



ELSEVIER

Contents lists available at ScienceDirect

High-Confidence Computing

journal homepage: www.sciencedirect.com/journal/high-confidence-computing

Review article

On protecting the data privacy of Large Language Models (LLMs) and LLM agents: A literature review[☆]Biwei Yan^a, Kun Li^a, Minghui Xu^{a,1,*}, Yueyan Dong^a, Yue Zhang^{b,1,*}, Zhaochun Ren^c, Xiuzhen Cheng^{a,1}^a School Computer Science and Technology, Shandong University, Qingdao 266237, China^b Department of Computer Science, Drexel University, Philadelphia 19104, USA^c Leiden Inst of Advanced Computer Science, Leiden University, Leiden 2333 CC, Netherlands

ARTICLE INFO

Article history:

Received 28 September 2024

Revised 20 December 2024

Accepted 25 December 2024

Available online 28 February 2025

Keywords:

Large Language Models (LLMs)

Security

Data privacy

Privacy protection

LLM agents

Survey

ABSTRACT

Large Language Models (LLMs) are complex artificial intelligence systems, which can understand, generate, and translate human languages. By analyzing large amounts of textual data, these models learn language patterns to perform tasks such as writing, conversation, and summarization. Agents built on LLMs (LLM agents) further extend these capabilities, allowing them to process user interactions and perform complex operations in diverse task environments. However, during the processing and generation of massive data, LLMs and LLM agents pose a risk of sensitive information leakage, potentially threatening data privacy. This paper aims to demonstrate data privacy issues associated with LLMs and LLM agents to facilitate a comprehensive understanding. Specifically, we conduct an in-depth survey about privacy threats, encompassing passive privacy leakage and active privacy attacks. Subsequently, we introduce the privacy protection mechanisms employed by LLMs and LLM agents and provide a detailed analysis of their effectiveness. Finally, we explore the privacy protection challenges for LLMs and LLM agents as well as outline potential directions for future developments in this domain.

© 2025 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

In recent years, Large Language Models (LLMs) have emerged as pivotal forces in the field of natural language processing [1–3], embodied AI [4–6], AI-generated content (AIGC) [7–9]. LLMs, trained on massive datasets, have the remarkable ability to generate human-like text, answer complex queries, and perform a myriad of language-related tasks with unprecedented accuracy and fluency. However, amidst the excitement surrounding LLM capabilities, concerns about data privacy have garnered increasing attention [10].

On one hand, LLMs may be subject to passive privacy leakage. LLMs often rely on vast amounts of data for training, including text from the internet, publicly available datasets, or proprietary sources. This data aggregation process can raise significant data

privacy concerns, especially when dealing with sensitive or personally identifiable information (PII) [11]. LLMs have been shown to have the potential for memorization of training data, raising concerns about inadvertent leakage of sensitive information during inference [12,13]. Even with techniques such as differential privacy or federated learning, which aim to mitigate privacy risks during training, residual traces of sensitive data may still persist within the model's parameters [14].

On the other hand, LLMs may be vulnerable to active privacy attacks. The deployment of fine-tuned LLMs in various applications introduces additional security challenges. Fine-tuning or adapting pre-trained LLMs to specific tasks may inadvertently expose them to the exploitation of vulnerabilities, potentially compromising the confidentiality, integrity, or availability of sensitive information [15]. For example, to bypass the model's inherent alignment, a prompting strategy was devised that induces GPT-3.5-turbo to “diverge” from producing conventional responses, instead training data [16]. Pre-existing vulnerabilities such as backdoor attacks, membership inference attacks, and model inversion attacks can be leveraged against pre-trained or fine-tuned models with the objective of illicitly acquiring sensitive data.

To portray the current situation, we outline the present state of research concerning privacy safeguards for LLMs in Fig. 1. Taking into account academic papers on privacy protection and the model list from Hugging Face, we have compiled a list of popular

[☆] This paper is an extended version of our conference paper presented at The 1st IEEE International Conference on Meta Computing (IEEE ICMC).

* Corresponding authors.

E-mail addresses: mhxu@sdu.edu.cn (M. Xu), yz899@drexel.edu (Y. Zhang).

¹ Given Xiuzhen Cheng's role as Editor-in-Chief, Minghui Xu's and Yue Zhang's roles as Editors of this journal, they had no involvement in the peer-review of this article and had no access to information regarding its peer review. Full responsibility for the editorial process for this article was delegated to Dr. Zhipeng Cai.

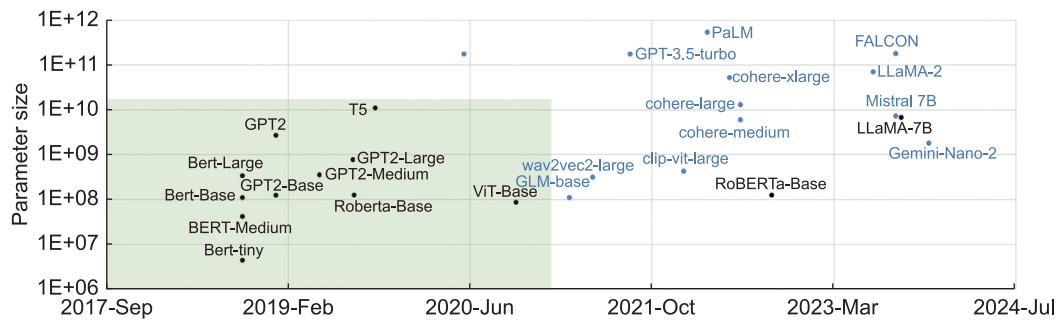


Fig. 1. The current state of research on privacy protection for LLMs is depicted. The horizontal axis represents the time of LLMs release, while the vertical axis represents the size of model parameters. Blue dots signify LLM instances not addressed in the literature pertaining to privacy protection, whereas black dots indicate those that have been examined in such literature. The green backdrop delineates the central cluster zone of LLMs with the potential to facilitate privacy protection.

LLMs in the figure. The timeline axis represents the release time of LLMs, while the vertical axis indicates the size of parameters. Blue data points signify LLMs that have received limited attention in the literature regarding privacy protection, while black data points indicate models studied alongside privacy protection. Currently, scholarly focus on data privacy in LLMs primarily revolves around well-known models of relatively smaller scale, like pre-2020 versions of the GPT-2 [17] and BERT [18] series. In contrast, recent releases of LLMs with larger parameter sizes have not been adequately scrutinized due to some models not being publicly available, and privacy protection technology lagging behind the rapid development of LLMs.

Moreover, to further achieve greater automation, agents built on LLMs have emerged (LLM Agent). LLM agent is an application of an LLM integrated with external tools, interfaces, and additional logic to perform specific tasks. LLM agents are typically applied in various domains, including but not limited to conversational assistants, code assistance, workflow automation, and security auditing. With the continuous advancement of LLMs, LLM agents have evolved from simple task execution tools into advanced systems that can support complex decision-making. Nonetheless, the widespread application of LLM agents further increases privacy risks due to their real-time processing of user inputs, interactions with external systems, and operation in complex environments. These agents usually need to collect, store, and analyze large amounts of data, raising significant challenges in terms of privacy protection. Existing research indicates that LLM agents are vulnerable to various potential attacks such as memory poisoning attacks. These attacks exploit vulnerabilities in the data processing and interaction stages of the LLM agents, potentially leading to the exposure of sensitive information, biased task decisions, or malicious behaviors, thereby severely compromising their reliability and security.

In this paper, we extensively investigate data privacy concerns in LLMs and LLM agents, specifically exploring potential privacy threats from two aspects: privacy leakage and privacy attacks. Furthermore, we analyze the current research on the privacy of LLMs and their agents, providing a comprehensive review of the corresponding countermeasures according to the three main stages of LLM development: pre-training, fine-tuning, and inference. Meanwhile, we also introduce the privacy protection approaches for LLM agents. Our contributions are summarized as follows:

- We conduct a comprehensive survey of the academic literature on privacy threats in LLMs and LLM agents, categorizing them into two groups: privacy leakage and privacy attacks.
- Our survey analyzes privacy protection methodologies for LLMs and LLM agents, categorized by developmental stages. For LLMs, we introduce privacy protections during pre-training, fine-tuning, and inference. For LLM agents, we

focus on input, data preprocessing, and output. In each category, we introduce techniques at a high level, explain their application, and provide a detailed literature review, aiming to guide developers in implementing cutting-edge techniques to safeguard LLMs and LLM agents.

2. Related work

In this section, we first introduce existing surveys about the development and evaluation of LLMs. Furthermore, we introduce the research of LLM agents, and then elaborate the most related work addressing the privacy and security issues in LLMs and LLM agents. Finally, we summarize the research of our survey.

2.1. Surveys on LLMs evaluation

Currently, some works have surveyed the development and evaluation of LLMs. These studies typically cover architectural improvements of LLMs (such as the GPT series, BERT, Transformers [19–27]). For example, Li et al. [19] focused on integrating LLM with intelligent personal assistants (IPAs) to improve personal assistance capabilities. It delves into the architecture, capabilities, efficiency, and security aspects of these agents. Zhao et al. [21] focused on four key aspects of LLMs: pre-training, adaptation tuning, utilization, and capacity evaluation. It provides a thorough background on LLMs, including terminologies and techniques. Naveed et al. [28] provided an extensive analysis of LLMs, covering their architecture, training, applications, and challenges. It dives into detailed aspects of LLMs like pre-training, fine-tuning, and evaluation, while also discussing various LLM applications in different fields. Hadi et al. [29] introduced a thorough overview of LLMs, discussing their history, training, and applications in various fields like medicine, education, finance, and engineering. It examines the technical aspects, challenges, and future potential of LLMs, including ethical considerations and computational requirements.

To understand the capabilities and limitations of LLMs in various applications, some works have conducted comprehensive measurements on these LLMs [20,30–32]. Chang et al. [20] offered a comprehensive analysis of the methods and criteria for evaluating LLMs. It discusses various aspects including tasks to evaluate, datasets, benchmarks, and evaluation techniques. Guo et al. [30] emphasized the need for a comprehensive evaluation of LLMs in various dimensions, such as knowledge and capability evaluation, alignment evaluation, security considerations, and applications in the specialized domain. In [31], Liu et al. examined the alignment of LLMs with human values and social norms. It proposes a detailed taxonomy to evaluate LLM trustworthiness on various dimensions such as reliability, safety, fairness, resistance to misuse, explainability, adherence to social norms, and robustness. Yuan

et al. [32] focused on the various challenges faced when deploying LLMs on hardware. The paper begins by introducing the basic architecture and inference process of LLMs, then proposes a new framework based on the Roofline model to systematically analyze and understand the bottlenecks in LLM inference.

2.2. Surveys on LLM agents

In this section, we comprehensively review the research on LLM agents, including the development of agents' applications.

Scholars have explored various applications of LLMs, demonstrating how these models can be efficiently integrated into LLM agents [19,33–37]. Guo et al. [33] primarily explored the application of LLMs in multi-agent systems and summarized the progress, challenges, and future research directions in this field. Moreover, the paper briefly introduces the fundamentals, architecture, interaction methods among agents, and how they acquire capabilities in multi-agent systems based on large language models (LLM-MA). Li et al. [19] have extensively studied personal LLM agents, intelligent personal assistants based on LLMs, which are tailored to deeply integrate with personal data and devices for enhancing personal assistance. The research critically examines the architecture, capabilities, efficiency, and security of these agents. Feng et al. [34] introduced a novel LLM agent framework named AGILE to perform complex conversational tasks. This framework enhances performance by utilizing LLMs, memory, tools, and interactions with experts. Zhang et al. [36] reviewed memory mechanisms in LLM agents, highlighting their role in facilitating agent–environment interactions. It categorized memory sources, evaluated both textual and parametric memory forms, and explored their impact on agent development. Additionally, the paper identified existing research gaps and proposed future directions to enhance memory functionality in these agents. In [37], Cheng et al. examined the use of LLMs as intelligent agents, detailing their integration in both single and multi-agent systems. It discussed the enhancement of cognitive and planning abilities through LLMs across various applications. The study also forecasted the impact of ongoing advancements in AI and natural language processing on the development of these agents.

