{e_t^i\}_{i=1}^N$  is first processed through a self-attention mechanism to capture inter-agent dependencies, producing intermediate embeddings  $\{v_t^i\}_{i=1}^N$ . These intermediate embeddings  $\{v_t^i\}_{i=1}^N$  are then combined with the group-level belief representation  $E_t$  to produce feature transformations  $\{F_t^i\}_{i=1}^N$ . This combination enables the network

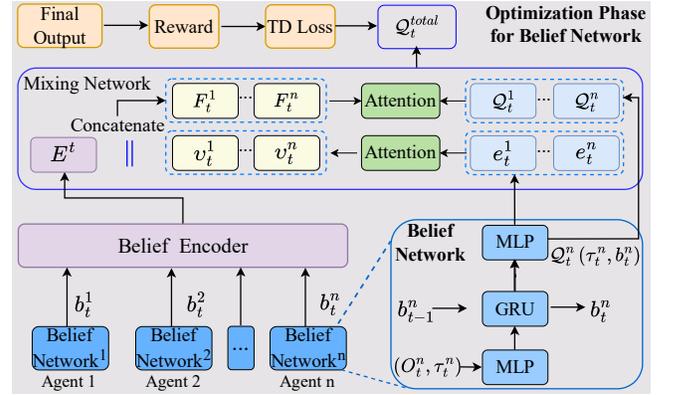


Figure 3: The train process of MALLM.

to jointly integrate individual belief information (captured in  $e_t^i$ ) and collective belief dynamics (captured in  $E_t$ ), thereby facilitating coordinated optimization. Next, the local Q-values  $\{Q_t^i\}_{i=1}^N$  and the transformed features  $\{F_t^i\}_{i=1}^N$  are fed together into multi-head attention layers to compute the global Q-value  $Q_t^{tot}$ . This global Q-value function accounts for local-global interactions, ensuring that improvements in individual behaviors also contribute to overall performance enhancement. To train the mixing network, the following loss function is minimized:

$$\mathcal{L}_{\text{mix}}(\phi) = \mathbb{E}_{\mathcal{D}} \left[ \left( r_{tot} + \gamma \max_{\{e_{t+1}^i\}_{i=1}^N} Q_{tot}^{t+1}(\tau_{t+1}, \{e_{t+1}^i\}_{i=1}^N; \phi') - Q_{tot}^t(\tau_t, \{e_t^i\}_{i=1}^N; \phi) \right)^2 \right] + \lambda_m \sum_{i=1}^N \|Q_i^t - Q_{tot}^t\|^2, \quad (6)$$

the term  $\|Q_i^t - Q_{tot}^t\|^2$  ensures that local Q-values remain consistent with the global estimate. In addition, the target network parameters  $\phi'$  are updated using a soft update rule:  $\phi' \leftarrow \tau \phi + (1 - \tau) \phi'$ .

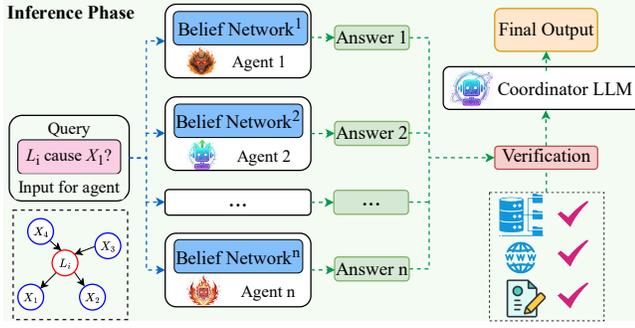


Figure 4: The reasoning process of MALLM.

Through this mechanism, the mixing network can optimize local policies to improve the global objective, thereby promoting stable convergence during training.

**(3) Reward Design.** The reward function  $\mathcal{R}_{\text{design}}$  consists of the following four components. (a) *Action Likelihood Reward*  $r_i^{AL} = \min(R_{\text{max}}, \text{sim}(u_i, C))$  measures the consistency of the final output with the target via cosine similarity  $\text{sim}(u_i, C) = \frac{u_i \cdot C}{\|u_i\| \|C\|}$ , where  $C$  denotes the output of the coordinator and  $u_i$  represents the output of an individual executor. (b) *Uncertainty Reduction Reward*  $r_i^{UR} = \text{Con}_i \cdot \text{sim}(u_i, C)$  assesses the uncertainty of LLMs by considering both the model's direct confidence and the similarity to the overall answer, where  $\text{Con}_i$  represents the confidence of agent  $i$ . (c) *Collaborative Contribution Reward*  $r_i^{CC} = \min(R_{\text{max}}, \text{quality}(u_i, \{u_j\}_{j \neq i}))$  assesses each agent's contribution to the collective solution [50]. (d) *Evidence Reliability Reward*  $r_i^{ER} = \text{CAcc}_i$  measures the reliability of the evidence corresponding to each query result across multiple web data sources, where  $\text{CAcc}$  represents the evidence ratio. For further details, see Appendix A. The total reward is computed as:  $r_i = \alpha_1 r_i^{AL} + \alpha_2 r_i^{UR} + \alpha_3 r_i^{CC} + \alpha_4 r_i^{ER}$ , where  $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$ .

**(4) Inference Phase.** During the inference phase, the coordinator LLM distributes query information containing latent variables to multiple execution LLMs, which independently generate their own answers. By incorporating the retrieved evidence from Section 3.3, the coordinator LLM aggregates both the answers and the evidence to produce the final output, as shown in Figure 4.

**3.2.3 Theoretical Analysis.** In this subsection, we mainly analyze the existence and convergence of BNE.

**(1) Existence of BNE.** To bridge the DEC-POMDP formulation with BNE analysis, we treat each agent  $i$  as a player, where each player forms and updates beliefs about other agents' types based on a common prior and its own observations. Each player's type  $\theta_i$  is determined by their internal beliefs and private observation history. Formally, if there exists a strategy profile  $\{\pi_i^*\}_{i=1}^N$  such that for any player  $i$ :

$$\mathbb{E}_{\theta_{-i}} [U_i(\pi_i^*(\theta_i), \pi_{-i}^*(\theta_{-i}))] \geq \mathbb{E}_{\theta_{-i}} [U_i(\pi_i'(\theta_i), \pi_{-i}^*(\theta_{-i}))], \quad \forall \pi_i', \quad (7)$$

where  $\theta_{-i} = (\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_N)$  denotes the types of all agents except  $i$ . The utility function is defined as:

$$U_i(\pi_i^*, \pi_{-i}^*, \theta_i, \theta_{-i}) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r_i^t | \pi_i^*, \pi_{-i}^*, \theta_i, \theta_{-i} \right], \quad (8)$$

where  $r_i^t = R(s^t, a^t)_i$  denotes the reward obtained by player  $i$  at time  $t$ ,  $N$  denotes the number of players.

**Theorem 3.1.** (Existence of BNE) In our MALLM framework, assuming certain conditions [51] hold, then by Glicksberg's Fixed Point Theorem [2], there exists a BNE strategy profile  $\pi^* = (\pi_1^*, \dots, \pi_N^*)$ . A complete proof is provided in Appendix B.1.

**(2) Convergence of BNE.** We analyze the convergence of the TLVD framework through Bayesian regret. First, to connect the theoretical analysis with the DEC-POMDP formulation, we define the Bayesian regret of each agent  $i$  over  $T$  steps as:

$$R_i(T) = \mathbb{E}_{s_t, \pi_t} \left[ \sum_{t=1}^T (V_i^*(s_t) - V_i^{\pi_t}(s_t)) \right], \quad (9)$$

where the optimal value function under BNE is:

$$V_i^*(s) = \max_{\pi_i} \mathbb{E}_{\pi_{-i}} \left[ \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)_i \mid s_0 = s, \pi_i, \pi_{-i}^* \right], \quad (10)$$

and  $V_i^{\pi_t}(s)$  is the value function under the current strategy profile  $\pi^t = (\pi_1^t, \dots, \pi_N^t)$  at time  $t$ . The expectation accounts for randomness in both state transitions (governed by  $P$ ) and policy selections. The total Bayesian regret is therefore defined as:  $R(T) = \sum_{i=1}^N R_i(T)$ .

