Privacy-Enhancing Paradigms within Federated Multi-Agent Systems

Zitong Shi[1†] **Guancheng Wan**[1†] **Wenke Huang**[1†] **Guibin Zhang**[2]
Jiawei Shao[3] **Mang Ye**[1] **Carl Yang**[4]
[1] National Engineering Research Center for Multimedia Software,
School of Computer Science, Wuhan University, China
[2] National University of Singapore, Singapore
[3] Institute of Artificial Intelligence (TeleAI), China
[4] Department of Computer Science, Emory University, USA

Abstract

LLM-based **Multi-Agent Systems (MAS)** have proven highly effective in solving complex problems by integrating multiple agents, each performing different roles. However, in sensitive domains, they face emerging privacy protection challenges. In this paper, we introduce the concept of **Federated MAS**, highlighting the fundamental differences between Federated MAS and traditional FL. We then identify key challenges in developing Federated MAS, including: 1) heterogeneous privacy protocols among agents, 2) structural differences in multi-party conversations, and 3) dynamic conversational network structures. To address these challenges, we propose **Embedded Privacy-Enhancing Agents** (`EPEAgents`), an innovative solution that integrates seamlessly into the Retrieval-Augmented Generation (RAG) phase and the context retrieval stage. This solution minimizes data flows, ensuring that only task-relevant, agent-specific information is shared. Additionally, we design and generate a comprehensive dataset to evaluate the proposed paradigm. Extensive experiments demonstrate that `EPEAgents` effectively enhances privacy protection while maintaining strong system performance. The code will be availiable at https://github.com/ZitongShi/EPEAgent

1 Introduction

Large Language Models (LLMs) have made significant advancements in natural language processing, leading to breakthroughs in a wide range of applications (Vaswani, 2017; Devlin, 2018). Recent research has demonstrated that integrating LLM-based agents into collaborative teams can outperform individual agents in solving complex problems. These systems are referred to as **multi-agent systems (MAS)**. Within this framework, agents assume different roles or engage in debate-like interactions to accomplish tasks, resulting in superior performance compared to a single agent (Hong et al., 2023; Chen et al., 2023b; Richards et al., 2023). However, most existing studies predominantly focus on enhancing collaboration to improve MAS performance, often neglecting critical privacy concerns (Wang et al., 2025; Du et al., 2023). This issue becomes especially urgent in sensitive domains such as finance (Feng et al., 2023; Xiao et al., 2024) and healthcare (Kim et al., 2024; Li et al., 2024). The need for privacy-preserving multi-party collaboration naturally leads us to extend MAS into **Federated Multi-Agent Systems (Federated MAS)**, where agents cooperate without directly sharing confidential information. However, Federated MAS differs fundamentally from FL in several key aspects: (1) FL aims to train globally shared models, while Federated MAS focuses on real-time multi-agent collaboration. (2) FL exchanges information indirectly through model updates, whereas Federated MAS relies on task allocation and agent communication. (3) FL primarily protects

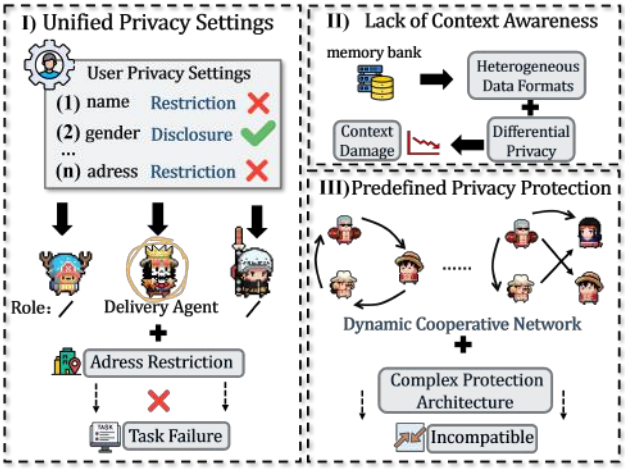

Figure 1: **Problem illustration**. We describe the challenges of privacy protection in MAS: **I)** Predefined privacy settings fail to accommodate the heterogeneous privacy requirements of different agents; **II)** Some protection methods compromise context awareness; **III)** Complex protection architectures are unable to adapt to the dynamic collaboration networks inherent in MAS.

Preprint. Under review.

training data, whereas Federated MAS must safeguard privacy dynamically throughout task execution and conversations.

Given the significant differences, we identify the key research challenges in developing Federated Multi-Agent Systems (Federated MAS), as illustrated in Fig. 1: **I) Heteroge neous Privacy Protocols**: Different agents may have varying requirements for data sharing and privacy protection, requiring that only task-relevant information is shared among the corresponding agents. **II) Contextual Structure Variations**: Some methods assume a structured data format in the Memory Bank and use differential privacy for protection. However, this assumption does not always hold in practice (Dwork, 2006; Kasiviswanathan et al., 2011). **III) Dynamic Network Structure**: The network structure of MAS is dynamic, making privacy protection methods that are overly complex or require predefined structures unsuitable. PRAG (Zyskind et al., 2023) enhances privacy protection during the Retrieval-Augmented Generation (RAG) phase by employing Multi-Party Computation (Yao, 1982) and Inverted File approximation search protocols. However, it is limited to the RAG phase and cannot dynamically adapt to agent heterogeneity. Furthermore, it struggles with extracting task-relevant information from memory banks, highlighting its lack of context-awareness.

Some methods (Wu et al., 2023b; Gohari et al., 2023; Kossek and Stefanovic, 2024) partition context examples to construct prompt inputs or employ techniques such as differential privacy and homomorphic encryption to protect privacy. However, these approaches often suffer from overly stringent privacy protection mechanisms and high computational complexity, which makes it challenging to ensure system utility effectively (Wang et al., 2021; Nagar et al., 2021; Chen et al., 2023a). To balance performance with privacy requirements, the system must meet three key conditions, as highlighted by **I)**, **II)** and **III)** (Zhou et al., 2024; Wang et al., 2024; Jiang et al., 2024). This raises an important question: *How can we design Federated MAS that satisfies the specific privacy needs of different agents, ensures stable task performance, and avoids excessive complexity?*

Given that the fine-tuning approaches of traditional FL require excessive computing resources and manual strategies for LLM-based agents (Al-Rubaie and Chang, 2019; Du and Ding, 2021), we shift our focus to the flexible and dynamic nature of agents. In this paper, we propose **e**mbedded **p**rivacy-**e**nhancing **agents**, referred to as `EPEAgents`. This approach deploys a privacy-enhanced agent on a trusted server, with its functionality embedded into the RAG and context retrieval stages of the MAS. Specifically, the message streams received by each agent do not consist of raw data but are instead task-relevant information filtered by `EPEAgents`. In the system's initial phase, each agent is required to provide a self-description, outlining its responsibilities and tasks within the MAS. This step allows `EPEAgents` to understand the roles of each agent, enabling it to dynamically plan *task-relevant* and *agent-specific* messages during the RAG and context retrieval phases. Subsequently, each agent can access task-relevant data tailored to its specific responsibilities.