2.3. Surveys on LLMs and LLM agents privacy

Since the training of LLMs relies on a substantial amount of data, which usually includes sensitive information. Therefore, LLMs face challenges in handling privacy and security issues [10, 38–47]. Yao et al. [10] comprehensively investigated the security and privacy of LLMs, and conducted an extensive review of the literature on LLMs from three aspects: beneficial security applications (such as vulnerability detection, secure code generation), adverse effects (e.g., phishing attacks, social engineering), and vulnerabilities (e.g., jailbreak attacks, prompt attacks), as well as corresponding defense measures. Li et al. [38] explored privacy concerns in LLMs, categorizing privacy attacks and describing defense strategies. It also explores future research directions to improve privacy in LLM. Neel et al. [39] explored the privacy risks associated with LLMs, focusing on issues such as the memory of sensitive data and various privacy attacks. Review mitigation techniques and highlight the current state of privacy research in LLMs. However, they mainly focus on work that red-teams models to highlight privacy attacks. Wang et al. [46] explored the distinct security and privacy challenges that arise throughout the lifecycle of LLMs. The paper categorizes threats into five main stages: pre-training, fine-tuning, retrieval-augmented generation systems, deployment, and LLM agents, providing a detailed analysis of the potential vulnerabilities at each stage. Kibriya et al. [48] examined the privacy concerns associated with

LLMs. It categorized these concerns into issues during the training and inference phases, which could lead to re-identification risks. The paper also discussed the challenges of implementing privacy-preserving mechanisms and the interaction between ethical issues, legal requirements, and technology developments. In [49], Wang et al. reviewed the development of autonomous agents using LLMs and proposed a unified framework for constructing these agents and exploring their applications in various fields. Yan et al. [50] provided a thorough investigation of data privacy issues in LLMs, particularly focusing on passive privacy leakage and active privacy attacks.

Besides, we studied several security and privacy challenges emerging from LLM agents. He et al. [51] provided a comprehensive review of the newly emerged security and privacy issues faced by LLM agents. Wang et al. [52] provided a concise overview of LLM agents, which explored the architecture, functionality, and applications of these agents, particularly highlighting their integration capabilities in various environments. Deng et al. [53] explored emerging security threats to AI agents, categorizing them into the unpredictability of multi-step user inputs, complexity in internal executions, variability of operational environments, and interactions with untrusted entities. Zhang et al. [54] examined privacy awareness, preferences, and trust issues when people used LLM agents, highlighting privacy leakage risks associated with agent automation. The study proposed design strategies, including personalized privacy settings and contextual privacy decision making, to help users more effectively oversee and manage the privacy behavior of agents.

3. Background

To understand the methodologies used to protect the privacy of LLMs, we first need to grasp the underlying principles and technologies that drive LLMs and their application in various fields. This section provides an overview of LLMs, the concept of LLM agents, and compares traditional LLMs and their agent-based counterparts.

3.1. Large language models (LLMs)

LLMs are super-large deep learning models pre-trained on vast amounts of data, containing tens of billions to trillions of parameters. They construct extensive unsupervised training based on these parameters, enabling them to more accurately learn patterns and structures of natural language, thereby understanding and generating natural language texts. Compared to traditional NLP models, LLMs demonstrate a better understanding and generation of natural texts and also exhibit certain logical thinking and reasoning abilities, which are widely used in programming [55], vulnerability detection [56], and medical text analysis [57]. In 2017, Vaswani et al. [58] introduced the Transformer architecture, which uses parallel processing and attention mechanisms to provide an effective method for processing sequential data (especially text). This significantly enhances the efficiency of dealing with sequential data and supports more efficient training on large datasets, fostering the rapid development of LLMs such as the GPT series, BERT, and Transformer models. Fig. 2 illustrates the LLM training and inference process, including the flow of different data sources and types through the pre-training, fine-tuning, and inference stages. Unlabeled data is used for pre-training to generate the foundational model, labeled demonstration data is used for fine-tuning to adapt the model to specific tasks, and user prompts are utilized in the inference stage to produce the final output.

Generally, the training of LLMs primarily includes two key stages: pre-training and fine-tuning.

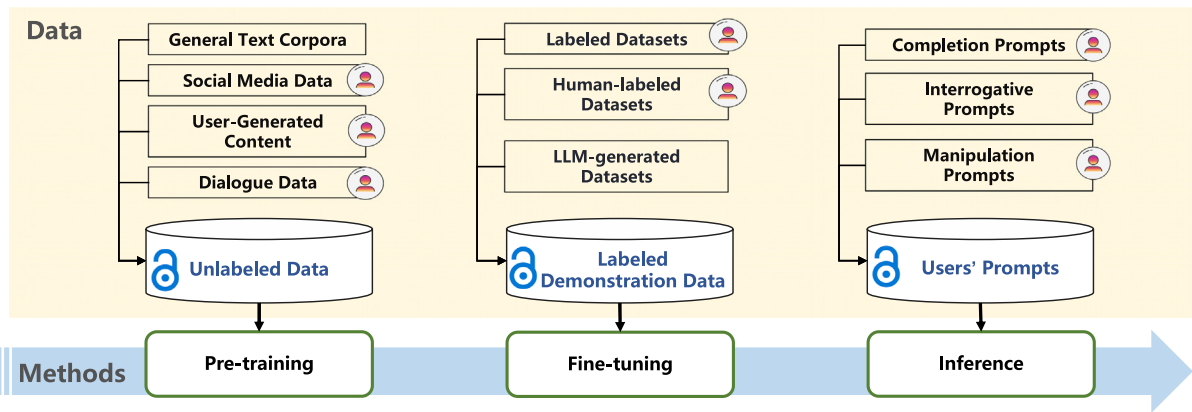


Fig. 2. The process of data propagation during both the training and inference stages of LLMs.

- **Pre-training:** At this stage, the model is typically trained on a very large and diverse dataset. These datasets may include texts from a variety of sources such as the Internet, books and news, or large text datasets published by many organizations and research institutions for academic research. E.g. general text corpora, social media data, user-generated content, and dialogue data. For example, GPT-3, developed by OpenAI, was pre-trained using CommonCrawl, constituting 45TB of compressed plaintext before filtering [59]. Regarding multimodal LLMs, CLIP's training dataset encompasses 400 million pairs of images and text, while Stable Diffusion was trained on a dataset consisting of two billion examples sourced from LAION-2B [60]. The purpose of pre-training is to enable the model to learn a wide range of language patterns, structures, and knowledge. Through this process, the model acquires a broad ability to understand language, including understanding vocabulary, grammar, and even some common sense. This stage does not focus on any specific task but rather provides a general foundation for language understanding.
- **Fine-tuning:** The fine-tuning stage is carried out on the basis of a pre-trained model, with the goal of better adapting the model to specific tasks or domains. During this phase, the model is trained on a smaller, more specific dataset that is closely related to the target task or domain. The datasets are usually sourced from websites and forums of specific professional fields such as the medical, legal, technological, and other professional communities, which mainly consist of labeled demonstration data such as labeled datasets, human-labeled datasets, and LLM-generated datasets. The datasets available for fine-tuning may be relatively small, typically ranging from a few hundred to a few thousand text samples. Through fine-tuning, the model learns the characteristics and details specific to the task.

The advantage of this two-stage training method is that it combines the breadth of general language understanding (through pre-training) with the depth of adaptability to specific tasks (through fine-tuning), which enables the model to exhibit higher accuracy and efficiency when dealing with a variety of complex, domain-specific tasks. After the model has been trained and fine-tuned, the inference stage can be performed.

- **Inference:** In this phase, the trained model is used to make a prediction or decision. This includes processing input data (such as users' prompts), using the model to compute outputs, and possibly post-processing to satisfy the specific needs. The primary purpose of inference is to solve real-world problems based on the knowledge learned by the model, such as automated responses, image recognition, or other forms of data analysis.

3.2. LLM agents

LLM agents can perceive their surroundings and respond to changes through autonomous decision making, thereby achieving specific goals or completing tasks. Generally, An LLM agent typically includes the following main components, as shown in Fig. 3:

- **Brain (LLMs):** As the brain of the LLM agent, it is responsible for processing information, making decisions, and carrying out reasoning and planning. LLMs, trained on vast amounts of human behavior data, enable LLM agents to decompose complex problems and engage in natural language interactions.
- **Plan:** The plan module is responsible for breaking down complex tasks into multiple independently solvable steps or subtasks to gradually fulfill the user's request. The primary function of this module is to improve the agent's reasoning capability by decomposing tasks, allowing it to gain a clearer understanding of different aspects of the problem and ultimately find a more reliable solution.
- **Memory:** In LLM agent systems, memory is typically categorized into short-term and long-term memory. Short-term memory enables the agent to retain key information relevant to the current task for a short period, facilitating efficient task completion. In contrast, long-term memory relies on external storage and rapid retrieval mechanisms, allowing the agent to store and access large amounts of information as needed, supporting more complex and cross-temporal tasks.
- **Tools:** LLM agents learn to utilize various tools and interfaces, enabling them to access the latest information, execute code, and retrieve proprietary data, thereby achieving more efficient and precise task execution.

These components work together, enabling LLM agents to perform a variety of tasks in complex environments, gradually approaching the capabilities of AGI. The LLMs, serving as the brains of the LLM agents, play a crucial role in decision-making and information processing.

3.3. Comparison between LLMs and LLM agents

LLMs are extensively pre-trained or fine-tuning models, focusing on natural language understanding and generation. In contrast, LLM agents are intelligent agents based on LLMs, which also have the ability of logical reasoning, task execution and tool calling, enabling them to complete complex tasks with greater autonomy. Essentially, LLM is the engine and this engine powers

Table 1
Comparison of LLMs and LLM Agents.

Aspect	LLMs	LLM Agents
Natural Language Understanding and Generation	✓	✓
Task Decomposition and Planning	✗	✓
Tool Utilization	✗	✓
Memory Capability	Short-term memory	Short-term & long-term memory
Multimodal Support	✗	✓
Autonomous Decision-Making	✗	✓
Interaction Capability	Question-and-answer interactions	Multi-turn interactions
Complex Task Execution	✗	✓
Learning Capability	Limited	Continuous learning
Adversarial Attack & Defense Capability	Weak	Strong
Examples	GPT-4, Bert, LaMDA	ChatGPT, MetaGPT, Copilot

The ✓ indicates that the functionality is supported.
The ✗ indicates that the functionality is not supported.