Second, we introduce standard assumptions [51] and propose Lemma 3.1, which provides an upper bound on the Bayesian regret. A proof sketch is given below, with more details provided in Appendices B.2, C.1, and C.2.

**Lemma 3.1.** (Performance Difference) For joint policies  $\pi = (\pi_i, \pi_{-i})$  and  $\pi' = (\pi_i', \pi_{-i}')$ , the difference in their value functions satisfies:

$$V_i^{\pi'}(s) - V_i^{\pi}(s) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{\pi'}} \left[ \mathbb{E}_{a \sim \pi'} Q_i^{\pi'}(s, a) - \mathbb{E}_{a \sim \pi} Q_i^{\pi}(s, a) \right], \quad (11)$$

where  $d_{\pi'}$  is the state distribution under policy  $\pi'$ , and  $a = (a_i, a_{-i})$  denotes the joint action from the action space  $\mathcal{A}$ .

Note that the function  $Q_i^{\pi}(s, a)$  in this lemma will be approximated by neural networks in our implementation (see Section 3.2.2 for details). Consistent with existing studies [15, 51], we apply this lemma to our regret analysis and obtain:

$$R(T) = \sum_{i=1}^N \frac{1}{1-\gamma} \mathbb{E}_{s_t, \pi_t} \left[ \sum_{t=1}^T (\mathbb{E}_{a_t \sim \pi'} Q_i^{\pi'}(s_t, a_t^*) - \mathbb{E}_{a_t \sim \pi} Q_i^{\pi}(s_t, a_t)) \right], \quad (12)$$

where  $\pi^*$  denotes the BNE policies. By bounding the suboptimality  $\delta_t$ , we obtain:

$$\mathbb{E}_{a_t \sim \pi'} Q_i^{\pi'}(s_t, a_t^*) - \mathbb{E}_{a_t \sim \pi} Q_i^{\pi}(s_t, a_t) \leq \delta_t, \quad (13)$$

where  $\delta_t = O(1/\sqrt{t})$  bounds policy suboptimality,  $O(\cdot)$  is the asymptotic upper bound. Under standard regularity conditions, these errors can be bounded by constants  $C_{\delta}$ , respectively, which yields:

$$R(T) \leq \sum_{i=1}^N \frac{1}{1-\gamma} C_{\delta} \sum_{t=1}^T \frac{1}{\sqrt{t}} = O\left(\frac{N\sqrt{T}}{1-\gamma}\right). \quad (14)$$

### 3.3 Verification of Latent Variables

Despite the ability of multiple agents in TLVD to infer potential latent variables, further validation is necessary to ensure their plausibility and to achieve the traceability of the proposed method. Therefore, we leverage data sources (e.g., arXiv, Wikipedia, databases,

and patients’ anonymized personal text reports) for verification, and feed the retrieved evidence back to the execution LLMs and the coordinator LLM to update their strategies and beliefs.

### 3.4 Complexity Analysis

The space complexity of our proposed TLVD mainly depends on the belief networks, belief encoder, and mixing network. For the belief networks, the complexity is  $O(Ndd')$ , where  $d$  denotes the input dimension, and  $d'$  denotes the hidden layer dimension; for the belief encoder, the complexity is  $O(Nd'^2)$ ; and for the mixing network, the complexity is  $O(d'^2)$ . Therefore, the space complexity of TLVD can be summarized as  $O(Nd'^2)$ .

## 4 Experiments

Our study primarily focuses on the following research questions: **RQ1**: How does the performance of TLVD compare with existing methods? **RQ2**: What is the impact of model configurations on overall performance? **RQ3**: How does each component of TLVD affect the overall performance? **RQ4**: How do hyperparameter settings affect the performance of TLVD? **RQ5**: How does TLVD operate in real-world examples? **RQ6**: What is the impact of web data sources on TLVD, and what are the main causes of its failures?

### 4.1 Experimental Setup

**4.1.1 Datasets & evaluations.** In this section, we evaluate our method on five real-world datasets. As shown in Table 1, we utilized the de-identified WCHSU-Cancer and WCHSU-Pain datasets from real hospitals scenarios. Additionally, we also use two generic domain benchmark datasets (Multitasking Behaviour Study [19] and Teacher’s Burnout Study [5]). *Please note that the WCHSU dataset is newly collected and has not been made publicly available online, making it unlikely to have been used in training any existing LLMs, and therefore posing no risk of data leakage.* The evaluation metrics are ACC, CAcc, and ECit. Detailed descriptions can be found in Appendix A. For each dataset, we conduct five experiments with different random seeds and report the average performance.

**4.1.2 Baselines.** We compare four types of baselines: single LLMs, deep research agents, multi-agent platforms, and multi-LLM reasoning frameworks. The single LLMs mainly include GPT-5. The deep research agents include WideSearch [47], Gemini-deepresearch, Openai-deepresearch, Qwen-deepresearch, and Doubao-deepresearch. The multi-agent platforms include Autogen [48] and MiniMax, whereas the multi-LLM reasoning frameworks include CAMEL [27], Multiagent (Majority) [28], and Multiagent (Debate) [13].

**4.1.3 Implementation details.** In this section, the TLVD framework consists of one coordinator and two executor LLMs. During training, the episodes per task is set to 100, buffer size  $|\mathcal{D}|$  to 32, optimizer to Adam, learning rate to 0.001, discount factor to 0.99, entity dimension to 256, belief state dimension to 128, MLP hidden size to 256, and  $\lambda_m$  to 0.1.  $R_{max}$  is 1. For the early stopping mechanism, we utilize the same termination threshold settings as in previous studies [51]. To ensure a fair comparison with baseline methods, we employed three identical models across these LLMs. For heterogeneous results, we also evaluated TLVD using different models, as shown in Table 4. All evaluations were conducted under a zero-shot

**Table 1: Statistics of datasets.**

Dataset	#Domain	#Sample	#Number of observed variables	#Number of latent variables
<i>WCHSU-Cancer</i>	Medical	200,000	12/22	6/4
<i>WCHSU-Pain</i>	Medical	1,568	16	4
<i>Multitasking Behaviour Study</i>	Social Science	202	9	4
<i>Teacher’s Burnout Study</i>	Social Science	599	32	11

setting on a server equipped with an NVIDIA GeForce A6000 GPU with 48GB of memory. For the LLMs used to validate latent variables in web data sources, we employed the same LLMs as those used for the executors. For each causal validation query, we retrieved the top five most relevant texts from arXiv and Wikipedia each time.

### 4.2 RQ1: Comparisons and Analysis

Table 2 shows a performance comparison of different methods on the three WCHSU datasets. The results indicate that TLVD outperforms all baseline methods across all evaluation metrics. On average, in terms of ACC, TLVD achieves a 131.68% improvement over MiniMax, a 124.30% improvement over Autogen, and a 58.07% improvement over OpenAI-deepresearch. Moreover, when we constructed new datasets by randomly sampling the original data (results in Table 3), TLVD still maintained the top performance: It outperformed the runner-up, OpenAI-deepresearch, with average improvement of 32.96% in ACC, 239.72% in CAcc, and 52.86% in ECit. Notably, TLVD achieves performance improvements with fewer communication tokens compared to CAMEL, Multi-Agent Debate, and Multi-Agent Majority, which can be attributed to the design of our belief network (see Figure 10 in Appendix D).