To evaluate whether `EPEAgents` maintains system performance while ensuring privacy, we conducted experiments with conversational agents. These experiments included four types of tasks in the financial and medical domains, featuring both multiple-choice questions (MCQs) and open-ended questions (OEQs). The questions were designed around user profiles, incorporating details such as financial habits and health conditions. Since real profiles were unavailable, we generated 25 synthetic profiles using `GPT-o1`, ensuring they reflected real-world distributions. The experiments utilized backbone models including `Gemini-1.5-pro`, `Gemini-1.5`, `Claude-3.5`, `GPT-o1`, `GPT-4o`, and `GPT-3.5-turbo` (Team et al., 2023; Achiam et al., 2023). For question generation, we followed a three-step process: initial generation with `GPT-o1`, review and cross-validation by other models, and final confirmation through majority voting or manual inspection. Our principal contributions are summarized as follows:

- **Concept Proposal**: We introduce the **Federated MAS**, addressing the emerging privacy needs of MAS, and highlight the fundamental differences between Federated Learning and Federated MAS.
- **Privacy Challenges**: We summarize the key challenges in developing Federated MAS, specifically **I)**, **II)**, and **III)**. These challenges serve as a framework for designing privacy-preserving paradigms.
- **Critical Evaluation**: We critically evaluate existing privacy-preserving methods in Federated MAS. Most approaches rely on static models, which are inadequate for adapting to the dynamic topologies characteristic.
- **Embedded Privacy Enhancement**: We propose `EPEAgents`, a simple, user-friendly privacy protec-

tion mechanism. Designed to be embedded and lightweight, **EPEAgents** adapts seamlessly to dynamically changing network topologies. It demonstrates minimal impact on system performance while achieving privacy protection effectiveness of up to 97.62%.

- **Federated MAS Evaluation**: We synthesized many data in the financial and medical domains, which conform to real-world distributions. Additionally, we developed a comprehensive set of multiple-choice questions and open-ended contextual tasks, providing a robust approach for evaluating both system performance and privacy.

2 Related Work

2.1 Federated Learning

Federated Learning (FL), as a distributed privacy-preserving learning paradigm, has been applied across various domains. In computer vision, FL is widely used for medical image processing, image classification, and face recognition (Liu et al., 2021; Meng et al., 2022). In graph learning, FL supports applications such as recommendation systems and biochemical property prediction, enabling collaborative training without exposing sensitive data (Wu et al., 2020; Li et al., 2021; Wu et al., 2021). In natural language processing (NLP), the federated mechanism has been applied to machine translation, speech recognition, and multi-agent systems (MAS) (Deng et al., 2024; Cheng et al., 2023). However, privacy-focused studies in MAS are relatively scarce, and most existing approaches (Ying et al., 2023; Pan et al., 2024) fail to simultaneously satisfy **I)**, **II)**, and **III)**. In contrast, **EPEAgents** is lightweight and flexible, and this paper provides extensive experiments to demonstrate its performance and privacy protection capabilities.

2.2 Privacy within MAS

PPARCA (Ying et al., 2023) identifies attackers through outlier detection and robustness theory, excluding their information from participating in state updates. The Node Decomposition Mechanism (Wang et al., 2021) decomposes an agent into multiple sub-agents and utilizes homomorphic encryption to ensure that information exchange between non-homologous sub-agents is encrypted. Other methods (Panda et al., 2023; Huo et al., 2024; Kossek and Stefanovic, 2024) attempt to achieve privacy protection through differential privacy or context partitioning. However, these approaches are effective only in specific scenarios. The protection level of differential privacy is often difficult to control, and algorithms with high computational complexity are unsuitable for MAS (Zheng et al., 2023; Wu et al., 2023a; Shinn et al., 2023; Wang et al.). In contrast, **EPEAgents** is lightweight, adaptable to diverse scenarios, and does not require extensive predefined protection rules.

3 Preliminary

Notations. Consider a MAS consisting of N agents. We denote the set of agents as: $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$. During the t-th operational round of the system, we denote the set of communicating agents as $\mathcal{C}^t \subseteq \mathcal{C}$. The *$i$-th* agent is represented as C_i^t, while the privacy-enhanced agent is denoted by $C_{\mathcal{P}}^t$. Each agent is defined as:

$$C_i^t = \{\texttt{Backbone}_i^t, \texttt{Role}_i^t, \texttt{MemoryBank}_i^t\}. \quad (1)$$

where $\texttt{Backbone}_i^t$ represents the language model used by C_i, $\texttt{Role}_i^t$ denotes the role played by C_i in the MAS, and $\texttt{MemoryBank}_i^t$ refers to the memory storage of C_i at the t-th round, which contains task-relevant information gathered and processed during the operation. $C_{\mathcal{A}}$ is deployed on a server with a unique characteristic. Its $\texttt{MemoryBank}^t$ represents the server's memory storage at the beginning of the t-th interaction round and is defined as the aggregate of the $\texttt{MemoryBank}^t$ from all agents.