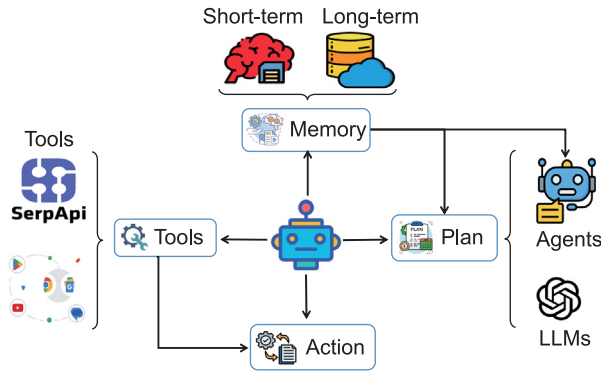


Fig. 3. The structure of LLM agents.

the LLM agent. For example, OpenAI’s GPT-4 is an advanced LLM that powers ChatGPT, an LLM agent that can engage in multi-turn conversations, provide detailed answers, and execute tasks such as code generation and document summarization. Similarly, Google’s LaMDA serves as the LLM engine for its intelligent conversational agent Google Bard, which can enable more dynamic and context-aware interactions. This difference mainly lies in the functionality of LLMs and LLM agents. As a basic model, LLMs provide powerful language processing capabilities, while LLM agents expand these capabilities to solve more complex real-world problems, showing greater flexibility and intelligence. We can see in Table 1 that LLMs can handle single-step tasks such as text generation, translation, and basic logical reasoning, but lack initiative and environmental interaction capabilities. Comparing LLMs, LLM agents add task decomposition and planning, tool calls, memory management, multimodal support, and autonomous decision-making capabilities, enabling them to perform complex tasks in a dynamic environment, actively initiate interactions, and combine external tools to achieve goals, showing capabilities close to AGI.

The structure of LLM agents is more complex, often using LLMs as the “brain” for language processing but also incorporating other components (e.g. plan, memory and tools) that allow for continuous autonomous operation. Next, we introduce the LLM agents in detail.

4. Scope, methodology, and overview

In the following, we outline the scope of this survey, the methods employed for data collection and analysis, as well as the structure of the paper.

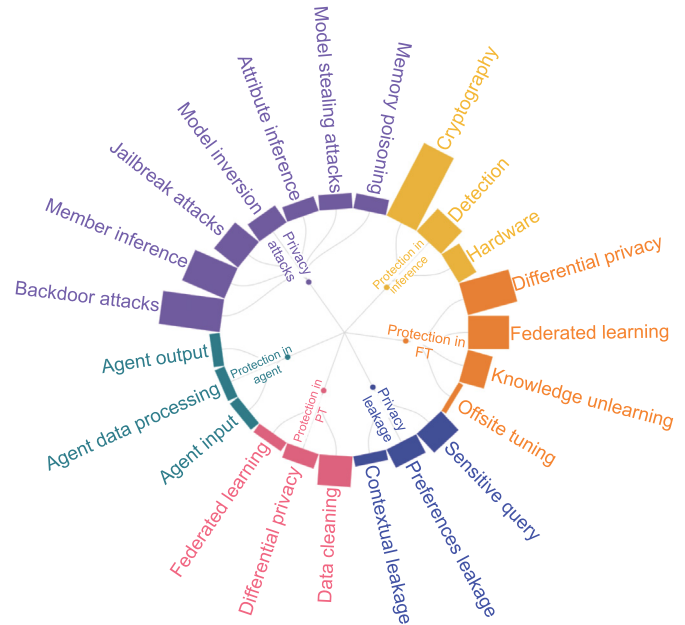


Fig. 4. The distribution of research papers concerning the data privacy in LLMs and LLM Agents. “PT” and “FT” represent abbreviations for Pre-Training and Fine-Tuning, respectively.

4.1. Scope

Our paper is dedicated to conducting a comprehensive literature review in the field of data privacy for LLMs, organizing and reviewing existing research. We conduct a comprehensive and in-depth privacy analysis, including privacy leakage, privacy attacks and privacy protection methods at different stages in LLMs and LLM agents. Our focus is not only on the implementation details of these technologies but also on a deep exploration of their effectiveness in protecting privacy, as well as their potential limitations.

4.2. Methodology

We collected and organized the main the current research in LLMs and LLM agents: data collection, structure and analysis.

Data Collection: To comprehensively understand the landscape of data privacy concerns in LLMs, we executed a structured literature search on Google Scholar. The results are summarized in Fig. 4, wherein we categorized the retrieved literature into distinct themes. From the 118 collected papers, we identified 51 that specifically highlight the privacy threats confronting LLMs.

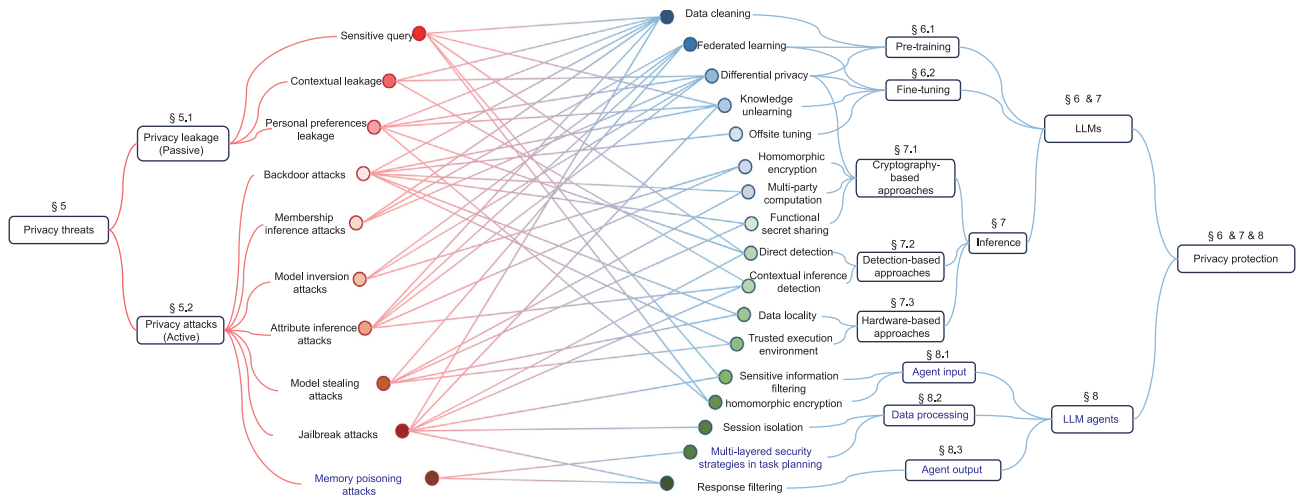


Fig. 5. Privacy threats, protection, and their defensive correlations.

Within this subset, a division reveals that 11 papers focus on privacy leakage, while the remaining 40 delve into various privacy attacks. Additionally, we found 67 papers dedicated to exploring privacy protection strategies for LLMs. We classified them according to different phases: 11 during pre-training, 24 during fine tuning, 26 in inference phase, and 6 in LLM Agents. An analysis of publication trends shows that the majority of these papers, representing 44.92%, were published in 2023, with only 30 released in between 2021 and 2022, indicating a significant recent interest in the topic. Notably, there are also 31 cutting-edge studies from 2024, which underscores the ongoing and dynamic nature in this crucial area of research.

Structure and Analysis: Fig. 5 presents the organizational structure of this study, which outlines the current privacy threats faced by LLMs and LLM agents, and their corresponding protections as well as the relevance between privacy threats and defense technologies. In the section on privacy threats, this paper reviews existing research from two dimensions: privacy attacks and privacy leakage, detailing common attack methods and instances of privacy leakage in LLMs and LLM agents. Regarding privacy protection approaches, we systematically summarize them according to the three stages of LLMs: pre-training, fine-tuning, and inference. And we summarize the key privacy protection technologies, including data sanitization, federated learning, differential privacy, homomorphic encryption, and secure multi-party computation. Then, we introduce the privacy protection method and its corresponding technologies of LLM agents from three stages: agent input, data processing and agent output. Finally, we establish a connection between these key technologies and the privacy threats they may defend against, providing a framework for understanding the data privacy in LLMs and LLM agents.

4.3. Overview

Fig. 5 depicts the privacy issues in detail, including privacy leakage and attacks, as well as the tailored defensive technologies deployed at various phases of the LLM lifecycle, including the pre-training, fine-tuning and inference stages. Meanwhile, we outline the privacy protection methods of LLM agents from three aspects: Agent input, data processing, and agent output.

- **Privacy Threats** (Section 5): We first conduct a literature review on privacy threats against LLMs and LLM agents. Based on whether the attackers are active or passive, we further categorize the threats into two groups: privacy leakage,

where the attackers passively collect sensitive information due to vulnerabilities and privacy attacks, where the attackers actively break LLMs and LLM agents to access sensitive information.

- **Privacy Protections** (Section 6 & Section 7 & Section 8): Based on where privacy protection is located, we can group them into three categories: privacy protection in pre-training (Section 6.1), privacy protection in fine-tuning (Section 6.2), privacy protection in inferences (Section 7), and privacy protection in LLM agent (Section 8). Among them, privacy protection in inferences can be further grouped based on the methods adopted (e.g., whether it is a cryptography-based approach). Similarly, privacy protection in LLM agents can be categorized into three stages according to their lifestyle. In each of these protections, we first introduce the techniques at a high level; then, we explain how they can be used in LLMs and LLM agents (see those **Tech Tips**), and finally, we provide a detailed literature review.

5. Privacy leakage and privacy attacks

We undertake a literature review focusing on privacy threats against LLMs and LLM agents. We categorize these threats into two groups based on the attackers' activity: privacy leakage, where attackers passively collect sensitive information due to vulnerabilities, and privacy attacks, where attackers actively breach LLMs or LLM agents to access sensitive information.

5.1. Privacy leakage (passive)

5.1.1. Sensitive query

Users may input queries containing sensitive or personally identifiable information (PII) into LLMs. For example, asking questions about medical conditions, financial situations, or personal relationships could reveal private details about the user's life. If users input sensitive information as prompts, there arise concerns regarding data privacy [61,62]. For example, Samsung Electronics staff provided sensitive corporate data through the prompts. Besides, various LLM plugins also raise privacy concerns of user's sensitive data. Iqbal et al. [63] proposed a systematic framework to evaluate the security, privacy, and safety of third-party plugins integrated into LLM platforms. Some plugins were found to collect excessive user data, including personal and sensitive information. Some plugins did not provide clear details on how they use user data, potentially violating privacy policies.

Furthermore, as LLMs become widely integrated into various applications, their agents raise additional concerns regarding privacy protection [64,65]. Zhang et al. [64] investigated privacy leakage issues when users interacted with LLM-based conversational agents, particularly focusing on the disclosure of sensitive information in high-risk domains. The study found that the memorization characteristics of LLMs and human-like interactions could lead users to unintentionally disclose personal information, thereby increasing the risk of privacy leakage. Bagdasaryan et al. [65] introduced AirGapAgent, a privacy-preserving conversational agent designed to prevent the leakage of sensitive user information during irrelevant tasks through context isolation and data minimization. They proposed a “context hijacking” attack model, where malicious third-party applications manipulate the interaction context to deceive LLM-driven agents into disclosing private data unrelated to the task.

5.1.2. Contextual leakage

Even seemingly innocuous queries could indirectly reveal sensitive information about the user when combined with other contextual factors. For instance, asking about nearby landmarks or local events could inadvertently disclose the user’s location or activities. Over time, repeated interactions with the model could lead to the accumulation of enough information to uniquely identify the user, posing a risk to privacy. The study [12] focuses on the capabilities of LLMs to infer personal attributes from text, particularly in the context of privacy concerns and the threat of privacy-invasive chatbots. They evaluated LLMs’ ability to infer personal attributes (like location, occupation, age, gender, etc.) from text on the PersonalReddit dataset, containing 520 profiles with 5814 comments. They evaluated 9 state-of-the-art LLMs on the PR dataset, with GPT-4 achieving top-1 accuracy of 84.6% and top-3 accuracy of 95.1%.

In LLM agents, the risk of context leakage may also occur during multi-turn interactions with users. Since LLM agents typically rely on user context information to provide personalized responses, this dependence on context can lead to privacy leakage. Li et al. [66] proposed a framework called MobAgent, which simulates individuals’ daily mobility behaviors by integrating personal attributes and real urban environments to generate travel trajectories that reflect real-world social and spatial constraints. However, during multi-turn generation, MobAgent utilizes contextual information, such as previously generated travel diaries or historical interaction data, to guide subsequent outputs. This could lead to the unintended disclosure of sensitive information related to the context.