### 4.3 RQ2: Model Configuration Analysis

To evaluate the impact of the coordinator LLM and execution LLM performance on the TLVD, and to investigate whether heterogeneous execution LLMs can also achieve a BNE, we conducted two types of experiments: one pairing a strong coordinator LLM with weaker execution LLMs, and another pairing a weaker coordinator LLM with stronger execution LLMs. These experiments were further divided into homogeneous and heterogeneous execution groups for detailed analysis. To ensure a fair comparison, the coordinator LLM was consistently set to Kimi k2 32B across all experiments. For the heterogeneous execution group, we used the following configurations: LLaMA3.1 8B and Qwen2.5 7B; as well as another configuration consisting of Qwen 2.5 72B and LLaMA3.1 70B. For the homogeneous execution group, two configurations were tested: one with two weak models (LLaMA 3.1 8B) and another with two strong models (LLaMA 3.1 70B). As shown in Table 4, stronger execution LLMs achieve BNE more efficiently by providing higher-quality answers. Additionally, heterogeneous models perform worse than homogeneous models due to the increased difficulty in reaching BNE.

### 4.4 RQ3: Ablation Studies

To evaluate the effectiveness of different components in TLVD, we design the following variants: **TLVD-v**: Removes the multi-agent

**Table 2: Performance evaluation of different models.**

Methods	WCHSU-Cancer ( $n = 12$ )			WCHSU-Cancer ( $n = 22$ )			WCHSU-Pain		
	ACC	CAcc	ECit	ACC	CAcc	ECit	ACC	CAcc	ECit
GPT5	0.134±0.067	0.100±0.094	0.031±0.029	0.150±0.122	0.116±0.039	0.079±0.027	0.150±0.122	0.095±0.061	0.095±0.061
Gemini-deepresearch	0.267±0.081	0.286±0.046	0.154±0.024	0.300±0.100	0.347±0.042	0.236±0.029	0.250±0.000	0.338±0.031	0.284±0.026
OpenAI-deepresearch	0.433±0.082	0.560±0.049	0.215±0.019	0.550±0.100	0.214±0.165	0.214±0.089	0.550±0.100	0.253±0.050	0.200±0.039
Qwen-deepresearch	0.200±0.066	0.500±0.000	0.077±0.000	0.250±0.000	0.271±0.029	0.271±0.029	0.300±0.100	0.500±0.158	0.105±0.033
Doubao-deepresearch	0.000±0.000	0.150±0.050	0.038±0.070	0.000±0.000	0.000±0.000	0.000±0.000	0.150±0.122	0.000±0.000	0.000±0.000
Autogen	0.300±0.125	0.000±0.000	0.000±0.000	0.450±0.100	0.255±0.036	0.200±0.029	0.350±0.122	0.089±0.044	0.084±0.042
MiniMax	0.467±0.067	0.600±0.000	0.231±0.000	0.300±0.100	0.192±0.030	0.171±0.027	0.300±0.100	0.071±0.014	0.053±0.008
CAMEL	0.267±0.081	0.000±0.000	0.000±0.000	0.400±0.122	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000
MULTI-AGENT-MAJORITY	0.134±0.067	0.000±0.000	0.000±0.000	0.200±0.100	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000
MULTI-AGENT DEBATE	0.234±0.081	0.000±0.000	0.000±0.000	0.350±0.122	0.000±0.000	0.000±0.000	0.200±0.100	0.000±0.000	0.000±0.000
TLVD	<b>0.833±0.000</b>	<b>0.900±0.033</b>	<b>0.415±0.015</b>	<b>0.750±0.000</b>	<b>0.920±0.040</b>	<b>0.329±0.014</b>	<b>0.800±0.100</b>	<b>0.860±0.049</b>	<b>0.453±0.026</b>

**Table 3: Performance evaluation of different models, where we randomly sample half of the original dataset.**

Methods	WCHSU ( $n = 12$ )			WCHSU ( $n = 22$ )			WCHSU-Pain		
	ACC	CAcc	ECit	ACC	CAcc	ECit	ACC	CAcc	ECit
GPT5	0.120±0.098	0.074±0.042	0.074±0.042	0.000±0.000	0.111±0.070	0.033±0.021	0.133±0.163	0.51±0.0037	0.36±0.0056
Gemini-deepresearch	0.240±0.013	0.084±0.042	0.084±0.042	0.200±0.163	0.160±0.013	0.160±0.013	0.266±0.133	0.000±0.000	0.000±0.000
OpenAI-deepresearch	0.600±0.126	0.388±0.029	0.347±0.026	0.667±0.000	0.180±0.016	0.180±0.016	0.600±0.134	0.318±0.047	0.318±0.047
Qwen-deepresearch	0.440±0.080	0.225±0.094	0.095±0.039	0.200±0.163	0.289±0.089	0.087±0.027	0.266±0.133	0.000±0.000	0.000±0.000
Doubao-deepresearch	0.000±0.000	0.000±0.000	0.000±0.000	0.133±0.163	0.000±0.000	0.000±0.000	0.067±0.133	0.000±0.000	0.000±0.000
Autogen	0.480±0.098	0.000±0.000	0.000±0.000	0.333±0.000	0.186±0.042	0.173±0.039	0.400±0.250	0.000±0.000	0.000±0.000
MiniMax	0.560±0.080	0.000±0.000	0.000±0.000	0.533±0.164	0.400±0.122	0.053±0.016	0.467±0.267	0.350±0.062	0.247±0.044
CAMEL	0.360±0.080	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000	0.266±0.133	0.000±0.000	0.000±0.000
MULTI-AGENT-MAJORITY	0.160±0.080	0.000±0.000	0.000±0.000	0.133±0.163	0.000±0.000	0.000±0.000	0.067±0.133	0.000±0.000	0.000±0.000
MULTI-AGENT DEBATE	0.320±0.098	0.000±0.000	0.000±0.000	0.333±0.211	0.175±0.061	0.047±0.016	0.266±0.133	0.000±0.000	0.000±0.000
TLVD	<b>0.800±0.126</b>	<b>0.886±0.057</b>	<b>0.467±0.112</b>	<b>0.734±0.133</b>	<b>0.914±0.070</b>	<b>0.350±0.064</b>	<b>0.933±0.133</b>	<b>0.900±0.082</b>	<b>0.412±0.043</b>

LLM collaboration module and uses only a voting strategy. **TLVD-d**: Removes the multi-agent LLM collaboration module and uses only a debate strategy. **TLVD-I**: Uses a single LLM to generate latent variables and their semantics. **TLVD-R1**: Uses only the Action Likelihood Reward. **TLVD-R2**: Considers both the Action Likelihood Reward and the Uncertainty Reduction Reward. **TLVD-R3**: Considers the Action Likelihood Reward, the Uncertainty Reduction Reward, and the Collaborative Contribution Reward. We conducted ablation experiments using WCHSU-Cancer ( $n=12$ ) and WCHSU-Pain as examples. Additionally, we analyzed the impact of different LLM types on the results.

As shown in Figure 8 and Table 7 (see Appendix D), we have the following observations: (i) TLVD achieves the best performance when all components are included. (ii) The Uncertainty Reduction Reward and the Collaborative Contribution Reward have a significant impact on model performance. (iii) There are noticeable performance differences among different LLMs, with GPT-oss-120B and DeepSeek-v3 performing relatively better.