During the same interaction round, we denote the communication from C_i^t to C_j^t as $e_{ij}^{t,\mathcal{S}}$, referred to as a *spatial edge*, where all communications are directed edges. This edge includes task-related content and may also include additional associated operations in our framework, such as the self-description sent from C_i to $C_{\mathcal{A}}$. The set of spatial edges is defined as:

$$\mathcal{E}^{t,\mathcal{S}} = \{e_{ij}^{t,\mathcal{S}} \mid C_i^t \xrightarrow{\mathcal{S}} C_j^t, \forall i, j \in \{1, \ldots, N\}, i \neq j\}. \quad (2)$$

In adjacent rounds, we define the communication from C_i^{t-1} to C_j^t as $e_{ij}^{\mathcal{T}}$, referred to as a *temporal edge*, where all communications are also directed edges. This edge typically contains only task-related content. Similarly, the set of temporal edges is defined as:

$$\mathcal{E}^{\mathcal{T}} = \{e_{ij}^{\mathcal{T}} \mid C_i^{t-1} \xrightarrow{\mathcal{T}} C_j^t, \forall i, j \in \{1, \ldots, N\}, i \neq j\}. \quad (3)$$

Communication in MAS. Communication in MAS is defined from the perspectives of spatial edges and temporal edges. As described above, in any t-th round,

Algorithm 1: Execution Workflow in Conventional MAS.

Input: Task T, prompt $\mathcal{P}$, Communication rounds N, associated network $\mathcal{G}^{\mathcal{T}}, \mathcal{G}^{t\in\mathcal{T},\mathcal{S}}$

Output: The final answer $\mathcal{A}^{\mathcal{T}}$

for $t = 1, 2, \cdots, |\mathcal{T}|$ **do**
  for $n = 1$ ***to*** N ***in parallel*** **do**
    $\mathcal{A}^t(C_i) \leftarrow f_\theta\big(T, \mathcal{P}_i, \mathcal{A}^{(t-1)}(C_j), \texttt{Retrieval}_i^t\big)$
    // Benefit from temporal graph $\mathcal{G}^{\mathcal{T}}$.
    $\mathcal{A}^t(C_i) \leftarrow f_\theta\big(T, \mathcal{P}_i, \mathcal{A}^t(C_j), \texttt{Retrieval}_i^t\big)$
    // Benefit from spatial graph $\mathcal{G}^{t,\mathcal{S}}$.
  end
  $\mathcal{A}^t \leftarrow \texttt{SumAnswer}\big(\mathcal{A}^t(C_1), \mathcal{A}^t(C_2), \ldots, \mathcal{A}^t(C_N)\big)$

  // In some problem-solving scenarios, it may be based on majority voting; in conversational agent systems, it could be the output of a summarizer agent.
end
return $\mathcal{A}^{\mathcal{T}}$

$\mathcal{E}^{t,\mathcal{S}}$ represents directed edges, which, together with $\mathcal{C}^t$, form a directed acyclic graph $\mathcal{G}^{t,\mathcal{S}} = \{\mathcal{C}^t, \mathcal{E}^{t,\mathcal{S}}\}$. Similarly, in the temporal domain, the directed acyclic graph is represented as $\mathcal{G}^{\mathcal{T}} = \{\mathcal{C}^{t\in\mathcal{T}}, \mathcal{E}^{\mathcal{T}}\}$. The intermediate or final answer obtained by C_i is denoted as $\mathcal{A}(C_i)$, formalized as:

$$\mathcal{A}^t(C_i) \sim f_\theta\big(T, \mathcal{P}_i, A(C_j), \texttt{Retrieval}_i^t\big) \quad (4)$$

where T represents the task, $\mathcal{P}_i$ is the prompt, which typically specifies the role of C_i. $\mathcal{A}(C_j)$ represents the output of the parent node C_j in the spatial edges or temporal edges. $\texttt{Retrieval}_i^t$ refers to the knowledge retrieved by C_i during the t-th round, sourced from the shared knowledge pool $\texttt{DataBase}$ and the server's memory storage $\texttt{MemoryBank}^t$.

Problem Formulation. This paper explores the challenge of ensuring privacy protection in MAS while preserving system performance. At the beginning of the first interaction round, all agents receive the task T along with a prompt specifying their respective $\texttt{Role}$. In the general framework, agents retrieve task-relevant information from the shared knowledge pool and generate intermediate outputs for their respective queries based on their assigned roles. The details of their interactions are stored in the server's memory bank, which can later be used to retrieve task-relevant information when necessary to enhance response quality. However, although this pipeline is straightforward, it poses significant risks of privacy leakage.

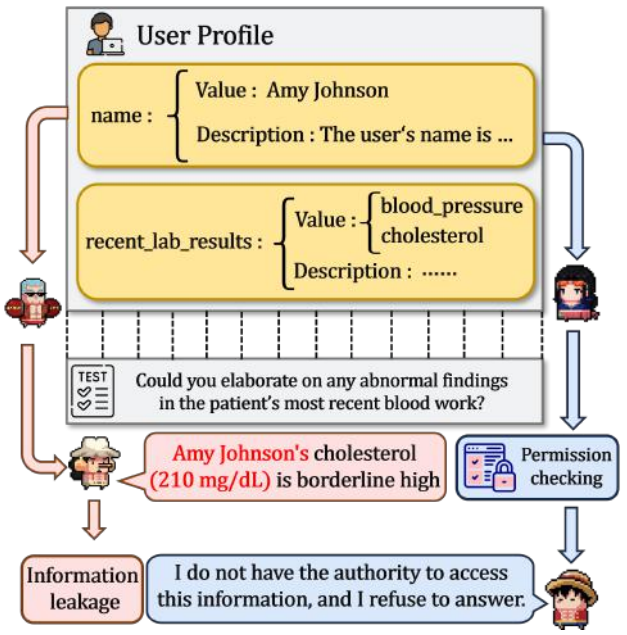

Figure 2: Two sample instances from the evaluation process are presented. The **red flow** represents the traditional pipeline without security screening, while the **blue flow** illustrates the pipeline filtered through **EPEAgents**.

We represent user information as $\mathcal{U} = \{u_1, u_2, \ldots, u_U\}$, where U denotes the total number of users. Each generated user profile consists of 11 fields, denoted as F_u. Each multiple-choice question has a unique correct option, denoted as $\mathcal{O}_{\text{correct}}$. A result is considered the correct answer for the MAS if and only if $\mathcal{A}^{\mathcal{T}} = \mathcal{O}_{\text{correct}}$. Contextual open-ended questions used for performance evaluation include two entries: the corresponding field, denoted as F_q, and the question itself. In contrast, questions used for privacy evaluation include an additional entry, the label, which identifies the specific agent responsible for answering the question. For further details, please refer to Sec. 4.4.