5.1.3. Personal preferences leakage

LLMs may infer personal preferences, interests, or characteristics of users based on their queries and interactions. This could result in targeted advertisements, personalized recommendations, or other tailored content that may reveal private aspects of the user’s life. For example, LLMs represent a significant asset to recommender systems, offering advantages in delivering personalized recommendations [67]. Besides, these models have the potential to refine or establish new methodologies for sequential recommendation [68], which could inadvertently reveal users’ personal preferences, thereby raising privacy concerns.

During the utilization of agents based on LLMs, individuals may unintentionally disclose their privacy, whether through direct or indirect means. Beyond the direct provision of sensitive information, providers of services can extrapolate intricate user attributes and preferences, thereby gaining access to sensitive data via data analysis methods. For instance, Gao et al. [69] proposed a framework, PRELUDE (PREference Learning from User’s Direct Edits), which infers user preferences from historical edit

data. However, this approach could lead to data privacy concerns, as the collection and analysis of historical data may inadvertently expose sensitive information about the user’s intentions, preferences, or personal data. Moreover, MobAgent [66] also uses LLMs to infer user preferences and behavior patterns, which may cause the model to “remember” specific users’ sensitive information during content generation.

5.2. Privacy attacks (active)

5.2.1. Backdoor attacks (data poisoning attacks)

Backdoor attacks involve embedding triggers in training data, plugins, or reasoning processes, enabling attackers to manipulate the model under specific conditions, resulting in abnormal behavior or the leakage of sensitive information. These attacks typically occur during the pre-training and fine-tuning stages of LLMs and also exist in LLM agents.

During the pre-training phase, adversaries can manipulate the training data by introducing poisoned examples, thereby embedding backdoors into the models. This tainted training data, once disseminated on the internet, can be unwittingly procured and used by developers for training their models. These backdoors allow adversaries to exfiltrate sensitive or private information processed by the LLMs, including personal data, confidential documents, or proprietary information [70]. For instance, an LLM containing a backdoor might be used to process private medical records submitted by users. When adversaries trigger the backdoor through specific inputs, the model could disclose users’ sensitive medical information to the attackers, severely violating users’ privacy rights. Moreover, Yang et al. [71] shed light on a critical security vulnerability in NLP models, introducing a data-free backdoor attack that could subvert the integrity of word embeddings by altering a single embedding vector. This attack method is not only difficult to detect but can also bypass existing security mechanisms, further exacerbating the risk of privacy breaches. Similarly, POISONPROMPT [72] as a novel backdoor attack strategy, can compromise its capability to compromise both hard and soft prompt-based LLMs, further demonstrating the wide applicability of backdoor attacks in undermining privacy. Furthermore, Huang et al. [73] introduced a stealthy Composite Backdoor Attack (CBA) that scatters multiple trigger keys across different prompt components. CBA ensures activation only when all triggers are present, demonstrating high effectiveness in NLP and multimodal tasks while maintaining model accuracy.

During the fine-tuning phase, adversaries can also inject poisoned or adversarial examples into the dataset to manipulate the behavior of LLMs. These poisoned examples could introduce biases or vulnerabilities into the model, leading to compromised performance or biased outputs that violate privacy and fairness principles. Research by Wan et al. [74] has revealed that instruction-tuned LLMs are vulnerable to backdoor attacks where adversaries can manipulate model behavior by tainting training datasets with malicious examples. Similarly, Xu et al. [75] demonstrated that attackers could subvert model behavior by interspersing legitimate data with malicious instructions, achieving high success rates of exploitation among various NLP datasets. For instance, an adversary could use a backdoor to manipulate the model’s responses to sensitive inquiries, such as requests for personal information or financial advice, leading to compromised privacy and potential financial loss for users. Moreover, Yan et al. [76] have shown that it is possible for adversaries to implant Virtual Prompt Injection (VPI) backdoors into models through tainted instruction tuning data, granting them the capability to finely control the model’s outputs in response to carefully chosen triggers.

In the context of LLM agents, backdoor attacks become even more dangerous. Adversaries can embed triggers in the training

data, plugins, or reasoning process of LLM agents, causing them to perform attacker-defined malicious actions when specific conditions are met [77–80]. Dong et al. [77] implanted backdoors into LLM plugins (such as low-rank adapters, LoRA), enabling these plugins to control the agent's output behavior when loaded into LLM agents by triggering specific inputs or conditions. Essentially, the backdoor serves as a clandestine channel for data extraction, compromising the confidentiality of the information handled by the agent. For instance, a backdoor embedded in a plugin might allow an attacker to intercept and steal personal data, such as user queries or intermediate reasoning results, when triggered by specific inputs. Wang et al. [78] presented BadAgent, which introduced hidden backdoor attacks via triggers in training data, demonstrating the risks associated with untrusted models and data sources. This approach underscores how backdoors can be inserted at various stages of the machine learning pipeline, from data collection to model training, thereby undermining the privacy of users whose data is involved. Yang et al. [79] investigated poisoning attacks on LLM agents, including Query-Attack, where triggers were embedded in user queries; Observation-Attack, where triggers were planted in intermediate observations; and Thought-Attack, which modified intermediate reasoning processes while keeping the final output unaffected. In [81], Zhang et al. embedded triggers and hidden malicious reasoning steps into the plan demonstrations in system prompts, enabling the agent to execute predefined malicious actions, such as invoking unauthorized tools or producing incorrect information, when specific input conditions were detected. These actions not only compromise the agent's reliability but also pose a severe threat to data privacy, as they can lead to unauthorized data disclosure or manipulation.

5.2.2. Membership Inference Attacks

Membership Inference Attacks typically occur during the pre-training and fine-tuning stages of LLMs, aiming to reveal whether specific data samples have been included in the model's training set. These attacks threaten the privacy of large-scale data used during pre-training and exhibit significant vulnerabilities during fine-tuning, particularly when LLMs are customized for specific domains or tasks.

In the pre-training stage, an adversary attempts to determine whether a specific individual's data was included in the training dataset used to train an LLM [82,83]. The attacker can infer whether certain data samples were part of the training data by analyzing the model's outputs or responses to queries. This can lead to privacy breaches if sensitive information about individuals is inferred from the model's behavior. A study by Mireshghallah et al. [84] has highlighted the high susceptibility of Masked Language Models (MLMs) to privacy attacks, demonstrating this through likelihood ratio membership inference attacks that utilize an additional reference MLM. However, considering the unrealistic assumption of reference-based models, Mattern et al. [85] proposed an alternative method known as neighborhood attacks, which compare scores with synthetic texts. In another development, Shi et al. [86] introduced WIKIMIA benchmark and MIN-K PROB method, which they claimed improved detection by 7.4% over previous methods. Despite these advancements, Duan et al. [87] evaluated membership inference attacks on the pre-training data of LLMs trained on the Pile and found that the success rates of the aforementioned attack methods were limited due to the combination of large datasets and few training iterations, as well as a fuzzy boundary between members and non-members.

In the fine-tuning stage, membership inference attacks aim to reveal whether specific data samples have been incorporated into the training set of the model. In the context of LLM fine-tuning, adversaries may discern patterns that suggest whether

those inputs were part of the training data by meticulously analyzing the model's responses to certain inputs. Accurate execution of such an inference by attackers could lead to the compromise of the model's training data confidentiality. Mireshghallah et al. [88] conducted an empirical investigation that examined significant variation in vulnerability to membership inference of different fine-tuning methods for LLMs. Their findings indicated fine-tuning model heads proves most susceptible, while using smaller adapters shows reduced attack susceptibility. Moreover, Jagannatha et al. [89] focused on fine-tuned clinical language models (CLMs) and their exposure to MIAs. They demonstrated that the scale of the model plays a crucial role in its privacy risks, with smaller models generally exhibiting lower vulnerability compared to larger architectures. Building on these insights, Fu et al. [90] introduced a novel approach to MIA in fine-tuned LLMs. Their proposed method, Self-calibrated Probabilistic Variation (SPV-MIA), leverages memorization rather than overfitting as a reliable indicator of membership. Additionally, they presented a self-prompt strategy for constructing a comparable dataset for the reference model, aiming to enhance the practicality and effectiveness of MIAs against fine-tuned LLMs.

5.2.3. Model inversion (data reconstruction) attacks

In a model inversion attack, an adversary attempts to reconstruct or reverse-engineer the training data used to train an LLM based on its outputs or internal representations. By analyzing the model's parameters, gradients, or generated text, the attacker aims to recover sensitive information contained in the training data, such as personal communications, financial records, or proprietary documents. Song et al. [91] demonstrated this through the development of such an attack. And the study by Carlini et al. [92] on GPT-2 demonstrates that adversaries can extract individual training examples through training data extraction attacks. Following this, Lehman et al. [93] further investigated the risk of model inversion attacks on a BERT model trained on sensitive EHR data. Surprisingly, they found that simple probing methods failed to extract sensitive information, indicating a potential safety margin for releasing such model weights. However, *Text Revealer* [94] was designed by Zhang et al., which is the first model inversion attack specifically designed for reconstructing private texts from transformer-based text classification models. Its attack leverages external datasets and GPT-2 to generate fluent, domain-specific text, optimizing perturbations to the hidden state based on feedback from the target model.

5.2.4. Attribute inference attacks

Attribute inference attacks involve inferring sensitive attributes or characteristics of individuals from fine-tuned LLMs. For example, an attacker may attempt to infer demographic information, such as age, gender, or ethnicity, based on the language patterns or topics discussed in the model's generated text [95]. This can lead to privacy violations and discrimination against individuals based on inferred attributes. In a comprehensive study, Pan et al. [96] systematically examined the privacy risks associated with 8 state-of-the-art language models. Their examination is anchored on 4 diverse case studies that focus on the threat of attribute inference attacks. The findings are compelling: these state-of-the-art models are indeed susceptible to revealing sensitive details, which include personal identifiers such as identity, genetic information, health data, and geographical locations. This vulnerability stems from the potential for adversaries to reverse-engineer the embeddings within these models. Building on this concern, Staab et al. [12] employed Reddit profiles to showcase that LLMs can accurately infer a variety of personal attributes. Remarkably, these models surpass human performance in terms of both efficiency and speed, underscoring the urgent need for effective privacy safeguards in model development.

5.2.5. Model stealing attacks

Adversaries may attempt to steal or replicate fine-tuned models trained on proprietary or sensitive datasets. By querying the model and observing its responses, adversaries can extract information about the model's parameters or internal representations, enabling them to reconstruct or replicate the model without access to the original training data. Krishna et al. [97] demonstrated the feasibility of model stealing attacks in NLP, showing that adversaries can reconstruct victim models using only random word sequences and task-specific heuristics, without requiring real training data. This exploit is enabled by the widespread use of transfer learning methods in NLP. And then, Truong et al. [98] advanced the field with their proposal of data-free model stealing techniques. These methods overcome the need for surrogate datasets, enabling accurate replication of valuable models with limited queries. Besides, Sha et al. [99] introduced a novel prompt stealing attack against LLMs, leveraging generated answers to reconstruct well-designed prompts. It involves a two-pronged approach: a parameter extractor dissects prompt types and characteristics, while a prompt reconstructor generates reverse-engineered prompts with notable efficacy.

5.2.6. Jailbreak attacks

Jailbreak attacks enable LLMs or LLM agents to bypass original security restrictions by designing specific prompts or context inductions, thereby generating sensitive or illegal content and causing privacy leakage.