#### 4.5 RQ4: Parameter Sensitivity

In the WCHSU-Cancer ( $n = 12$ ) and WCHSU-Pain datasets, we further investigate the impact of the number of execution LLMs  $N$

**Table 4: Performance of different configurations.**

Methods	WCHSU-Cancer ( $n = 12$ )			WCHSU-Pain		
	ACC	CAcc	ECit	ACC	CAcc	ECit
Homo. (2× LLaMA3.1 8B)	0.500	1.000	0.115	0.500	1.000	0.474
Homo. (2× LLaMA3.1 70B)	0.833	1.000	0.462	0.850	1.000	0.737
Hetero. (LLaMA3.1 8B, Qwen2.5 7B)	0.333	1.000	0.346	0.250	1.000	0.421
Hetero. (LLaMA3.1 70B, Qwen2.5 72B)	0.500	1.000	0.346	0.750	1.000	0.632
TLVD	0.833	0.900	0.415	0.800	0.886	0.453
LLAMA-3.1-8B (Zero-shot)	0.000	0.000	0.000	0.250	0.191	0.057

and the discount factor  $\gamma$  on model performance. As shown in Figure 9, the results indicate that when the number of execution LLMs exceeds six, performance gains become very limited, and in some cases, performance even declines. When using weaker LLMs (e.g., LLaMA 3.1 8B), the number of execution LLMs has little impact on model performance. In contrast, with stronger LLMs (e.g., LLaMA 3.1 70B), increasing the number of execution LLMs yields some performance improvements. We attribute this to the challenges faced by the coordinator LLM when managing too many execution LLMs, making it difficult to achieve effective coordination through information from additional agents. Moreover, as the discount factor  $\gamma$  increases, model performance shows an upward trend.

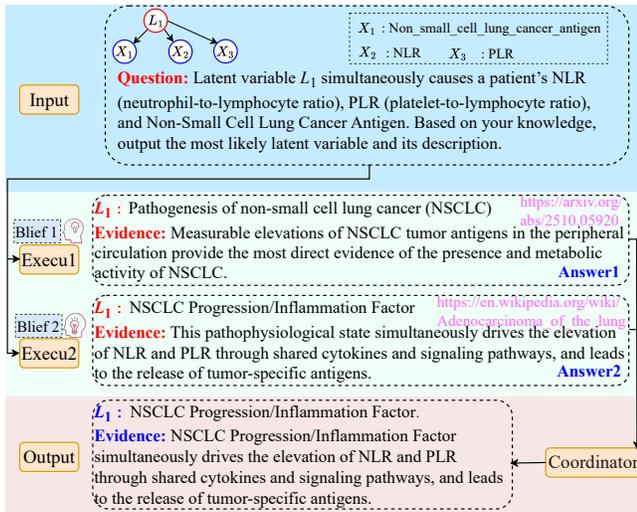


Figure 5: Case study.

## 4.6 RQ5: Case Studies

We take the specific latent variable  $L_1$  from the WCHSU-Cancer ( $n=22$ ) dataset as an example to demonstrate the latent variables identified by our method and the supporting evidence. As shown in Figure 5, Executors 1 and 2 output  $L_1$  and its semantics based on their own beliefs and observations. Through the latent variable verification process, retrieved information is returned. The coordinator aggregates all information to infer the final latent variables and simultaneously outputs the corresponding causal evidence.

## 4.7 RQ6: Other In-Depth Analysis

**4.7.1 Impact of Data Sources.** We further analyzed the impact of different data sources on TLVD to validate its applicability under varying conditions. We explored three scenarios on the WCHSU-Pain dataset: the W/O ARR model excludes literature from arXiv, the W/O WIKI model excludes literature from WIKIPEDIA, and the W/O DB model excludes medical-related knowledge graphs. As shown in Figure 6, we observe that articles on arXiv contain more medically relevant causal information, which aligns well with reality and is consistent with existing studies [8]. The influence of the databases on our model is relatively small, which is reasonable since the data we use contains knowledge that is not publicly available and is rarely stored or used in existing databases.

**4.7.2 Error Analysis.** To further dissect our approach, we employed the MAST taxonomy [6] to attribute failures in multi-agent systems. Through the analysis of 100 execution traces, we computed the frequency of occurrence for each failure mode and category. Detailed definitions and examples for each failure mode can be found in existing studies [6]. As shown in Figure 7, the primary causes of multi-agent system failures are Specification Issues and Task Verification. For Specification Issues, failures stem from system design decisions and incomplete or ambiguous prompts, mainly including vague task instructions and unclear role definitions. For Task Verification, failures are primarily related to output quality control, such as insufficient verification or premature task termination.

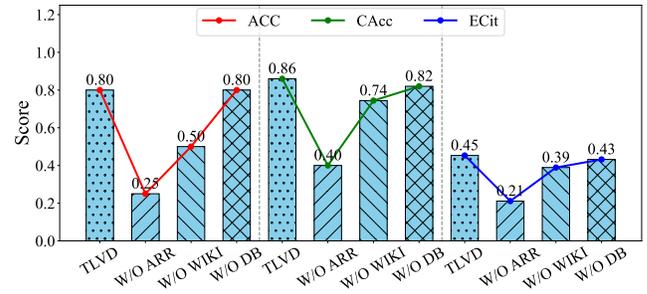


Figure 6: Impact of data sources on model performance.

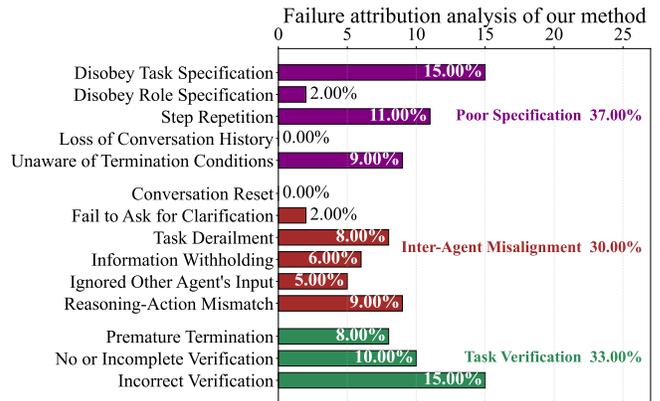


Figure 7: Failure Attribution of Our Method. Note that the vertical axis represents specific failure modes, and the horizontal axis represents their respective proportions.

## 5 Conclusion

In this work, we propose TLVD, a multi-LLM collaborative framework for traceable latent variable discovery. TLVD first constructs a hierarchical coordination mechanism for multi-agent LLMs using game theory and reinforcement learning, enabling multiple execution LLMs to perform distributed reasoning under the guidance of a coordinator LLM to infer latent variables and their semantic meanings. Subsequently, the framework conducts causal validation of the inferred latent variables using multiple real-world data sources. Finally, we conduct extensive experiments on three real hospital datasets that we constructed, as well as two benchmark datasets, demonstrating the effectiveness of the TLVD in latent variable discovery.

## 6 Acknowledgements

This research is partially supported by funding from Xiangjiang Laboratory (25XJ02002), the National Natural Science Foundation of China (62376228, 62376227), the Science and Technology Innovation Program of Hunan Province (2024RC4008), the China Postdoctoral Science Foundation (2025M770766), Sichuan Provincial Postdoctoral Research Project Special Funding (TB2025043) and Sichuan Science and Technology Program (2023NSFSC0032). Carl Yang is not supported by any funds from China.