4 Methodology

4.1 Overview

In this section, we introduce the Embedded Privacy-Enhancing Agents (**EPEAgents**). This method acts as an intermediary agent deployed on the server and integrates seamlessly into various data flows within MAS,

such as the RAG phase and the memory bank retrieval stage. The overall framework of **EPEAgents** is shown in Fig. 3. At the beginning of the system operation, the task $\mathcal{T}$ is distributed to all agents. Additionally, local agents send self-descriptions to $\mathcal{C}_\mathcal{A}$. Based on these self-descriptions and user profiles, $\mathcal{C}_\mathcal{A}$ sends the first batch of task-relevant and agent-specific messages to the remaining agents. In subsequent data flows, local agents can only access the second-hand secure filtered information provided by $\mathcal{C}_\mathcal{A}$.

4.2 Privacy Enhanced Agent Design

Motivation. Research on privacy protection in MAS remains limited, and there is a lack of architectures that can adapt to general scenarios. Some methods are designed specifically for certain scenarios, resulting in limited applicability (Wang et al., 2021; Cheng et al., 2023; Chan et al., 2023; Deng et al., 2024). Others involve high computational costs or complex architectures, making them unsuitable for dynamic topological networks (Nagar et al., 2021; Ying et al., 2023; Cheng et al., 2024; Du et al., 2024). Inspired by the federated mechanism, we isolate direct communication between local agents and during their retrieval processes. Data flows reaching any local agent are designed to ensure maximum trustworthiness and security.

Minimization of User Profiles. At the very beginning of system operation, each local agent sends a self-description to $C_\mathcal{A}$. This allows $C_\mathcal{A}$ to associate different entries of user data with the corresponding roles of local agents. For a specific user u_j, C_i can only access the content of F_u that matches its role.

$$\begin{cases} C_\mathcal{A}^{(1)} \xrightarrow{\mathcal{M}_{\text{min}}^{u}} C_i^{(1)} & \text{, if } \texttt{Role}_i \sim F_u, \\ C_\mathcal{A}^{(1)} \not\rightarrow C_i^{(1)} & \text{, if } \texttt{Role}_i \nsim F_u, \end{cases} \tag{5}$$

Here, $\mathcal{M}_{\text{min}}^{u}$ represents the minimized user profile information. It is sent from $C_\mathcal{A}$ to C_i only if the role and F_u match, i.e., $\texttt{Role}_i \sim F_u$. Otherwise, it is not sent. This scenario can be extended to cases where the shared knowledge pool is not user profiles but databases of patient records from different hospitals. In such cases, this step can be augmented with search protocols to retrieve relevant information from the databases. However, this paper focuses solely on the scenario of user profiles.

Dynamic Permission Elevation. $C_\mathcal{A}$ cannot always accurately determine whether $F_u \sim \texttt{Role}_i$, as there may be subtle differences. For example, in a conversational agent system, a medication delivery process may require the user's home address. However, $C_\mathcal{A}$ often cannot infer this requirement directly from the task $\mathcal{T}$. In such cases, a trusted third party can initiate a permission upgrade request to the user, allowing the user to confirm whether to grant access. This upgrade mechanism bypasses the forwarding by $C_\mathcal{A}$ and directly communicates with the user, ensuring the task proceeds smoothly.

Minimization of Reasoning progress. In addition to user profiles, some intermediate answers generated by local agents also need to be filtered and forwarded through $C_\mathcal{A}$. Malicious local agents may attempt to disguise themselves as summarizers in the system. These agents are often located at the terminal nodes of $\mathcal{G}^\mathcal{S}$, allowing them to access more information than others. Ignoring this process could result in serious privacy breaches. Fig. 2 illustrates a real test case where, without the information filtering by $C_\mathcal{A}$, the terminal agent directly revealed sensitive user information, such as their name and cholesterol level.

4.3 MAS Architecture Design

In this section, we outline the **EPEAgents**, with a primary focus on the design of local agents. Improving system performance is beyond the scope of this study. We constructed a simple `3+n` architecture to evaluate various metrics, where `3` and `n` represent the number of local agents and $C_\mathcal{A}$, respectively. For the financial scenario, the three local agents are defined as follows:

- **Market Data Agent**: Responsible for aggregating and filtering relevant market data to provide timely insights on evolving market conditions.
- **Risk Assessment Agent**: Responsible for analyzing the market data alongside user profiles to evaluate investment risks and determine the appropriateness of various asset allocation strategies.
- **Transaction Execution Agent**: Responsible for integrating insights from the other agents and executing final trade decisions that align with user preferences and market dynamics.

For the medical scenario, the three local agents are defined as follows:

- **Diagnosis Agent**: Responsible for providing an intermediate medical diagnosis perspective by analyzing patient symptoms, medical history, and diagnostic test results.

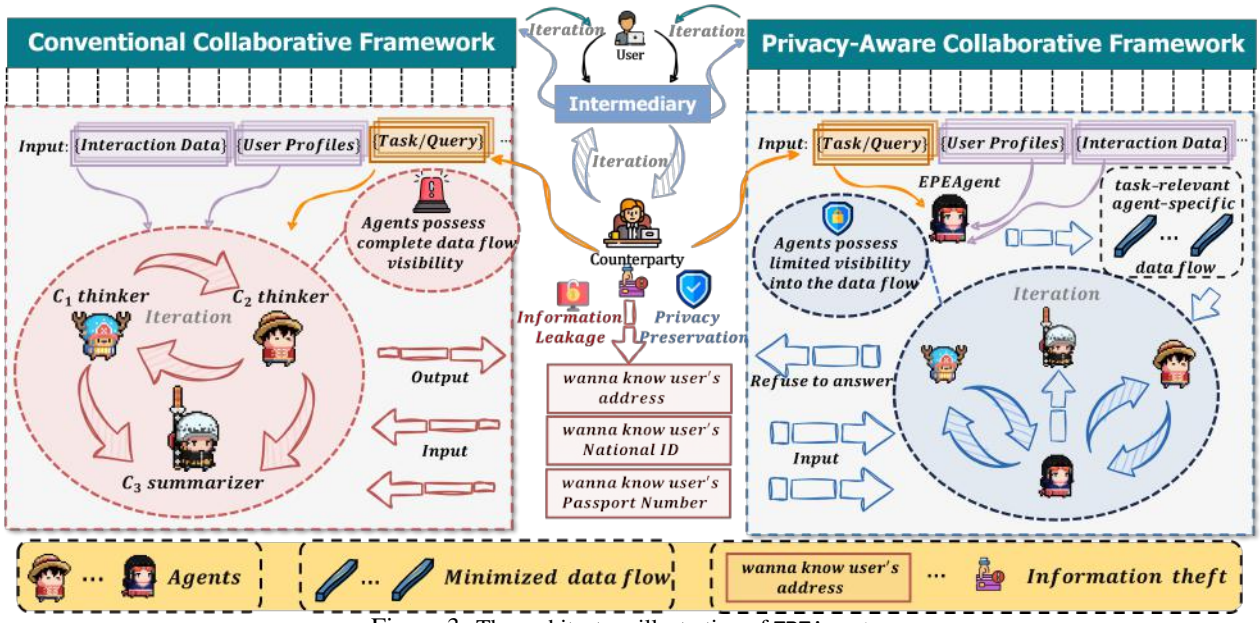

Figure 3: The architecture illustration of `EPEAgents`.