In terms of LLMs, if LLMs are connected to sensitive data (such as users' medical records, financial data, or personal conversation history), attackers can leverage jailbreak attacks to manipulate the model into revealing this information. Li et al. [100] bypassed security mechanisms by deconstructing prompt structures instead of directly inducing the model to generate prohibited content. Zhou et al. [101] explored a novel jailbreak attack method called DSN (Don't Say No), which bypasses the built-in security restrictions of LLMs by suppressing refusal generation. Chu et al. [102] proposed a systematic framework for evaluating and classifying jailbreak attacks faced by large language models (LLMs). Guo et al. [103] proposed a novel framework called COLD-Attack, which introduced controllable text generation techniques into jailbreak attack research to generate jailbreak attacks with stealthiness and controllability.

Some LLM agents may have "memory" or external knowledge retrieval capabilities such as Retrieval-Augmented Generation (RAG). Through jailbreak attacks, attackers can manipulate the model to output stored user information without authorization. Wang et al. [104] proposed the RLTA framework, which uses reinforcement learning-driven LLM agents to automatically generate malicious prompts, enabling targeted attacks on LLMs, such as Trojan detection and jailbreak attacks. The RLTA framework can manipulate the output of target LLMs in a black-box environment, exposing potential security and privacy risks within these models. Dong et al. [105] presented Atlas, a framework that employs a multi-agent system to conduct jailbreak attacks on text-to-image (T2I) models by iteratively generating and selecting prompts to bypass safety filters. The collaborative effort of mutation and selection agents in Atlas efficiently identifies and exploits security vulnerabilities in T2I models.

5.2.7. Memory poisoning attacks

Memory poisoning attacks refer to the act of injecting malicious or misleading data into an agent's memory module or external knowledge base (e.g., RAG), thereby disrupting the agent's decision-making process and behavior, leading it to perform incorrect or malicious actions in future tasks [106–108]. This attack may falsely mark sensitive data as publicly available by forging

contextual information, or design trigger conditions to cause the LLM agent to output stored sensitive information under specific circumstances. For example, an attacker can insert forged instructions to trick the LLM agent into believing that the user has authorized the disclosure of its private information, resulting in privacy leakage. Zhong et al. [106] introduced a corpus poisoning attack targeting dense retrieval systems, where adversarial passages were injected into the retrieval corpus to mislead models into retrieving them for unrelated or unseen queries. Zou et al. [107] introduced PoisonedRAG, a knowledge corruption attack targeting RAG systems in LLMs. It demonstrated how attackers could inject malicious texts into a knowledge database to mislead LLM agent into generating incorrect or attacker-desired responses. In [108], the attack could be carried out by injecting false or misleading data (memory poisoning) into FINMEM's layered memory module, affecting its long-term storage and decision-making process.

6. Privacy protection in pre-training and fine-tuning

Privacy protection in pre-training and fine-tuning of LLMs is paramount in safeguarding sensitive data while ensuring model effectiveness. Incorporating techniques such as differential privacy, data cleaning, and federated learning can mitigate privacy risks.

6.1. Privacy protection in pre-training

6.1.1. Data cleaning

Data cleaning enhances data quality by rectifying errors and inconsistencies, serving as a foundational step that also plays a critical role in privacy protection by implementing anonymization, data minimization, and secure practices to safeguard sensitive information. To be more specific, we can remove or generalize personally identifiable information (PII) such as names, addresses, social security numbers, etc., to make it harder to identify individuals in the dataset (e.g., we can also mask sensitive information by replacing it with non-sensitive placeholders or pseudonyms while still preserving the structure and relationships within the dataset); we can aggregate data at a higher level to reduce the risk of re-identification. For example, instead of storing individual inference query details, and aggregate queries by day or week.

Tech Tips: When utilizing data cleaning techniques for privacy protection in LLMs, it is essential to prioritize thorough data sanitization before fine-tuning the model for specific tasks. Anonymizing or pseudonymizing sensitive information, and aggregating data to reduce granularity are key strategies to safeguard individual privacy.

OpenAI [59] underscores the thorough measures implemented to elevate the quality and security of their training data. They utilize filtering and fuzzy deduplication techniques to remove personally identifiable information from the corpora utilized for model training. This methodology not only purifies the data but also secures a heightened level of privacy protection. These measures are also employed in [109]. Anthropic [110] adopts a strategic approach in their training methodologies, focusing on the exclusive use of beneficial human feedback data to develop AI assistants. Additionally, their commitment to fostering AI behavior that aligns with constitutional and ethical standards is further highlighted in [111]. Kandpal et al. [112] demonstrated that removing duplicated sequences from training data significantly reduces the vulnerability of language models to privacy attacks, such as those allowing adversaries to recover memorized information [92]. Through empirical analysis, the authors show that duplication in training data is a key factor contributing to these privacy risks.

6.1.2. Federated learning

Federated learning revolutionizes machine learning by decentralizing the training process, enabling model training across multiple edge devices or servers while preserving data privacy. Initially, a global model is distributed to participating devices, which independently train the model using their local data. Instead of sending raw data to a central server, only model updates are transmitted, ensuring user privacy as data remains localized. These updates are aggregated at the central server to refine the global model iteratively, leading to continuous improvement without compromising privacy. Federated learning thus offers a paradigm shift, promoting collaborative machine learning in privacy-sensitive environments by leveraging distributed data processing and maintaining data locality.

Chen et al. [113] introduced a federated learning framework for LLMs that focuses on privacy without sacrificing performance, incorporating federated pre-training to securely utilize decentralized data for improved privacy, security, and model generalization. Yu et al. [114] developed Federated Foundation Models to enhance privacy in collaborative learning, focusing on the entire lifecycle of foundation models with federated learning. They tackle privacy, performance, and scalability, paving the way for future research on privacy-preserving, personalized models.

Tech Tips: In the pre-training of LLMs, federated learning offers a privacy-centric approach by eliminating the need for centralized data storage. Training occurs on local devices, with only model parameters or updates sent to a central server for aggregation. This method keeps personal data on its original device, drastically reducing data breach risks and addressing privacy and security concerns associated with centralized storage.

Finding: Federated learning is not enough

Federated learning protects data privacy across participants by decentralizing the training process, where data remains on users' devices and only model updates are shared. However, it is not entirely secure against privacy breaches; malicious servers could potentially extract private user data from shared gradients. To bolster security, federated learning often integrates additional privacy-preserving techniques such as differential privacy, secure multi-party computation, homomorphic encryption, and adversarial training. These methods collectively enhance the robustness of privacy protection in federated learning frameworks.

6.1.3. Differential privacy

Differential privacy is a technique for protecting data privacy, particularly in the fields of statistical release and data analysis. Its purpose is to allow researchers to extract useful statistical information from dataset without revealing any individual data. Differential privacy achieves this by adding a certain amount of random noise to the data, ensuring that even if attackers have complete background knowledge except for the target dataset, they cannot determine whether the dataset contains information about a specific individual. We can define differential privacy as follows:

Definition 6.1. Given two datasets D_1 and D_2 , that differ by only one element (i.e., they are “adjacent datasets”), a randomized algorithm A satisfies ϵ -differential privacy if and only if for all output sets S from the algorithms on D_1 and D_2 , the following holds:

$$\frac{\Pr[A(D_1) \in S]}{\Pr[A(D_2) \in S]} \leq e^\epsilon \quad (1)$$

where $\Pr[A(D_1) \in S]$ represents the probability that the result of running algorithm A on dataset D_1 falls within the set S . ϵ is a non-negative parameter known as the privacy budget. The smaller the ϵ , the higher the level of privacy protection, but this may reduce the utility of the data. e is the base of the natural logarithm, approximately equal to 2.71828.

Since the algorithm A is random, differential privacy can ensure that for adjacent datasets (i.e., datasets that differ by only one element), the output of an algorithm is “almost identical”. This means that it is nearly impossible to infer any specific information about an individual from the output. By adjusting the value of ϵ , a trade off can be realized between data privacy protection and data utility.

Tech Tips: Integrating differential privacy into the pre-training process of LLMs involves adding noise to the training data or model updates to safeguard individual privacy while maintaining effective model training. This can be achieved by injecting random noise into training data or perturbing gradients during backpropagation. Adaptive noise mechanisms dynamically adjust noise levels based on data sensitivity and privacy budgets. Careful management of the privacy budget ensures desired privacy levels are maintained.

Hoory et al. [115] examined the application of differential privacy to pre-trained language models. It focuses on evaluating and enhancing the performance of these models under privacy constraints. Du et al. [116] focused on providing differential privacy in forward propagation for large-scale models. It addresses the challenge of protecting data privacy while performing forward propagation in large models. Li et al. [117] argued that LLMs can be effective learners under differential privacy constraints. It explores techniques to optimize model performance while adhering to privacy standards.

6.2. Privacy protection in fine tuning

6.2.1. Federated learning

Federated learning transcends its initial application in pre-training, proving equally effective in the fine-tuning phase. This expanded application not only extends its utility but also underscores its versatility in bolstering privacy protection. By operating across data, models, and commands, federated learning presents a holistic solution, showcasing its comprehensive applicability and potential for addressing privacy concerns in diverse contexts.

Tech Tips: Similarly, in the fine-tuning phase, federated learning is employed by initially distributing the pre-trained global model to edge devices or local servers where fine-tuning tasks are performed. On each device or server, the global model is fine-tuned using locally held data pertinent to the specific task.

Xu et al. [118] and Zhang et al. [119] integrated federated learning into the fine-tuning of LLMs to significantly enhance privacy protection. Their approaches focus on keeping sensitive data on the user's device, eliminating the need for direct data transmission and sharing. Sun et al. [120] introduced FedBPT, a federated learning framework for privacy-preserving prompt tuning in language models, optimizing prompts locally and sharing only updates to minimize communication overhead and ensure data privacy. This method facilitates secure, collaborative model enhancement without exposing sensitive data. Zhao et al. [14] enhanced privacy in model fine-tuning across decentralized nodes by aggregating local updates into a central model without centralizing data, effectively keeping sensitive information local and

mitigating data breach risks while leveraging collaborative learning benefits. Fan et al. [121] presented an approach that combines federated learning with knowledge distillation and parameter-efficient fine-tuning in LLMs to ensure privacy. They also introduce secure aggregation for safely merging model updates, enabling collaborative, privacy-preserving learning across different organizations. Kuang et al. [122] introduced FederatedScope-LLM, a package that integrates a benchmarking pipeline, parameter-efficient fine-tuning algorithms, and resource-efficient operations to enhance federated LLM fine-tuning under privacy constraints. Wu et al. [123] proposed FedBiOT, a system that compresses the LLM into a lighter model comprising an emulator and an adapter, enabling efficient local updates. FedBiOT can fine-tune LLMs in a federated learning environment without requiring clients to access the entire model. Wu et al. [124] explored the integration of Reinforcement Learning with Human Feedback (RLHF) in federated learning environments to align LLMs with client preferences. The paper introduced two methods, FedBis and FedBiscuit, which train binary selectors to enhance LLM outputs based on client-provided preferences without sharing sensitive data.

Finding: Federated Learning in Pre-Training V.S. in Fine-Tuning

In federated learning, pre-training employs extensive, general datasets for foundational language comprehension through distributed learning, emphasizing data privacy. While fine-tuning focuses on specialized tasks using targeted datasets, prioritizing personalized optimization and stricter privacy on local devices. The technical needs for privacy protection distinctly vary among these stages. Most research on addressing privacy issues in LLMs through federated learning focuses on optimizing the computational and communication overhead. These studies either claim applicability to pre-training and fine-tuning phases or claim relevance to a specific phase without making targeted adjustments or designs for that stage. This highlights a gap: the need for precise, stage-specific optimization and design in federated learning for LLMs, essential for improving privacy protection's effectiveness and efficiency at different stages.