## References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv:2303.08774* (2023).
- [2] Jamshaid Ahmad, Abdullah Eqal Al-Mazrooei, and Themistocles M Rassias. 2023. Common fixed point theorems with applications to theoretical computer science. *International Journal of Nonlinear Analysis and Applications* 14, 2 (2023), 1–10.
- [3] Sina Akbari, Ehsan Mokhtarian, AmirEmad Ghassami, and Negar Kiyavash. 2021. Recursive causal structure learning in the presence of latent variables and selection bias. *NeurIPS* 34 (2021), 10119–10130.
- [4] Animashree Anandkumar, Daniel Hsu, Adel Javanmard, and Sham Kakade. 2013. Learning linear bayesian networks with latent variables. In *ICML*. 249–257.
- [5] Barbara M Byrne. 2013. *Structural equation modeling with Mplus: Basic concepts, applications, and programming*. routledge.
- [6] Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. 2025. Why do multi-agent llm systems fail? *arXiv preprint arXiv:2503.13657* (2025).
- [7] Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024. ReConcile: Round-Table Conference Improves Reasoning via Consensus among Diverse LLMs. In *ACL*. 7066–7085.
- [8] Xuanzhong Chen, Xiaohao Mao, Qihan Guo, Lun Wang, Shuyang Zhang, and Ting Chen. 2024. RareBench: can LLMs serve as rare diseases specialists?. In *KDD*. 4850–4861.
- [9] Zhengming Chen, Feng Xie, Jie Qiao, Zhifeng Hao, Kun Zhang, and Ruichu Cai. 2022. Identification of linear latent variable model with arbitrary distribution. In *AAAI*, Vol. 36. 6350–6357.
- [10] David Maxwell Chickering. 2002. Optimal structure identification with greedy search. *Journal of machine learning research* 3, Nov (2002), 507–554.
- [11] Xinshuai Dong, Biwei Huang, Ignavier Ng, Xiangchen Song, Yujia Zheng, Songyao Jin, Roberto Legaspi, Peter Spirtes, and Kun Zhang. 2024. A Versatile Causal Discovery Framework to Allow Causally-Related Hidden Variables. In *ICLR*.
- [12] Huaming Du, Yujia Zheng, Baoyu Jing, Yu Zhao, Gang Kou, Guisong Liu, Tao Gu, Weimin Li, and Carl Yang. 2025. Causal Discovery through Synergizing Large Language Model and Data-Driven Reasoning. In *KDD*. 543–554.
- [13] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *ICML*.
- [14] Chris Frith and Uta Frith. 2005. Theory of mind. *Current biology* 15, 17 (2005), R644–R645.
- [15] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *ICML*. PMLR, 1587–1596.
- [16] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirog Ma, Xiao Bi, et al. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature* 645, 8081 (2025), 633–638.
- [17] Thilo Hagendorff, Sarah Fabi, and Michal Kosinski. 2023. Human-like intuitive behavior and reasoning biases emerged in large language models but disappeared in ChatGPT. *Nature Computational Science* 3, 10 (2023), 833–838.
- [18] Elad Hazan et al. 2016. Introduction to online convex optimization. *Foundations and Trends® in Optimization* 2, 3-4 (2016), 157–325.
- [19] Samsad Afrin Himi, Markus Böhner, Matthias Schwaighofer, Anna Klapetek, and Sven Hilbert. 2019. Multitasking behavior and its related constructs: Executive functions, working memory capacity, relational integration, and divided attention. *Cognition* 189 (2019), 275–298.
- [20] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *ICLR*.
- [21] Biwei Huang, Charles Jia Han Low, Feng Xie, Clark Glymour, and Kun Zhang. 2022. Latent hierarchical causal structure discovery with rank constraints. *Advances in neural information processing systems* 35 (2022), 5549–5561.
- [22] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM TOIS* 43, 2 (2025), 1–55.
- [23] Bohdan Kivva, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. 2021. Learning latent causal graphs via mixture oracles. *NeurIPS* 34 (2021), 18087–18101.
- [24] Erich Kummerfeld and Joseph Ramsey. 2016. Causal clustering for 1-factor measurement models. In *KDD*. 1655–1664.
- [25] Hao Duong Le, Xin Xia, and Zhang Chen. 2024. Multi-agent causal discovery using large language models. *arXiv preprint arXiv:2407.15073* (2024).
- [26] Ao Li, Yuexiang Xie, Songze Li, Fugee Tsung, Bolin Ding, and Yaliang Li. 2025. Agent-Oriented Planning in Multi-Agent Systems. In *ICLR*.
- [27] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for “mind” exploration of large language model society. *NeurIPS* 36 (2023), 51991–52008.
- [28] Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024. More Agents Is All You Need. *Transactions on Machine Learning Research* (2024).
- [29] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate. In *EMNLP*. 17889–17904.
- [30] Yunhao Liang, Yuan Qu, Jingyuan Yang, Shaochong Lin, and Zuo-Jun Max Shen. 2025. Everyone Contributes! Incentivizing Strategic Cooperation in Multi-LLM Systems via Sequential Public Goods Games. *arXiv:2508.02076* (2025).
- [31] Tongxuan Liu, Xingyu Wang, Weizhe Huang, Wenjiang Xu, Yuting Zeng, Lei Jiang, Hailong Yang, and Jing Li. 2024. Groupdebate: Enhancing the efficiency of multi-agent debate using group discussion. *arXiv:2409.14051* (2024).
- [32] Weigang Lu, Ziyu Guan, Wei Zhao, Yaming Yang, and Long Jin. 2024. Nodemixup: Tackling under-reaching for graph neural networks. In *AAAI*. 14175–14183.
- [33] Weigang Lu, Yibing Zhan, Binbin Lin, Ziyu Guan, Liu Liu, Baosheng Yu, Wei Zhao, Yaming Yang, and Dacheng Tao. 2024. SkipNode: On alleviating performance degradation for deep graph convolutional networks. *TKDE* (2024), 7030–7043.
- [34] Ehsan Mokhtarian, Sepehr Elahi, Sina Akbari, and Negar Kiyavash. 2025. Recursive causal discovery. *JMLR* 26, 61 (2025), 1–65.
- [35] Ignavier Ng, Xinshuai Dong, Haoyue Dai, Biwei Huang, Peter Spirtes, and Kun Zhang. 2024. Score-based causal discovery of latent variable causal models. In *Forty-first International Conference on Machine Learning*.
- [36] J Pearl. 2009. *Causality: models, reasoning, and inference*. 2nd edn Cambridge University Press. New York (2009).
- [37] Judea Pearl. 2019. The seven tools of causal inference, with reflections on machine learning. *Commun. ACM* 62, 3 (2019), 54–60.
- [38] Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. 2014. Causal discovery with continuous additive noise models. *The Journal of Machine Learning Research* 15, 1 (2014), 2009–2053.
- [39] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2024. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. In *ICLR*.
- [40] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *ICML*. 4295–4304.
- [41] Shuo Ren, Pu Jian, Zhenjiang Ren, Chunlin Leng, Can Xie, and Jiajun Zhang. 2025. Towards scientific intelligence: A survey of llm-based scientific agents. *arXiv preprint arXiv:2503.24047* (2025).
- [42] Yangjun Ruan, Chris J Maddison, and Tatsunori B Hashimoto. 2024. Observational scaling laws and the predictability of language model performance. *Advances in Neural Information Processing Systems* 37 (2024), 15841–15892.
- [43] Saber Salehkaleybar, AmirEmad Ghassami, Negar Kiyavash, and Kun Zhang. 2020. Learning linear non-Gaussian causal models in the presence of latent variables. *Journal of Machine Learning Research* 21, 39 (2020), 1–24.
- [44] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. 2006. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research* 7, 10 (2006).
- [45] Peter Spirtes, Clark N Glymour, and Richard Scheines. 2000. *Causation, prediction, and search*. MIT press.
- [46] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science* 18, 6 (2024), 186345.
- [47] Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, et al. 2025. WideSearch: Benchmarking Agentic Broad Info-Seeking. *arXiv preprint arXiv:2508.07999* (2025).
- [48] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *COLM*.
- [49] Feng Xie, Ruichu Cai, Biwei Huang, Clark Glymour, Zhifeng Hao, and Kun Zhang. 2020. Generalized independent noise condition for estimating latent variable causal graphs. *NeurIPS* 33 (2020), 14891–14902.
- [50] Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Xie. 2023. Self-evaluation guided beam search for reasoning. *NeurIPS* 36 (2023), 41618–41650.
- [51] Xie Yi, Zhanke Zhou, Chentao Cao, Qiyu Niu, Tongliang Liu, and Bo Han. 2025. From Debate to Equilibrium: Belief-Driven Multi-Agent LLM Reasoning via Bayesian Nash Equilibrium. In *ICML*.
- [52] Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. 2025. G-Memory: Tracing Hierarchical Memory for Multi-Agent Systems. In *NeurIPS*.
- [53] Wanjia Zhao, Mert Yuksekogun, Shirley Wu, and James Zou. 2025. Sirius: Self-improving multi-agent systems via bootstrapped reasoning. *arXiv:2502.04780* (2025).
- [54] Lecheng Zheng, Zhengzhang Chen, Jingrui He, and Haifeng Chen. 2024. MULAN: multi-modal causal structure learning and root cause analysis for microservice

systems. In *WWW*. 4107–4116.