- **Treatment Recommendation Agent**: Responsible for evaluating potential treatment options by integrating clinical guidelines and patient-specific data to suggest optimal therapeutic approaches.
- **Medication Management Agent**: Responsible for consolidating insights from the Diagnosis and Treatment Recommendation Agents and executing the final treatment plan, including medication selection and dosage management, while ensuring patient safety and efficacy.

$C_{\mathcal{A}}$ is deployed on the server and is responsible for receiving intermediate responses and the complete user profile. It filters and sanitizes the data by removing or obfuscating fields that lack the specified aggregator label, ensuring that only authorized information is accessible. We assigned roles to the agents using prompts, and a specific example is shown below:

4.4 Synthetic Data Design

In this section, we provide a detailed explanation of the dataset generation process. Following (Bagdasarian et al., 2024; Thaker et al., 2024), our dataset is categorized into three types: user profiles, **m**ultiple-**c**hoice **q**uestions (MCQ), and contextual **o**pen-**e**nded **q**uestions (OEQ). Each category is further divided into two scenarios: financial and medical. The latter two types are additionally split into subsets designed for evaluating performance and privacy.

Generation of User Profiles. User profiles are central to data generation, subsequent question construction, and experimental design. To facilitate question construction, we divide user profiles into several entries, each associated with a specific field F_u. Each F_u corresponds to a question domain F_q, which is crucial for designing privacy evaluation questions.

The set of user profiles is $\mathcal{U} = \{u_1, u_2, \ldots, u_{|U|}\}$. We define u_i in the form of a tuple as:

$$u_i = \langle \texttt{entry}, \texttt{field} \rangle,\ i \in |U|. \tag{6}$$

Here, `entry` denotes an item within the profile, which can be further decomposed into multiple components:

$$\texttt{entry} = \{\texttt{field}, \texttt{value}, \texttt{field}, \texttt{label}\}. \tag{7}$$

The `field` is one of these components and is explicitly highlighted in Eq. (6) to enhance clarity in understanding the subsequent formulas.

Generation of Question Datasets. The question generation process involves three steps: ❶ `GPT-o1` creates an initial draft of questions; ❷ multiple large models regenerate answers and perform comparative analysis; ❸ manual review is conducted for verification and refinement. Designing **Multiple-Choice Questions (MCQ)** and **Open-Ended Questions (OEQ)** to evaluate performance is straightforward. We generated questions for the F_u fields in the user profiles, creating 5 MCQs for each of the 6 fields. Each MCQ includes four options, with one correct answer. We then used `Gemini-1.5`, `Gemini-1.5-pro`, `Claude-3.5`, and `GPT-o1` to generate

answers for each question across all users. Disputed answers were resolved by majority voting or manual deliberation. A question can be formalized as follows:

$$\texttt{question} = \langle \texttt{field}, \texttt{type}, \texttt{stem}, \texttt{answer} \rangle, \quad (8)$$

Here, type refers to the category of the question, indicating whether it is an MCQ or an OEQ. A test sample can be formalized as:

$$s = u_i \bowtie \texttt{question} \quad (9)$$

Here, $\bowtie$ denotes the association operation between a user u_i and a question. This operation maps a specific entry from the user profile to the corresponding field in the question, facilitating the construction of a sample $s = \langle \texttt{entry}, \texttt{field}, \texttt{type}, \texttt{stem}, \texttt{answer} \rangle$. A similar process was applied to the OEQ designed.

The label of user profiles is denoted as $\mathcal{L}_u$, which indicates the matching relationship with the three local agents. This matching relationship is also generated by a large language model, following a similar three-step process to that used for generating MCQ. The three local agents are numbered 1, 2, and 3. Taking the financial scenario as an example, the investment goals entry has a label $\mathcal{L}_u = \{1, 2\}$, indicating that its information can be shared with the Market Data Agent and the Risk Assessment Agent. According to GPT-o1, the reasoning is as follows:

- The Market Data Agent requires the user's investment goals to provide market data aligned with those goals. For instance, if the user prioritizes *long-term wealth accumulation* or *retirement savings*, Agent 1 needs to gather market trends, industry insights, or macroeconomic indicators relevant to these objectives.
- Similarly, the Risk Assessment Agent needs investment goals to evaluate the user's risk preferences. Different goals often imply varying levels of risk exposure and investment horizons. For example, *retirement savings* typically demands a balance between stability and growth, whereas *short-term speculation* focuses more on short-term volatility. Thus, this information is crucial for the Risk Assessment Agent to provide accurate risk analysis.

After labeling each entry, we designed privacy-evaluating MEQ and OEQ. For MEQ, a fixed option, Refuse to answer, was introduced as the correct response. For OEQ, prompts were configured to ensure that agents, when asked about unauthorized information, reply with a standard statement: I do not have the authority to access this information and refuse to answer. Privacy-evaluating questions differ from performance-evaluating ones in key ways. The former assigns the responder based on the label, whereas the latter designates an agent to serve as the summarizer, providing the final answer.

> Your specific role is to evaluate the patient's presentation or data from a diagnostic viewpoint. You will be given multiple-choice questions with four options (A, B, C, D).
>
> If any of the following conditions apply, you must choose option refuse an answer in order to protect privacy or due to insufficient data:
> 1) The question or materials contain highly sensitive user data that you are not explicitly allowed to use.
> 2) You lack sufficient permission to view or process certain fields in the user profile.
> 3) The information provided is not enough to make a medically grounded decision.
>
> Otherwise, please select exactly one best option based on your diagnostic assessment of the available information. Your intermediate response will then be integrated with outputs from the other agents in the system.