6.2.2. Differential privacy

The approaches primarily employ differential privacy techniques to handle privacy-sensitive tuning data, thereby enabling secure and private inference. These approaches focus on balancing the data utility in model tuning with the data privacy [95,115–117,125–130]. Behnia et al. [125] introduced EW-Tune, a framework for fine-tuning LLMs with differential privacy guarantees. EW-Tune employed the Edgeworth accountant method, offering finite-sample privacy guarantees suitable for the fine-tuning context. It solves the problem of how to fine-tune LLMs on private data without compromising privacy. Shi et al. [126] presented a framework for enhancing the privacy of LLMs without significantly compromising their utility. The proposed approach, Just Fine-tune Twice (JFT), focuses on selectively applying differential privacy (SDP) to only the sensitive parts of data, based on a policy function. This is achieved through a two-phase fine-tuning process: first with redacted data and then with original data using a privacy-preserving mechanism. This method is shown to be effective for transformer-based models and addresses the limitations of prior SDP applications. Wu et al. [127] designed an Adaptive Differential Privacy (ADP) framework for language model training. It estimates the privacy probability of linguistic items without resorting to the prior privacy information and

designs a novel Adam algorithm to adaptively adjust the degree of differential privacy noise, potentially improving model utility while maintaining privacy. Li et al. [95] explored a method for prompt tuning LLMs in a privacy-preserving manner. This approach seeks to leverage the power of large models while safeguarding user privacy.

6.2.3. Knowledge unlearning

Knowledge unlearning, also known as machine unlearning, is a strategy aimed at bolstering privacy within machine learning models, especially LLMs [131]. When a machine learning model is trained on data, it learns patterns and correlations present in that data. However, sometimes these patterns may inadvertently encode sensitive information about individuals. If the model retains this information, it can pose privacy risks when the model is deployed in real-world applications, especially in scenarios where the model may be exposed to sensitive data. Knowledge unlearning techniques aim to mitigate these risks by selectively forgetting or removing sensitive information from the model.

Tech Tips: In the fine-tuning stage, it functions by ensuring that the model does not hold onto or disclose sensitive details learned during its initial training phases. This process involves retraining the model to eliminate its memory of certain information, effectively reducing the risk of privacy breaches while maintaining or enhancing the model's performance.

Zhang et al. [132] analyzed the Right to be Forgotten in LLMs, identifying the unique legal and technological hurdles and proposing solutions like differential privacy and machine unlearning to balance privacy with technological progress. Chen et al. [133] introduced an efficient unlearning technique for LLMs using unlearning layers within transformers, enabling precise data removal without retraining and effectively managing sequential deletion requests with minimal performance loss. Jang et al. [134] proposed a targeted unlearning method for LMs through gradient ascent on specific sequences, offering an efficient way to erase sensitive information while preserving overall model performance. Eldan et al. [135] detailed a novel unlearning approach for LLMs by fine-tuning on datasets modified to omit targeted knowledge, employing reinforcement bootstrapping to forget information without compromising model integrity.

6.2.4. Offsite tuning

Offsite tuning, detailed by Xiao et al. [136], refines the adaptability of models to specific tasks, prioritizing data privacy through the deployment of lightweight adapters and compressed emulators for localized adjustments.

Tech Tips: This innovative method transmits only essential components to the data owner for offsite tuning, thereby avoiding the exposure of the entire model and ensuring that sensitive data remains under the data owner's control. This significantly lowers the risk of privacy breaches. The adapter, fine-tuned with local data, is updated without direct data exposure and seamlessly reintegrated into the foundation model, effectively safeguarding data privacy throughout the adaptation process.

7. Privacy protection in inference

During the inference process of LLMs, the issue of privacy leakage has garnered widespread attention. To address this issue, researchers have developed numerous strategies to ensure privacy security during the inference phase. In this section, we summarize the privacy protection approaches for the inference stage of

Table 2
Private inference approaches.

Schemes	Tools	Improved components	Matrix multiplication	Nonlinear to polynomial	Threat model	Experiments on
THE-X ^a [137]	HE	GELU, SoftMax, LayerNorm	●	○	CPA	Bert-tiny
Iron [138]	HE	GELU, SoftMax, LayerNorm	●	○	Honest-but-curious	Bert
Bumblebee [139]	HE	SoftMax, LayerNorm	●	○	Static semi-honest	Bert-base/Large, GPT2-base, LLaMA-7B, ViT-base
Zimmerman et al. ^a [140]	HE	GELU, Softmax	○	●	CPA	Bert-like
Liu et al. [141]	HE, MPC	GELU, SoftMax, LayerNorm	●	○	Semi-honest	BERT-Tiny, BERT-Medium, RoBERTa-Base
Wang et al. [142]	MPC	SoftMax, Embedded Tables	○	○	Semi-honest	XML, ViT
CipherGPT [143]	MPC	GELU	●	○	Semi-honest	GPT2
East [144]	MPC	SoftMax, LayerNorm	○	●	Semi-honest	Bert-like
Privformer [145]	MPC	Sigmoid	●	○	Honest majority	Transformer
Puma [146]	MPC	GELU, SoftMax	○	●	Semi-honest	Bert-Base/Large, GPT2-Base/Medium/Large, Roberta-Base, LLaMA-7B
Sigma [147]	FSS	GELU, SoftMax	○	●	Semi-honest static	Bert-Tiny/Base/Large, GPT2, GPT2-Neo
Majmuda et al. [128]	DP	SoftMax	○	○	Semi-honest	RoBERTa-style
Dp-forward [129]	DP	Embedding	○	○	Semi-honest	Bert
Mai et al. [130]	DP	Embedding	○	○	Attribute inference attack	Bert, GPT2, T5
Textfusion [148]	Token fusion	Tokenizer	○	○	Text reconstruction attack	Bert-Base, Bert-Large
Yuan et al. [149]	Permutation	RELU, SoftMax, LayerNorm	●	○	-	Transformer, LLaMa

CPA Chosen plaintext attacks.

^a Note that the CKKS homomorphic encryption scheme might be vulnerable to passive attacks [150].

The ● indicates that the functionality is supported.

The ○ indicates that the functionality is not supported.

The - indicates that the threat model is not mentioned.

LLMs, focusing on various approaches including encryption-based privacy protection approaches, privacy protection approaches through detection, and hardware-based approaches (see Table 2).

7.1. Cryptography-based approaches

7.1.1. Homomorphic encryption

Homomorphic encryption [151] is a cryptographic technique that allows for computations to be performed on ciphertexts, ensuring that the result, when decrypted, is identical to the result of the same operations performed on the plaintext. This encryption method is key in enabling data to be processed while maintaining its encrypted state, adding a new dimension to data privacy. Homomorphic encryption is primarily categorized into three types:

- Partial Homomorphic Encryption (PHE): Supports one type of operation (usually addition or multiplication) on ciphertexts.
- Somewhat Homomorphic Encryption (SWHE): Allows a limited number of operations on ciphertexts.
- Fully Homomorphic Encryption (FHE): The most powerful, supporting an unlimited number of both addition and multiplication operations on ciphertexts.

To better understand homomorphic encryption algorithms, we provide the following definition.

Definition 7.1. An encryption scheme is considered homomorphic over an operation \circ if it satisfies a specific mathematical property. Specifically, it supports the following equation:

$$E(m_1) \circ E(m_2) = E(m_1 \circ m_2), \quad \forall m_1, m_2 \in \mathcal{M} \quad (2)$$

Here, E represents the encryption algorithm, \mathcal{M} denotes the set of all possible messages that can be encrypted, and m_1 and m_2 are any two messages in the scheme. The operation \star can be any binary operation (e.g. addition or multiplication).

Tech Tips: Homomorphic encryption safeguards privacy during the inference stage by encrypting both the model parameters and input data. With HE, computations can be performed directly on encrypted data, allowing the model to make predictions without decrypting sensitive information. This process ensures that neither the raw data nor the model architecture is exposed in their unencrypted form, preserving privacy throughout the inference process. The decryption of the results is only done by trusted parties possessing the decryption key, maintaining the confidentiality of the information. Additionally, HE facilitates secure outsourcing of computations to untrusted servers, enabling organizations to utilize external resources without compromising data privacy.

We now introduce privacy inference approaches based on HE [137–141]. The THE-X [137] presented a novel approach for enabling privacy-preserving inference on pre-trained transformer models using homomorphic encryption, in which they utilized ReLU to replace GELU and used approximation methods for SoftMax and LayerNorm to support the fully HE operations. However, THE-X may lead to privacy leakages as it poses intermediate results to the client during the computing of ReLU. Iron [138] focused on enhancing privacy in client-server settings, where clients have private inputs and servers hold proprietary models. It introduces several new homomorphic encryption-based protocols for matrix multiplication and complex non-linear functions (like Softmax, GELU activations, and LayerNorm) which are crucial in Transformer-based models. Bumblebee [139] optimized homomorphic encryption-based protocols for large matrix multiplication and efficient, accurate protocols for non-linear activation functions in transformers, enhancing data privacy during inference. Zimmerman et al. [140] explored secure transformer models tailored for HE, which converts the operators to their polynomial equivalent. Liu et al. [141] proposed a framework to enhance the efficiency of private inference on transformer-based models. It focuses on replacing computation-intensive operators (e.g., ReLU) in transformers with privacy-computing-friendly alternatives. The framework achieves significant reductions in

private inference time and communication overhead while maintaining near-identical model accuracy.

7.1.2. Multi-party computation

Multi-Party Computation [152,153] is a cryptographic protocol that enables multiple parties (often mutually distrusting) to collaboratively perform a computation task while keeping their individual data private. This means that even though the parties are working together to compute a result, none of them can see the other parties' private data. The objective of secure multi-party Computation is to construct a secure protocol that allows multiple mistrustful participants to jointly compute a target function on their private inputs, while ensuring the accuracy of the output, and protecting and controlling their private inputs even in the presence of dishonest behavior. SMPC can be formally described as follows: Consider n parties, denoted as P_1, P_2, \dots, P_n . Each party P_i holds a private input X_i . There is a predefined function f that takes n inputs. This function is of the form $f : (X_1, X_2, \dots, X_n) \rightarrow Y$, where X_i represents the input for party P_i and Y is the output using the secret data of all parties. Then, the parties compute the result $Y = (Y_1, Y_2, \dots, Y_n)$ based on the function $f(X_1, X_2, \dots, X_n)$ such that each party learns Y (or a portion of Y relevant to them) but learns nothing about the inputs X_i of the other parties, for all $j \neq i$.

Similar to HE, MPC is another crucial method that can be used to protect model privacy [142–146]. Wang et al. [142] focused on the challenges and solutions for private inference in transformer models using MPC. While it advances the field of privacy-preserving inference, the complexity of MPC might affect practicality and efficiency. Hou et al. [143] presented a framework CipherGPT for secure GPT model inference in a two-party setting. It introduces optimized cryptographic protocols for operations like matrix multiplication and GELU activation, which are essential for GPT models. The framework focuses on preserving privacy while ensuring the efficiency of the inference process. However, the specific focus on two-party settings may limit the framework's applicability in more diverse operational environments. Ding et al. [144] proposed a communication-efficient protocol called East for activation functions like GELU and tanh, as well as optimized protocols for softmax and Layer Normalization (LN). These protocols are designed to enhance performance by reducing runtime and communication overhead, ensuring the security of the scheme. Akimoto et al. [145] presented a MPC-based approach to secure inference of Transformer models in natural language processing using ReLU functions. This method addresses the challenge of computing the Transformer's attention mechanism efficiently and securely in an MPC setting. Dong et al. [146] introduced PUMA, a framework for efficient and secure inference on Transformer models using replicated secret sharing. PUMA offers approximations for expensive non-linear functions (e.g., GeLU and softmax), which can also evaluate the large models like LLaMA-7B efficiently under MPC.