[55] Yaochen Zhu, Yinhan He, Jing Ma, Mengxuan Hu, Sheng Li, and Jundong Li. 2024. Causal inference with latent variables: Recent advances and future perspectives. In *KDD*. 6677–6687.

## A Reproducibility

In this section, we provide more details about the datasets and evaluation metrics to facilitate the reproducibility of the results. Our code is available at <https://github.com/HYJ9999/TLVD.git>.

**WCHSU-Cancer:** The WCHSU-Cancer dataset comes from the health management center of a large hospital in Asia, focusing on early screening for lung cancer, and includes health check-up data from a total of 200,000 participants. The original dataset contains 230 variables. WCHSU-Cancer (n=22) was curated by scoring the correlation between the original variables and the lung cancer variable of interest using LLMs, with scores ranging from 0 to 5. After filtering based on scores greater than 5, 22 relevant variables were selected. WCHSU-Cancer (n=12), on the other hand, was screened directly by lung cancer specialists from the same hospital.

**WCHSU-Pain:** The WCHSU-Pain comes from the same large hospital in Asia and focuses on postoperative pain in patients. It includes perioperative and postoperative analgesic usage data from a total of 1,568 patients. The dataset contains 16 variables. Please noted that all datasets used were properly pre-processed (including the de-identification of all patient information) and IRB-approved.

Next, we present the detailed computation of the evaluation metrics: (1) ACC represents the proportion of latent variables correctly predicted by the model. For the WCHSU dataset which lacks ground-truth, we invited five experts from relevant fields at Asia’s largest hospital to perform a consistency evaluation of the latent variables inferred by the model. Specifically, five experts assessed the latent variables and the authenticity of the evidence based on their professional experience; the consensus reached through consultation was adopted as the ground truth for this experiment. (2)  $CACc = \frac{n}{EG}$ ,  $n$  is the number of edges with real evidence, and  $EG$  is the number of edges for which evidence is found. (3)  $ECit = \frac{n}{EA}$ ,  $EA$  is the number of edges in the causal graph where at least one of the nodes is a latent variable.

## B Theoretical Proof

### B.1 Proof of Theorem 3.1

**PROOF.** We aim to prove the existence of a Bayesian Nash Equilibrium (BNE) in our multi-agent LLM framework under the specified conditions. Following existing studies, our proof primarily proceeds by verifying the conditions of Glicksberg’s Fixed Point Theorem, which guarantees the existence of a fixed point in continuous games with finite-dimensional strategy spaces.

**Step 1: Define the Best Response Correspondence** For each agent  $i$ , define the best response correspondence  $BR_i$  as:

$$BR_i(\pi_{-i}) = \left\{ \pi_i \in \Pi_i \mid \max_{\pi_i} U_i(\theta_i, \pi_i, \pi_{-i}) \right\},$$

where  $\Pi_i$  is the set of all admissible strategies for agent  $i$ , and  $\pi_{-i}$  denotes the strategies of all other agents.

**Step 2: Verify the Conditions of Glicksberg’s Fixed Point Theorem** To apply Glicksberg’s theorem, we need to verify the following conditions for each agent  $i$ :

(i) **Compactness and Convexity of Strategy Space:** The strategy space  $\Pi_i$  consists of all measurable functions from the type space  $\Theta_i$  to the action space  $A_i$ . By Tychonoff’s theorem, since  $\Theta_i$  and  $A_i$  are compact metric spaces,  $\Pi_i$  is also compact. Convexity follows similarly. (ii) **Continuity of the Payoff Function:** For fixed  $\theta_i$ , since  $U_i$  is continuous in actions and strategies map continuously from types to actions, the composition  $U_i(\theta_i, \pi_i(\theta_i), \pi_{-i}(\theta_{-i}))$  is continuous in  $(\pi_i, \pi_{-i})$ . (iii) **Quasi-Concavity of the Payoff Function:** For each  $\theta_i$  and  $\pi_{-i}$ , the mapping  $\pi_i \mapsto U_i(\theta_i, \pi_i, \pi_{-i})$  is linear in the space of mixed strategies, ensuring quasi-concavity.

**Step 3: Establish Upper Hemicontinuity and Non-Empty, Convex-Valuedness of the Best Response Correspondence** We need to show that  $BR_i(\pi_{-i})$  is upper hemicontinuous with non-empty, convex values.

(i) **Non-Empty and Convex Values:** The payoff function  $U_i$  is continuous and quasi-concave in  $\pi_i$ . By the Weierstrass Theorem, the extreme value must exist (and thus be non-empty); convexity follows from the quasi-concavity of  $U_i$ . (ii) **Upper Hemicontinuity** According to Berge’s Maximum Theorem, if: (1) The constraint set  $\Pi_i$  is compact. (2) The objective function  $U_i$  is continuous in  $(\pi_i, \pi_{-i})$ , then the best response correspondence  $BR_i(\pi_{-i})$  is guaranteed to be upper hemicontinuous.

**Step 4: Application of Glicksberg’s Fixed Point Theorem** After verifying the above conditions, applying Glicksberg’s Fixed Point Theorem in our framework leads to the following conclusion: if each player’s strategy set is compact and convex, and their payoff functions are continuous and quasi-concave in their own strategies, then the game has at least one mixed-strategy Nash equilibrium. That is, there exists a strategy profile  $\pi^* = (\pi_1^*, \pi_2^*, \dots, \pi_N^*)$  such that for each agent  $i$ :  $\pi_i^* \in BR_i(\pi_{-i}^*)$ , meaning that, given their beliefs about the types and strategies of other agents, no agent has an incentive to unilaterally deviate to improve their expected payoff. This strategy profile constitutes a Bayesian Nash Equilibrium in our TVLD framework.  $\square$

### B.2 Proof of Convergence to BNE

**PROOF.** We aim to show that by minimizing the TD loss of each agent’s Q-network, the agents’ strategies converge to a BNE.

**Core Assumptions:**

1. The Q-network  $Q_i(s, a_i; \theta_i)$  is parameterized by prompt embeddings  $\theta_i$ , and the mapping from  $\theta_i$  to  $Q_i$  is continuously differentiable.
2. The exploration strategy ensures sufficient coverage of the state-action space.
3. The loss function  $L_i(\theta_i)$  is convex, or its gradient is Lipschitz continuous with respect to  $\theta_i$ .
4. The gradient  $\nabla_{\theta_i} L_i(\theta_i)$  is Lipschitz continuous.
5. The learning rate  $\eta_t$  satisfies the Robbins–Monro conditions:  $\sum_{t=1}^{\infty} \eta_t = \infty$  and  $\sum_{t=1}^{\infty} \eta_t^2 < \infty$ .

**Step 1: Defining the TD Loss Function** For agent  $i$ , the TD loss is:

$$L_i(\theta_i) = \mathbb{E}_{(s, a_i, r_i, s') \sim D_i} \left[ \left( r_i + \gamma \max_{a'_i} Q_i(s', a'_i; \theta_i^-) - Q_i(s, a_i; \theta_i) \right)^2 \right],$$

which measures the discrepancy between the predicted Q-value and the target Q-value based on the reward and estimated optimal future Q-value.