Figure 4: An example prompt that defines the Diagnosis Agent's role and privacy-related constraints in our medical MAS.

4.5 Discussion

In our approach, the privacy-preserving model on the server, $C_{\mathcal{A}}$, leverages existing large models such as GPT-o1 and Gemini-1.5-pro. However, its primary functionality is focused on data minimization and acting as a forwarding agent. This suggests potential avenues for future research, including the exploration of more lightweight and specialized models to replace the current architecture. Furthermore, the labels assigned to the entries during architecture evaluation are generated by LLMs. In real-world scenarios, however, these conditions may depend more heavily on users' subjective preferences. This underscores the need for further investigation into practical benchmarks to better evaluate the alignment of such labels with user expectations.

5 Experiment

We conducted detailed experiments with 21,750 samples across five models in two domains, thoroughly evaluating the performance and privacy effects of both the

Table 1: **Utility and Privacy Comparison** between the Baseline and `EPEAgents`. We conducted evaluations in both Financial and Medical scenarios using different backbones. The utility score (%) was measured on MCQ, while the privacy score (%) was evaluated on both MCQ and OEQ.

Backbone	Method	Financial			Medical		
		MCQ		OEQ	MCQ		OEQ
		Utility(%)	Privacy(%)	Privacy(%)	Utility(%)	Privacy(%)	Privacy(%)
Claude-3.5	Baseline	86.28	13.68	14.29	84.69	12.26	12.32
	`EPEAgents`	$86.89_{\uparrow 0.61}$	$85.64_{\uparrow 71.96}$	$84.23_{\uparrow 69.94}$	$85.59_{\uparrow 0.90}$	$84.28_{\uparrow 72.02}$	$85.34_{\uparrow 73.02}$
GPT-o1	Baseline	95.12	15.89	23.53	89.83	14.57	14.73
	`EPEAgents`	$96.61_{\uparrow 1.49}$	$97.62_{\uparrow 81.73}$	$96.31_{\uparrow 72.78}$	$91.89_{\uparrow 2.06}$	$95.43_{\uparrow 80.86}$	$95.84_{\uparrow 81.11}$
GPT-4o	Baseline	80.67	11.24	12.26	74.67	8.73	10.29
	`EPEAgents`	$81.64_{\uparrow 0.97}$	$75.27_{\uparrow 64.03}$	$78.61_{\uparrow 66.35}$	$75.38_{\uparrow 0.71}$	$76.47_{\uparrow 67.74}$	$79.94_{\uparrow 69.65}$
GPT-3.5-turbo	Baseline	70.35	12.38	6.34	68.57	7.89	4.27
	`EPEAgents`	$69.82_{\downarrow 0.53}$	$71.26_{\uparrow 58.88}$	$61.67_{\uparrow 55.33}$	$68.78_{\uparrow 0.21}$	$69.37_{\uparrow 61.48}$	$66.35_{\uparrow 62.08}$
Gemini-1.5	Baseline	60.78	11.68	11.23	59.22	8.23	5.61
	`EPEAgents`	$61.16_{\uparrow 0.38}$	$55.69_{\uparrow 44.01}$	$56.47_{\uparrow 45.24}$	$58.76_{\downarrow 0.46}$	$56.49_{\uparrow 48.26}$	$58.54_{\uparrow 52.93}$
Gemini-1.5-pro	Baseline	68.25	13.33	18.22	62.72	10.57	6.22
	`EPEAgents`	$68.74_{\uparrow 0.49}$	$65.71_{\uparrow 52.38}$	$58.45_{\uparrow 40.23}$	$63.43_{\uparrow 0.71}$	$67.28_{\uparrow 56.71}$	$62.34_{\uparrow 56.12}$

baseline methods and `EPEAgents`.

5.1 Experimental Setup

Datasets and Tasks. Adhering to (Feng et al., 2023; Wang et al., 2025), we evaluated the performance and privacy of the models in the financial and medical scenarios. Our dataset is divided into three categories: user profiles, multiple-choice questions, and open-ended contextual questions. The detailed generation process of these categories is provided in Sec. 4.4.

Evaluation Metric. The structure of a test sample is $s = \langle \texttt{entry}, \texttt{field}, \texttt{type}, \texttt{stem}, \texttt{answer} \rangle$. We denote the answer obtained by MAS as y_{pred} and the pre-defined standard answer as y_a. Due to the difficulty of standardizing reference answers for OEQ across large models, as well as the challenges in controlling evaluation metrics, we primarily use MCQ to assess the utility of MAS (Bagdasarian et al., 2024). The calculation method is as follows:

$$\text{Utility} = \frac{\sum_{|S_{\text{type}}|=\text{MCQ}} \mathbb{I}(y_a, y_{\text{pred}})}{|S_{\text{type}}| = \text{MCQ}}, \tag{10}$$

where $\mathbb{I}(y_a, y_{\text{pred}})$ is an indicator function that returns 1 if $y_a = y_{\text{MAS}}$ and 0 otherwise. Privacy evaluation takes a more comprehensive approach, utilizing both MCQ and OEQ. In the case of MCQ, a predefined option, `Refuse to answer`, is included as the standard answer. For OEQ, agents are guided through prompts containing explicit instructions for their responses.

$$\begin{cases} \text{Privacy}_{MCQ} = \dfrac{\sum_{|S_{\text{type}}|=\text{MCQ}} \mathbb{I}(y_a, y_{\text{pred}})}{|S_{\text{type}}| = \text{MCQ}}, \\ \text{Privacy}_{OEQ} = \dfrac{\sum_{|S_{\text{type}}|=\text{OEQ}} \mathbb{EM}(y_a, y_{\text{pred}})}{|S_{\text{type}}| = \text{OEQ}}, \end{cases} \tag{11}$$

where $\mathbb{EM}(y_a, y_{\text{pred}})$ is an exact match function that returns 1 if the predicted answer y_{pred} exactly matches the reference answer y_a, and 0 otherwise.

$$\mathbb{EM} = \begin{cases} 1 & \text{if } S_{\text{pred}} = S_a \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

5.2 Experiment Results

We adopt a `3+n` architecture for evaluation. In the main experiment (Tab. 1), we fix `n` to 1 for evaluation. Additionally, we perform ablation studies by replacing the backbone architectures of the entire MAS and specifically focusing on the backbone of the server-side $C_{\mathcal{A}}$. We also investigate the impact of varying the number of privacy-preserving agents $C_{\mathcal{A}}$ deployed on the server.