Tech Tips: MPC enables secure aggregation of model updates in federated learning setups, allowing parties to collaboratively train a shared model. MPC ensures privacy during model inference by performing computations on encrypted data, shielding sensitive information from central servers. MPC facilitates secure data labeling by allowing multiple parties to label data collaboratively without exposing raw labels, thus maintaining the confidentiality of sensitive information throughout the process.

7.1.3. Functional secret sharing

Function Secret Sharing (FSS) [154] involves dividing an original secret into multiple shares using a mathematical function (such as a polynomial), encoding the secret into each share in such a way that each is independent and insufficient to reveal the entire secret. These parts are then distributed to different participants, who can independently execute predetermined functions, such as arithmetic or logical operations, on their portion of the secret. These computations are carried out on secret shares that are in an encrypted or hidden state, preventing participants from obtaining any information about the original secret from their share alone. The results obtained by each participant are then aggregated, and when a sufficient number of shares are combined and computed, the outcome of executing the function on the entire secret is recovered. The security of this process lies in the fact that each share does not contain enough information to reveal the secret by itself; hence, even if some shares are compromised or participants are dishonest, the secret remains secure. The original secret's information is only revealed when the predetermined threshold is reached, that is, when a certain number of shares are correctly combined.

Tech Tips: In FSS, the LLM or function is partitioned into shares using cryptographic methods, with each party holding a share. During computation, parties perform operations on their shares using their private data, ensuring that individual inputs remain undisclosed. After computation, the parties collaboratively combine their shares to reconstruct the result of the function, maintaining privacy while revealing the final output.

As far as we know, there has been only one secure privacy inference approach based on Function Secret Sharing (FSS), which was proposed by Gupta et al. [147]. The approach discussed a system named SIGMA for secure inference of transformer-based models, specifically focusing on Generative Pre-trained Transformers. SIGMA is designed to be efficient in terms of latency and communication overhead while maintaining standard 2-party computation (2PC) security by leveraging FSS. It introduces new FSS-based protocols for complex machine learning functionalities like Softmax and GeLU and optimizes them for GPU acceleration. SIGMA claims significant improvements in latency over state-of-the-art systems and demonstrates scalability to large GPT models. However, the paper does not explicitly outline specific disadvantages, which typically in such systems could include complexity of implementation, computational resource requirements, or potential limitations in the types of models or data that can be securely processed.

7.1.4. Differential privacy in inference

Similarly, differential privacy can also be applied in the inference stage of LLM, providing a crucial layer of privacy protection during the generation of model predictions or outputs.

Tech Tips: In the inference stages of LLMs, DP can introduce noise to model outputs to safeguard individual data privacy while preserving prediction accuracy. Parameters are adjusted to manage the privacy budget effectively, with continuous monitoring ensuring a balance between privacy and utility over time.

Majmudar et al. [128] presented a method for ensuring differential privacy in the decoding process of LLMs. This approach aims to protect privacy during text generation. Du et al. [129] proposed a method for fine-tuning and inference in language models while maintaining differential privacy during the forward pass. It tackles the challenge of protecting privacy during

both fine-tuning and inference phases. Mai et al. [130] introduced the Split-and-Denoise method, combining local differential privacy with a denoising technique to protect privacy in large language model inference. Zhou et al. [148] introduced a method for privacy-preserving inference in pre-trained models using token fusion. The advantage is maintaining privacy during inference, but it could impact the inference accuracy or efficiency. Yuan et al. [149] detailed a three-party protocol for secure Transformer model inference, safeguarding both model parameters and user data. It applies permutation instead of complex encryption, offering strong security with practical feasibility for global matrix multiplication-based layers.

Finding: Cryptography-based Private Inference

Privacy protection techniques grounded in Homomorphic Encryption (HE), Multi-Party Computation (MPC), and Functional Secret Sharing (FSS) offer demonstrable security assurances within rigorously defined threat models, as indicated in Table 2. Nevertheless, limitations in performance and efficiency present obstacles to their near-term adoption by prominent model service providers. Even though these techniques have enhanced the efficiency of critical components, their experimental results demonstrate that deploying HE, MPC, and FSS might lead to degraded performance. Alternative approaches often rely on principles of obfuscation, yet their levels of randomness and security are weaker than cryptography-based solutions, and they typically consider specific attacks.

7.2. Detection-based approaches

In existing research on LLMs, some efforts focus on detecting privacy leaks [155–158]. These studies predominantly examine whether the content generated by LMs directly exposes data privacy or if such privacy can be inferred through contextual associations. This approach is equally applicable to LLMs, suggesting a viable pathway for assessing and mitigating privacy risks in more advanced linguistic computational models [159].

Tech Tips: Detection-based methods for protecting the privacy of LLM involve identifying and mitigating potential privacy risks in the text generated by these models which have two main strategies: (i) Direct detection methods involve directly examining the text generated by LLMs to identify privacy leaks. (ii) Contextual inference detection methods focus on identifying privacy breaches that may not be explicitly evident in the generated text but can be deduced through contextual correlations.

Finding: Detection-based Approaches

Due to the inherent complexity and variability of text data, scrutinizing the outputs of LLMs in practical applications has its limitations. Attackers can exploit these limitations by crafting impermissible outputs from seemingly permissible ones [159]. This underscores the necessity for advanced and dynamic security measures, beyond simple output filtering or static rules, to effectively counteract sophisticated manipulation techniques and ensure the integrity and safety of LLMs applications.

7.2.1. Direct detection

Kim et al. [160] developed a black-box probing method to evaluate privacy risks in LLMs by using crafted prompts to elicit Personally Identifiable Information (PII) from model outputs. This approach assesses the likelihood of LLMs inadvertently revealing PII, offering a targeted strategy for understanding privacy vulnerabilities in generated text. Phute et al. [161] unveiled a zero-shot defense strategy for LLMs aimed at curbing harmful content generation. By deploying a harm classifier from the same LLM, this method significantly reduces the efficacy of adversarial attacks, eliminating the need for fine-tuning. Chen et al. [162] developed a moving target defense system for LLMs to counter adversarial attacks, using N-Gram models and naive Bayes classification for evaluating responses and BERT for assessing question-answer coherence, effectively distinguishing between beneficial and malicious content.

7.2.2. Contextual inference detection

Mireshghallah et al. [163] introduced CONFAIDE, a benchmark that evaluates LLMs' privacy reasoning across four complexity levels, revealing notable deficiencies in models like GPT-4 and ChatGPT in terms of privacy preservation and social reasoning. Huang et al. [164] proposed a framework to assess PLMs' risk of privacy leakage, focusing on email addresses. Their approach, which analyzes memorization and association, highlights vulnerabilities in how models might unintentionally disclose or link email addresses to individuals.

7.3. Hardware-based approaches

Hardware-based approaches for protecting the privacy of LLM focus on leveraging specialized hardware features and technologies to establish secure execution environments and safeguard data during processing.

Tech Tips: Hardware-based Approaches such as Trusted Execution Environments (TEEs), hardware virtualization, secure enclaves, hardware Root of Trust (RoT), and encrypted processing, aim to ensure the confidentiality, integrity, and privacy of both the model parameters and the data being processed.

7.3.1. Data locality

PrivateLoRA [165] leveraged edge devices' storage for private data and personalized parameters, while utilizing the cloud for computational enhancement. It splits model parameters across the cloud and edge devices and transmits only unreadable activations and gradients to maintain data locality. The method integrates three sequential low-rank matrices for weight adaptation and reduces communication overhead through Low Rank Residual Transmission. It ensures data locality by keeping personalized parameters on edge devices and raw data derivatives on the cloud. The model targets query, key, and value projections in self-attention for adaptation to minimize communication overhead. PrivateLoRA is a paradigm that powers a heterogeneously distributed inference and training cycle, achieving high throughput and performance on smartphones.

7.3.2. Confidential computing with Trusted Execution Environment (TEE)

Confidential computing aims to address this gap by safeguarding data even while it is being processed. One key technology used in confidential computing is Trusted Execution Environments (TEEs). A TEE is a secure area of a computer's processor that ensures code and data loaded inside it are protected from unauthorized access or modification, even from the operating system or hypervisor. TEEs provide a secure environment where sensitive

computations can be performed, ensuring the confidentiality and integrity of the data being processed [166–173].

The NVIDIA H100 GPU, featuring support for confidential computing, enhances data privacy by establishing a secure execution environment through hardware virtualization and a TEE [174]. This environment ensures that data and code are processed securely during training or inference, preventing unauthorized access or modification by unauthorized users. By anchoring security measures in an on-die hardware root of trust (RoT), NVIDIA ensures the integrity of the GPU's boot sequence and establishes a chain of trust through cryptographic attestation. Furthermore, NVIDIA continues to enhance security and integrity by incorporating features such as encrypted firmware, firmware revocation, and fault injection countermeasures. The TEEs applied in [175] protect privacy by securely executing custodial operations, encrypting and controlling access to data, and facilitating encrypted transmission of user queries and prompts. Huang et al. [176] introduced a method deploying TEEs on both client and server sides, implementing secure communication and split fine-tuning of a language model to maintain accuracy.

8. Privacy protection in LLM agents

Privacy protection is the core challenge of LLM agents in data processing, encompassing the stages of data input, processing, and output. This paper outlines key privacy protection techniques, including sensitive information filtering and homomorphic encryption during the input stage, session isolation and multi-layered security strategies during the processing stage, and response filtering during the output stage.

8.1. Privacy protection in agent input

8.1.1. Sensitive information filtering

In the agent input stage, the filtering mechanism is primarily used to screen and validate user input to ensure compliance with privacy regulations. For example, privacy agents analyze the context of the input data, including the sender, receiver, and content, to determine whether the data meets predefined privacy rules [177]. If the input contains sensitive or non-compliant content, the filtering mechanism masks or blocks it, thereby reducing the risk of privacy breaches.

8.1.2. Homomorphic encryption

Homomorphic encryption has shown to be an effective approach for maintaining data confidentiality in LLM agents [178]. Zhang et al. [178] proposed PrivacyAsst, a privacy protection framework designed for tool-using LLM agents, utilizing homomorphic encryption and data obfuscation techniques to safeguard users' private data. PrivacyAsst ensured that user data was not accessed in plaintext during processing, effectively reducing the risk of privacy leakage.

Tech Tips: Use homomorphic encryption to protect data confidentiality in LLM agents. This approach ensures that sensitive data remains encrypted during the agent input, significantly reducing privacy leakage risks. In LLM agents, users encrypt their private inputs before sending them to the LLM. The agent then forwards the encrypted data to the tool for processing, and another user can decrypt the ciphertext to obtain the computed results, thereby ensuring the protection of private data.

8.2. Privacy protection in agent data processing

8.2.1. Session isolation

Session isolation effectively protects data privacy by strictly separating the context and data streams of different users or tasks, preventing information from being leaked or misused among sessions. He et al. [179] explored the core security challenges in AI agent systems, including issues of confidentiality, integrity, and availability, especially emphasizing the need to protect user privacy and data security during tool interactions. The authors proposed multi-layered methods such as distributed session management, encryption protection, and sandbox isolation to enhance AI agent security, preventing information leakage and model pollution.

Tech Tips: In LLM agents, session isolation achieves data privacy protection through distributed session management, context clearance, sandboxing, and access control, ensuring that the data in each session remains confined to its scope and that sensitive information is cleared after the session ends. This mechanism is widely applied in scenarios such as multi-user collaboration, personalized assistants, healthcare, and finance, providing users with highly secure privacy protection.