**Step 2: Convergence of Gradient Descent with TD Loss** Under Assumptions 1–4, stochastic gradient descent converges almost surely to a stationary point  $\theta_i^*$  of  $L_i(\theta_i)$ :

$$\lim_{t \rightarrow \infty} \theta_i^{(t)} = \theta_i^* \quad \text{and} \quad \nabla_{\theta_i} L_i(\theta_i^*) = 0.$$

This follows from standard stochastic approximation theory. The gradient condition implies:

$$\mathbb{E}_{(s, a_i, r_i, s') \sim D_i} \left[ (r_i + \gamma \max_{a'_i} Q_i(s', a'_i; \theta_i^-) - Q_i(s, a_i; \theta_i^*)) \cdot \nabla_{\theta_i} Q_i(s, a_i; \theta_i^*) \right] = 0$$

If the function approximator is sufficiently expressive, this implies that the TD error vanishes in expectation:

$$Q_i(s, a_i; \theta_i^*) = \mathbb{E} \left[ r_i + \gamma \max_{a'_i} Q_i(s', a'_i; \theta_i^*) \mid s, a_i \right].$$

Thus,  $Q_i(\cdot; \theta_i^*)$  satisfies the Bellman optimality equation. This ensures that the agent's policy  $\pi_i(a_i | s; \theta_i^*)$  is a best response to the current policies of the other agents, as it maximizes expected cumulative rewards.

**Step 3: Establishing Bayesian Nash Equilibrium** Since each agent's policy is a best response to others, the set of policies  $\pi_i^*$  constitutes a BNE. At this equilibrium, each agent maximizes its expected utility given its beliefs about other agents' types and strategies, thus fulfilling the definition of BNE.  $\square$

## C Detailed Proofs

### C.1 Proof of Lemma 3.1

PROOF. Consider the value functions under policies  $\pi'$  and  $\pi$ :

$$\begin{cases} V_i^{\pi'}(s) = \mathbb{E}_{\pi'} \left[ \sum_{k=0}^{\infty} \gamma^k r_i(s_k, a_k) \mid s_0 = s \right], \\ V_i^{\pi}(s) = \mathbb{E}_{\pi} \left[ \sum_{k=0}^{\infty} \gamma^k r_i(s_k, a_k) \mid s_0 = s \right]. \end{cases}$$

Subtracting the two equations yields:

$$\begin{aligned} V_{\pi'}(s) - V_{\pi}(s) &= \mathbb{E}_{\pi'} \left[ \sum_{k=0}^{\infty} \gamma^k r_i(s_k, a_k) \right] - \mathbb{E}_{\pi} \left[ \sum_{k=0}^{\infty} \gamma^k r_i(s_k, a_k) \right] \\ &= \sum_{k=0}^{\infty} \gamma^k \left( \mathbb{E}_{s_k \sim d_k^{\pi'}} [r_i(s_k, a_k)] - \mathbb{E}_{s_k \sim d_k^{\pi}} [r_i(s_k, a_k)] \right). \end{aligned}$$

Assuming the difference in state distributions is negligible, we focus on action differences. Using the Q-function definition:

$$Q_{\pi}(s, a_i, a_{-i}) = r_i(s, a_i, a_{-i}) + \gamma \mathbb{E}_{s' \sim P} [V_{\pi}(s')],$$

we can write:

$$V_{\pi'}(s) - V_{\pi}(s) = \sum_{k=0}^{\infty} \gamma^k \mathbb{E}_{s_k \sim d_k^{\pi'}} [Q_{\pi}(s_k, a'_k) - V_{\pi}(s_k)].$$

Since  $V_{\pi}(s_k) = \mathbb{E}_{a_k \sim \pi(s_k)} [Q_{\pi}(s_k, a_k)]$ , we have:

$$\begin{aligned} V_{\pi'}(s) - V_{\pi}(s) &= \sum_{k=0}^{\infty} \gamma^k \mathbb{E}_{s_k \sim d_k^{\pi'}} \left[ \mathbb{E}_{a'_k \sim \pi'(s_k)} [Q_{\pi}(s_k, a'_k)] \right. \\ &\quad \left. - \mathbb{E}_{a_k \sim \pi(s_k)} [Q_{\pi}(s_k, a_k)] \right]. \end{aligned}$$

Switching the order of expectations and summing over  $k$ , we get:

$$V_{\pi'}(s) - V_{\pi}(s) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{\pi'}} [Q_{\pi}(s, a'_i, a'_{-i}) - Q_{\pi}(s, a_i, a_{-i})]. \quad \square$$

### C.2 Bounding the Bayesian Regret

To prove the regret bound in our framework is  $R(T) = O\left(\frac{N\sqrt{T}}{1-\gamma}\right)$ , we employ the given lemma:

$$V_i^*(s_t) - V_i^{\pi_t}(s_t) = \frac{1}{1-\gamma} \mathbb{E}_{a_i^{*t}, a_{-i}^{*t}, a_i^t, a_{-i}^t} [Q_i^{\pi_t}(s_t, a_i^{*t}, a_{-i}^{*t}) - Q_i^{\pi_t}(s_t, a_i^t, a_{-i}^t)],$$

and combine it with the convergence properties of policy gradient methods. The detailed derivation is as follows:

**C.2.1 Advantage Function Bound.** From the convergence theory of policy gradient methods (based on online convex optimization) [18], for each agent  $i$ , state  $s_t$  and optimal action  $a_i^{*t}$ , the expected advantage function satisfies

$$\mathbb{E} [A_i^{\pi_t}(s_t, a_i^{*t})] \leq \frac{C}{\sqrt{t}}, \quad (15)$$

where  $A_i^{\pi_t}(s_t, a_i^{*t}) = Q_i^{\pi_t}(s_t, a_i^{*t}) - V_i^{\pi_t}(s_t)$ , and  $C = O\left(\frac{R_{\max}}{1-\gamma}\right) > 0$  is a constant depending on the reward bound  $R_{\max} = 1$  and the geometry of the policy space.

**C.2.2 Application of the Lemma.** According to the lemma above, the value function gap is linked to the expected advantage function:

$$V_i^*(s_t) - V_i^{\pi_t}(s_t) = \frac{1}{1-\gamma} \mathbb{E} [A_i^{\pi_t}(s_t, a_i^{*t})]. \quad (16)$$

This follows from the linearity of expectation and the determinism of the optimal action (i.e., when  $a_i^{*t}$  is fixed,  $\mathbb{E}_{a_i^{*t}} [A_i^{\pi_t}(s_t, a_i^{*t})] = A_i^{\pi_t}(s_t, a_i^{*t})$ ).

**C.2.3 Bound on Value Function Difference.** Combining Equation 15 and Equation 16, we obtain  $V_i^*(s_t) - V_i^{\pi_t}(s_t) \leq \frac{1}{1-\gamma} \cdot \frac{C}{\sqrt{t}}$ .

**C.2.4 Individual Agent Regret Summation.** The cumulative regret of agent  $i$  over  $T$  steps is

$$R_i(T) = \mathbb{E} \left[ \sum_{t=1}^T (V_i^*(s_t) - V_i^{\pi_t}(s_t)) \right] \leq \frac{C}{1-\gamma} \sum_{t=1}^T \frac{1}{\sqrt{t}}.$$

**C.2.5 Bounding the Harmonic Sum.** Using a standard integral bound on the partial harmonic series with exponent  $1/2$ :  $\sum_{t=1}^T \frac{1}{\sqrt{t}} \leq 2\sqrt{T}$ .