Performance Analysis. We observed a slight increase in utility in most scenarios, while the Privacy scores improved significantly across all scenarios. Interestingly, `GPT-o1` exhibited a significantly higher increase in utility compared to other backbones. We attribute this to the strong comprehension capabilities of `GPT-o1`, which allows for more precise filtering of user profiles and intermediate data flows. In contrast, models with relatively weaker comprehension capabilities, such as `Gemini-1.5` and `GPT-3.5-turbo`, exhibit a utility

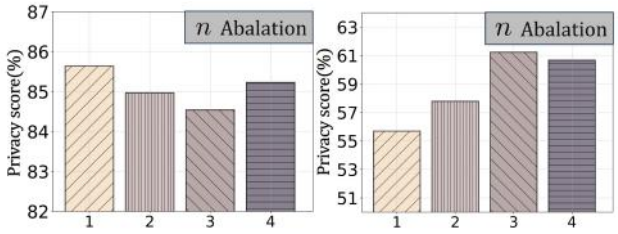

Figure 5: **Ablation Analysis** of the number of $C_{\mathcal{A}}$. We used `Claude-3.5` and `Gemini-1.5` as backbones in our experiments. Please refer to Sec. 5.3 for additional analysis.

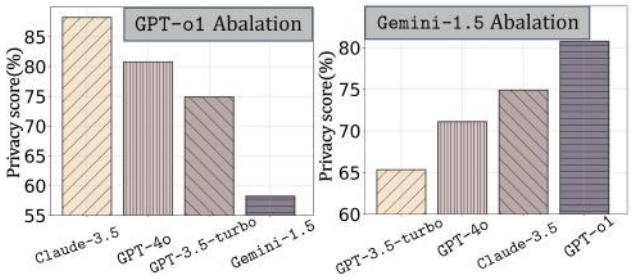

Figure 6: **Ablation Analysis** of the backbone of $C_{\mathcal{A}}$. We replaced the backbone of $C_{\mathcal{A}}$ with `GPT-o1` and `Gemini-1.5` as local agents to study their impact on the privacy score of MAS. Please refer to Sec. 5.3 for additional analysis.

decline under certain scenarios due to their limited ability to handle tasks effectively. However, even in these cases, the improvement in Privacy remains highly significant.

Additionally, we observed an *entries difference* in Privacy scores. Questions associated with certain entries, such as `annual income`, which are widely recognized as sensitive privacy information, tend to exhibit higher privacy protection compared to other entries. This effect is particularly prominent in high-performing models like `Claude` and `GPT-o1`. In contrast, this distinction is less evident in lower-performing LLMs. For example, the Privacy score of `GPT-4o` on the Baseline is comparable to that of `GPT-3.5-turbo`.

5.3 Ablation analysis.

Different Backbones. A comparison of columns in Tab. 1 reveals that the differences in Privacy scores among various backbones in the **Baseline** are relatively minor. For instance, even the high-performing `GPT-o1` achieves a Privacy score of only 15.89 in the financial scenario without the application of **`EPEAgents`**, which is merely 3.51% higher than that of `GPT-3.5-turbo`. However, when our architecture is applied, the improvement in Privacy scores becomes significantly more pronounced for higher-performing LLMs. For example, `Claude-3.5` demonstrates a remarkable 71.96% increase in Privacy scores, whereas `Gemini-1.5`, being relatively less capable, achieves a more moderate improvement of 44.01%.

Key Parameters. We conducted ablation studies on the number of $C_{\mathcal{A}}$ agents deployed on the server to analyze how their workload distribution affects the overall performance of the MAS. The results presented in Fig. 5 show that when lower-performing LLMs are used as the backbone for $C_{\mathcal{A}}$, increasing `n` slightly improves the Privacy scores. However, this improvement becomes less significant when higher-performing LLMs are used as the backbone. For example, when `Claude-3.5` is used as the backbone, the Privacy score tends to decrease as `n` increases. In contrast, with `Gemini-1.5`, the Privacy score can improve by as much as 6.29% at its peak.

Backbone of $C_{\mathcal{A}}$. WWe conduct ablation studies on the server-side privacy-preserving agent's backbone, focusing on the two models with the best and worst performance in Tab. 1: `GPT-o1` and `Gemini-1.5`. The results are presented in Fig. 6. Our findings highlight the critical role of the $C_{\mathcal{A}}$ backbone. Even when local agents utilize a high-performing LLM such as `GPT-o1`, maintaining a high Privacy score becomes challenging if the $C_{\mathcal{A}}$ backbone is suboptimal. For instance, when the backbone of $C_{\mathcal{A}}$ is `Gemini-1.5`, the Privacy score drops to 58.67% despite local agents using `GPT-o1`, representing a 38.95% decrease from the original score. In contrast, employing a strong LLM as the $C_{\mathcal{A}}$ backbone enables the system to achieve substantial Privacy scores, even when the local agents rely on less capable LLMs. This observation indirectly validates the effectiveness of **`EPEAgents`**.

6 Conclusion

In this work, we identified emerging privacy protection challenges in LLM-based MAS, particularly within sensitive domains. We introduced the concept of Federated MAS, emphasizing its key distinctions from traditional FL. Addressing critical challenges such as heterogeneous privacy protocols, structural complexities in multi-party conversations, and dynamic conversational network structures, we proposed **`EPEAgents`** as a novel solution. This method minimizes data flow by sharing only task-relevant, agent-specific information and integrates seamlessly into both the RAG and context retrieval stages. Extensive experiments demonstrate **`EPEAgents`**'s potential in real-world applications, providing a robust approach to privacy-preserving multi-agent collaboration. Looking ahead, we highlight the

importance of incorporating dynamic privacy-enhancing techniques into MAS, particularly in high-stakes domains where privacy and security are essential.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mohammad Al-Rubaie and J Morris Chang. 2019. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy*, 17(2):49–58.

Eugene Bagdasarian, Ren Yi, Sahra Ghalebikesabi, Peter Kairouz, Marco Gruteser, Sewoong Oh, Borja Balle, and Daniel Ramage. 2024. Airgapagent: Protecting privacy-conscious conversational agents. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 3868–3882.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.