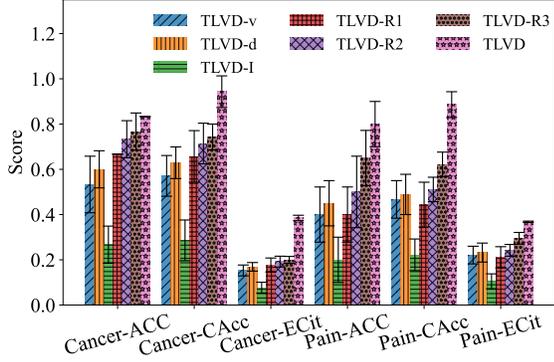
**C.2.6 Global Regret Bound.** The total regret is the sum over all agents:

$$R(T) = \sum_{i=1}^N R_i(T) \leq \sum_{i=1}^N \frac{2C\sqrt{T}}{1-\gamma} = \frac{2NC\sqrt{T}}{1-\gamma} = O\left(\frac{N\sqrt{T}}{1-\gamma}\right).$$

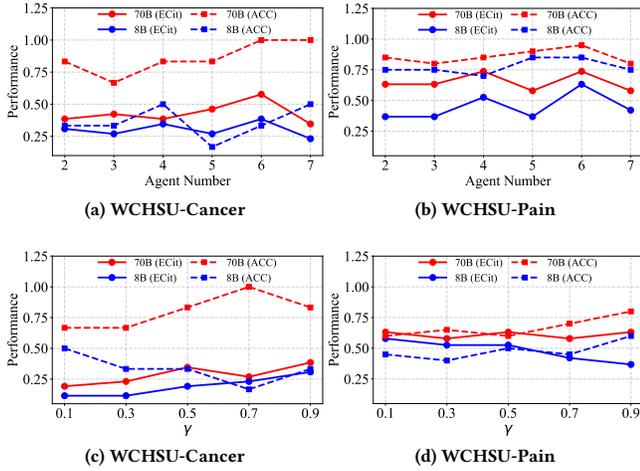
Note that this proof relies on the convergence property of policy gradient methods, which ensures that the advantage function  $A_i^{\pi_t}(s_t, a^{*t})$  decreases at a rate of  $O(1/\sqrt{t})$ . This convergence rate can be derived from online convex optimization theory, such as mirror descent.

**Table 7: Comparison of performance using different LLMs.**

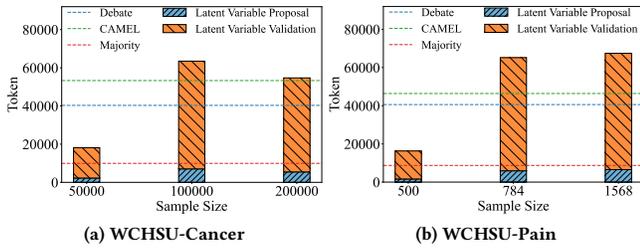
Methods	WCHSU-Cancer (n = 12)			WCHSU-Pain		
	ACC	CAcc	ECit	ACC	CAcc	ECit
Qwen3 235B	0.833	1.000	0.385	0.500	1.000	0.316
Llama 4 Maverick	0.500	0.750	0.115	0.750	1.000	0.474
GPT-oss-120B	0.667	1.000	0.269	1.000	1.000	0.632
DeepSeek-v3	0.833	1.000	0.385	0.750	0.857	0.368



**Figure 8: Performance of evaluation for ablation study on WCHSU-Cancer (n=12) and WCHSU-Pain.**



**Figure 9: Performance comparison with different N and  $\gamma$ .**



**Figure 10: The cost statistics of our model.**

**Table 5: Performance comparison on benchmark datasets.**

Methods	Multitasking Behaviour			Teacher’s Burnout Study		
	ACC	CAcc	ECit	ACC	CAcc	ECit
GPT5	0.250±0.047	0.083±0.053	0.083±0.053	0.145±0.045	0.069±0.029	0.032±0.013
Gemini-deepresearch	<b>0.500±0.000</b>	0.657±0.070	<b>0.383±0.041</b>	0.364±0.100	0.315±0.038	0.146±0.017
OpenAI-deepresearch	0.350±0.122	0.150±0.062	0.150±0.062	0.564±0.068	0.762±0.049	0.354±0.023
Qwen-deepresearch	0.050±0.100	0.000±0.000	0.000±0.000	0.164±0.068	0.154±0.024	0.071±0.011
Doubao-deepresearch	0.000±0.000	0.000±0.000	0.000±0.000	0.127±0.045	0.131±0.039	0.061±0.018
Autogen	0.050±0.100	0.000±0.000	0.000±0.000	0.255±0.068	0.208±0.039	0.096±0.018
MiniMax	0.150±0.122	0.000±0.000	0.000±0.000	0.327±0.109	0.300±0.051	0.139±0.026
CAMEL	0.050±0.100	0.000±0.000	0.000±0.000	0.236±0.072	0.192±0.024	0.089±0.011
MULTI-AGENT-MAJORITY	0.100±0.122	0.167±0.105	0.083±0.053	0.273±0.141	0.185±0.029	0.086±0.013
MULTI-AGENT-DEBATE	0.000±0.000	0.000±0.000	0.000±0.000	0.291±0.134	0.162±0.015	0.075±0.007
TLVD	<b>0.500±0.000</b>	<b>0.820±0.095</b>	0.283±0.041	<b>0.582±0.045</b>	<b>0.769±0.049</b>	<b>0.357±0.023</b>

**Table 6: Performance comparison of models using the Hier.rank [21] method.**

Methods	WCHSU-Cancer (n = 12)			WCHSU-Pain		
	ACC	CAcc	ECit	ACC	CAcc	ECit
GPT5	0.200±0.245	0.125±0.065	0.125±0.065	0.300±0.187	0.100±0.033	0.100±0.033
Gemini-deepresearch	0.100±0.200	0.000±0.000	0.000±0.000	0.100±0.122	0.191±0.021	0.163±0.018
OpenAI-deepresearch	0.500±0.316	0.250±0.258	0.250±0.135	0.450±0.100	0.450±0.041	0.200±0.018
Qwen-deepresearch	0.200±0.245	0.217±0.027	0.208±0.000	0.300±0.100	0.443±0.029	0.237±0.018
Doubao-deepresearch	0.100±0.200	0.208±0.037	0.250±0.113	0.150±0.122	0.000±0.000	0.000±0.000
Autogen	0.200±0.245	0.083±0.026	0.083±0.026	0.250±0.158	0.000±0.000	0.000±0.000
MiniMax	0.300±0.245	0.216±0.017	0.216±0.017	0.350±0.122	0.412±0.037	0.259±0.023
CAMEL	0.400±0.200	0.000±0.000	0.000±0.000	0.350±0.122	0.000±0.000	0.000±0.000
MULTI-AGENT-MAJORITY	0.300±0.245	0.000±0.000	0.000±0.000	0.300±0.100	0.000±0.000	0.000±0.000
MULTI-AGENT-DEBATE	0.400±0.200	0.286±0.090	0.092±0.017	0.500±0.000	0.600±0.122	0.104±0.028
TLVD	<b>0.700±0.245</b>	<b>1.000±0.035</b>	<b>0.646±0.012</b>	<b>0.750±0.106</b>	<b>1.000±0.172</b>	<b>0.471±0.164</b>

## D More experimental results

We conducted additional experiments on two benchmark datasets with ground-truth labels: Multitasking Behaviour and Teacher’s Burnout Study. As shown in Table 5, our method still achieves the best performance. Although these two datasets might have been included in the training data of LLMs, all the baselines we compared against are also LLM-based, making the comparison reasonably fair and still demonstrating the effectiveness of our method.

Additionally, to further investigate the impact of different causal discovery algorithms on our framework, we employed Hier.rank [21] to uncover latent causal graph structures. As shown in Table 6, the results indicate that our method is robust and maintains optimal performance across different causal discovery algorithms. Although some edges still lack explicit causal evidence, based on the identifiability guarantees of the Hier.rank and RLCD [11] algorithms, the latent variables inferred by our framework are also highly likely to be correct. The results of the ablation experiments are shown in Table 7 and Figure 8. See Figure 9 for the parameter analysis.

Finally, we also analyzed the token consumption of our method across different modules and sample sizes. As shown in Figure 10, the token consumption is higher during the latent variable validation phase, primarily due to the processing of papers and Wiki texts containing rich textual content. In contrast, the latent variable proposal phase requires relatively few tokens, mainly thanks to modeling agent interactions via the belief network. Although our method consumes slightly more tokens than the baseline models, this is acceptable given the performance improvements achieved. Moreover, the token usage of our method is primarily determined by the number of latent variables and the size of their corresponding Markov blankets.