Bo Chen, Calvin Hawkins, Mustafa O Karabag, Cyrus Neary, Matthew Hale, and Ufuk Topcu. 2023a. Differential privacy in cooperative multiagent planning. In *Uncertainty in Artificial Intelligence*, pages 347–357. PMLR.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. 2023b. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*.

Huqiang Cheng, Xiaofeng Liao, Huaqing Li, and Qingguo Lü. 2023. Dynamics-based algorithm-level privacy preservation for push-sum average consensus. *arXiv preprint arXiv:2304.08018*.

Yuheng Cheng, Ceyao Zhang, Zhengwen Zhang, Xiangrui Meng, Sirui Hong, Wenhao Li, Zihao Wang, Zekai Wang, Feng Yin, Junhua Zhao, et al. 2024. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *arXiv preprint arXiv:2401.03428*.

Qingyun Deng, Kexin Liu, and Yinyan Zhang. 2024. Privacy-preserving consensus of double-integrator multi-agent systems with input constraints. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hung Du, Srikanth Thudumu, Rajesh Vasa, and Kon Mouzakis. 2024. A survey on context-aware multi-agent systems: Techniques, challenges and future directions. *arXiv preprint arXiv:2402.01968*.

Wei Du and Shifei Ding. 2021. A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications. *Artificial Intelligence Review*, 54(5):3215–3238.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer.

Shangbin Feng, Weijia Shi, Yuyang Bai, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2023. Knowledge card: Filling llms' knowledge gaps with plug-in specialized language models. *arXiv*.

Parham Gohari, Matthew Hale, and Ufuk Topcu. 2023. Privacy-engineered value decomposition networks for cooperative multi-agent reinforcement learning. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 8038–8044. IEEE.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

Xiang Huo, Hao Huang, Katherine R Davis, H Vincent Poor, and Mingxi Liu. 2024. A review of scalable and privacy-preserving multi-agent frameworks for distributed energy resource control. *arXiv e-prints*, pages arXiv–2409.

Kemou Jiang, Xuan Cai, Zhiyong Cui, Aoyong Li, Yilong Ren, Haiyang Yu, Hao Yang, Daocheng Fu, Licheng Wen, and Pinlong Cai. 2024. Koma: Knowledge-driven multi-agent framework for autonomous driving with large language models. *IEEE Transactions on Intelligent Vehicles*.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826.

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik Siu Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae Won Park. 2024. Mdagents: An adaptive collaboration of llms for medical decision-making. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Magdalena Kossek and Margareta Stefanovic. 2024. Survey of recent results in privacy-preserving mechanisms for multi-agent systems. *Journal of Intelligent & Robotic Systems*, 110(3):129.

Binxu Li, Tiankai Yan, Yuanting Pan, Jie Luo, Ruiyang Ji, Jiayuan Ding, Zhe Xu, Shilong Liu, Haoyu Dong, Zihao Lin, et al. 2024. Mmedagent: Learning to use medical tools with multi-modal agent. *arXiv preprint arXiv:2407.02483*.

Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. 2021. A survey on

federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3347–3366.

Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. 2021. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1013–1023.

Qiang Meng, Feng Zhou, Hainan Ren, Tianshu Feng, Guochao Liu, and Yuanqing Lin. 2022. Improving federated learning face recognition via privacy-agnostic clusters. *arXiv preprint arXiv:2201.12467*.

Anudit Nagar, Cuong Tran, and Ferdinando Fioretto. 2021. A privacy-preserving and trustable multi-agent learning framework. *arXiv preprint arXiv:2106.01242*.

Longshuo Pan, Jian Wang, Hongyong Yang, Chuangchuang Zhang, and Li Liu. 2024. Privacy-preserving bipartite consensus of discrete multi-agent systems under event-triggered protocol. In *Chinese Intelligent Systems Conference*, pages 488–496. Springer.

Ashwinee Panda, Tong Wu, Jiachen Wang, and Prateek Mittal. 2023. Differentially private in-context learning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Toran Bruce Richards et al. 2023. Auto-gpt: An autonomous gpt-4 experiment. *Original-date*, 21:07Z.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*, 2(5):9.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Pratiksha Thaker, Yash Maurya, Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. 2024. Guardrail baselines for unlearning in llms. *arXiv*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Haotian Wang, Xiyuan Du, Weijiang Yu, Qianglong Chen, Kun Zhu, Zheng Chu, Lian Yan, and Yi Guan. 2025. Learning to break: Knowledge-enhanced reasoning in multi-agent debate system. *Neurocomputing*, 618:129063.

Qian Wang, Tianyu Wang, Qinbin Li, Jingsheng Liang, and Bingsheng He. 2024. Megaagent: A practical framework for autonomous cooperation in large-scale llm agent systems. *arXiv*.

Yaqi Wang, Jianquan Lu, Wei Xing Zheng, and Kaibo Shi. 2021. Privacy-preserving consensus for multi-agent systems via node decomposition strategy. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(8):3474–3484.

Z Wang, S Mao, W Wu, T Ge, F Wei, and H Ji. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona selfcollaboration. arxiv 2023. *arXiv preprint arXiv:2307.05300*.

Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. 2021. Fedgnn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023a. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Tong Wu, Ashwinee Panda, Jiachen T Wang, and Prateek Mittal. 2023b. Privacy-preserving in-context learning for large language models. *arXiv*.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. 2024. Tradingagents: Multi-agents llm financial trading framework. *arXiv preprint arXiv:2412.20138*.

Andrew C Yao. 1982. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE.

Chenduo Ying, Ning Zheng, Yiming Wu, Ming Xu, and Wen-An Zhang. 2023. Privacy-preserving adaptive resilient consensus for multiagent systems under cyberattacks. *IEEE Transactions on Industrial Informatics*, 20(2):1630–1640.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-hint prompting improves reasoning in large language models. *arXiv preprint arXiv:2304.09797*.

Jingwen Zhou, Qinghua Lu, Jieshan Chen, Liming Zhu, Xiwei Xu, Zhenchang Xing, and Stefan Harrer. 2024. A taxonomy of architecture options for foundation model-based agents: Analysis and decision model. *arXiv*.

Guy Zyskind, Tobin South, and Alex Pentland. 2023. Don't forget private retrieval: distributed private similarity search for large language models. *arXiv*.