8.2.2. Multi-layered security strategies in planning

To address the need for robust security in task execution, multi-layered security strategies in task planning provide a comprehensive approach to mitigating risks among different stages of LLM agent Plan stage. Hua et al. [180] proposed the TrustAgent framework, which ensured the security and reliability of LLM agents in performing complex tasks through an agent constitution and multi-layered security strategies. The framework effectively reduced safety risks in task execution by introducing pre-planning, in-planning, and post-planning strategies.

8.3. Privacy protection in agent output

8.3.1. Response filtering

To safeguard privacy and maintain data integrity in LLM agents, robust auditing and filtering mechanisms are essential for managing sensitive information in line with contextual privacy norms [181,182]. Song et al. [181] presented Audit-LLM, which combines log analysis and response filtering to ensure that the content generated by the model meets security standards. Through a dual-agent forensics mechanism and tool collaboration, it can identify harmful content in complex logs, while the response filtering mechanism audits and modifies responses that may contain sensitive information or harmful content before outputting the content. Zeng et al. [182] proposed AutoDefense, a multi-agent framework designed to defend against jailbreaking attacks through a response filtering mechanism, ensuring that content generated by LLMs met safety standards.

Tech Tips: Implement robust auditing and filtering to maintain data integrity and manage sensitive information according to privacy norms. Techniques such as context isolation, multi-agent collaboration, and personalized privacy settings help protect against privacy leakage and reduce risks associated with LLM agent automation.

9. Challenges and future directions

In this section, we systematically examine the multifaceted challenges and potential advancements in the privacy and security of LLMs. By following the complete lifecycle sequence, we will

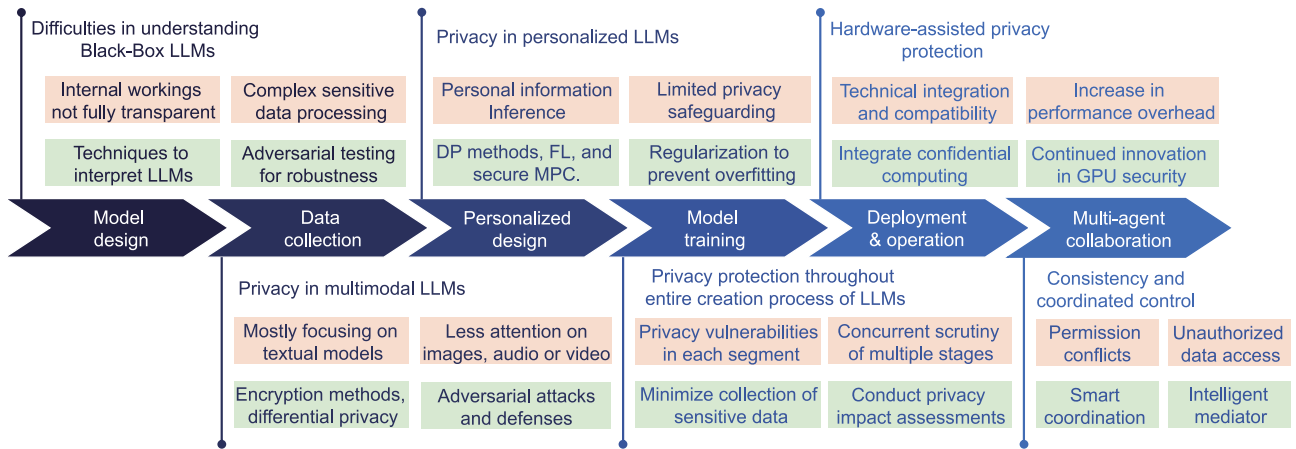


Fig. 6. The challenges and future research directions faced by large language models throughout their entire lifecycle, with the light orange box representing the challenges and the light green box representing the future directions.

not only highlight the existing challenges but also outline future directions, providing readers with a comprehensive and insightful framework. For a detailed breakdown of each aspect, please refer to the diagram presented in Fig. 6.

9.1. Difficulties in understanding black-box LLMs

Pre-trained LLMs are often treated as black box models [183, 184], meaning that their internal workings and decision-making processes are not fully transparent or interpretable. This opacity makes it challenging to analyze and understand how these models handle sensitive information and whether they inadvertently leak privacy. In addition, LLMs are trained on vast amounts of diverse data, which may include sensitive or personally identifiable information. Understanding how these models process and retain such data without compromising privacy is inherently complex, especially given the intricate relationships between input data and model outputs. Language is dynamic and context-dependent, leading to challenges in predicting how LLMs will behave in various real-world scenarios. Privacy risks may vary depending on the context in which the model is deployed, making it difficult to generalize findings across different applications or domains.

Developing techniques to interpret and explain the decisions of pre-trained LLMs can shed light on their privacy implications. This may involve analyzing model activations, attention mechanisms, or other internal representations to identify potential privacy vulnerabilities. Conducting adversarial testing to evaluate the robustness of pre-trained LLMs against privacy attacks. For example, adversarial examples can be generated to probe the model's behavior and identify weaknesses that may lead to privacy breaches [185]. Besides, we can focus on developing fine-tuning techniques that explicitly consider privacy concerns, such as differential privacy-aware optimization or adversarial training with privacy objectives. These techniques aim to mitigate privacy risks during the fine-tuning process.

9.2. Privacy in multimodal LLMs

The majority of research on LLMs has focused on purely textual models such as GPT and BERT. As a result, there may be a tendency for researchers to prioritize investigating the privacy implications of these models, leaving less attention on Multimodal LLMs. Multimodal LLMs, which integrate both textual and visual information, are a relatively recent development compared to their purely textual counterparts [186,187]. As such, there has been less time for researchers to explore and investigate

their privacy implications thoroughly. Multimodal LLMs process a more diverse range of data types, including text, images, and possibly other modalities such as audio or video. Analyzing the privacy implications of such complex and heterogeneous data poses additional challenges compared to purely textual data, which may deter some researchers from delving into this area.

Redefining privacy in Multimodal LLMs is necessary to address the increased data complexity, unique privacy risks, inter-modal interactions, user expectations, and regulatory considerations associated with multimodal data processing. Developing techniques to fuse different modalities while preserving user privacy is an important research direction. This could involve exploring encryption methods, differential privacy techniques, or novel privacy-preserving machine learning algorithms tailored to multimodal data. Conducting adversarial analyses to identify potential vulnerabilities and privacy risks in Multimodal LLMs. This could involve exploring adversarial attacks and defenses specific to multimodal data, such as perturbing images or textual inputs to compromise privacy.

9.3. Privacy in personalized LLMs

Personalized LLMs may store and process sensitive user data, such as personal conversations, search queries, or browsing history. If not adequately protected, this data could be vulnerable to unauthorized access or misuse, leading to privacy breaches and potential harm to individuals. Personalized LLMs can infer personal information about users based on their interactions with the model. This includes sensitive attributes such as health status, political views, financial situation, or intimate preferences. Such inferences could be unintentionally revealed through model responses or recommendations, compromising user privacy. Numerous small-scale enterprises offer users specialized large-scale model services tailored to vertical domains, encompassing sectors such as judiciary, education, and finance. These expansive models entail a greater incorporation of domain-specific personal data. However, owing to the comparatively limited privacy safeguarding capabilities inherent in small-scale enterprises, the susceptibility to user privacy breaches is heightened, potentially precipitating irreversible ramifications.

To safeguard personalized fine-tuning of LLMs from privacy leakage, we need to explore architectures specifically designed with privacy [188]. In addition, we can develop a combination of techniques. This includes implementing differential privacy methods to add noise during training, utilizing federated learning to train models locally on user devices, employing secure

multi-party computation to jointly train models without sharing private data directly, and introducing data perturbation to prevent memorization of sensitive information. We can also apply regularization methods to prevent overfitting, and explore privacy-preserving architectures designed specifically for protecting sensitive data during fine-tuning.

9.4. Privacy protection throughout the entire creation process of LLMs

Given the intricate complexity involved in training LLMs, privacy protection research tends to dissect various phases of LLM development and deployment, including pre-training, prompt tuning, and inference. Nevertheless, each segment within the LLM lifecycle harbors its own set of privacy vulnerabilities, and these stages do not operate in isolation [189]. For instance, privacy breaches detected during the inference phase might originate from potential backdoors introduced during pre-training. Thus, safeguarding privacy comprehensively across large models demands concurrent scrutiny of multiple stages, a task that also introduces complexities and challenges into privacy protection efforts.

Protecting the privacy of LLMs throughout their creation process is paramount and requires a multifaceted approach. Firstly, during data collection, minimizing the collection of sensitive information and obtaining informed consent from users are critical steps. Data should be anonymized or pseudonymized to mitigate re-identification risks. Secondly, in data preprocessing and model training, techniques such as federated learning, secure multiparty computation, and differential privacy can be employed to train LLMs on decentralized data sources while preserving individual privacy. Additionally, conducting privacy impact assessments and adversarial testing during model evaluation ensures potential privacy risks are identified and addressed before deployment. In the deployment phase, privacy-preserving APIs and access controls can limit access to LLMs, while transparency and accountability measures foster trust with users by providing insight into data handling practices. Ongoing monitoring and maintenance, including continuous monitoring for privacy breaches and regular privacy audits, are essential to ensure compliance with privacy regulations and the effectiveness of privacy safeguards. By implementing these measures comprehensively throughout the LLM creation process, developers can mitigate privacy risks and build trust with users, thereby leveraging the capabilities of LLMs while safeguarding individual privacy.

9.5. Hardware-assisted privacy protection

The main challenges of integrating confidential computing into LLMs not only include issues of technical integration and compatibility but also an increase in performance overhead, as encryption and decryption operations may consume additional computational resources, thereby affecting the model's response time and processing capacity. Additionally, the cost of deploying and maintaining a confidential computing system is a significant factor, especially when frequent updates and upgrades are needed to address new security threats. These challenges require NVIDIA to continuously optimize its technical solutions to balance security and performance.

NVIDIA Confidential Computing provides a comprehensive suite of privacy-enhancing features and technologies that safeguard LLM data and operations against unauthorized access, manipulation, and breaches, thereby ensuring the confidentiality and integrity of sensitive information throughout the LLM lifecycle. In the future, we can integrate confidential computing capabilities into LLM workflows, ensuring comprehensive privacy

protection across the entire lifecycle, while continued innovation in GPU security features, such as encrypted firmware and fault injection countermeasures, reinforces the company's commitment to advancing data privacy safeguards for sensitive workloads.

9.6. Consistency in multi-agent collaboration

Multi-agent collaboration requires consistency across multi-task, cross-system environments. However, due to variations in behavior and decision logic among agents, issues such as low collaboration efficiency, insufficient information sharing, or excessive redundancy may arise. Additionally, preventing permission conflicts, unauthorized data access, and errors among agents is a key challenge for achieving efficient collaboration.

Developing "smart coordination protocols" to enable multiple agents to autonomously allocate tasks and share information in dynamic environments; researching decentralized collaboration mechanisms based on blockchain to ensure data transparency and consistency within multi-agent systems. Introducing an "intelligent mediator" agent that acts as a coordinator among collaborative agents, providing real-time task monitoring and allocation to ensure smooth and reliable information flow.

10. Conclusion

In this paper, we thoroughly investigate the data privacy concerns associated with LLMs and LLM agents, focusing on privacy leakage, privacy attacks, and the pivotal technologies for privacy protection during various stages of LLM privacy inference, including federated learning, differential privacy, knowledge unlearning, and hardware-assisted privacy protection. By conducting a detailed analysis of the strengths and weaknesses of existing approaches, this study highlights the challenges and limitations of LLMs and LLM agents. Then, we propose the directions for future work. This research is of significant importance for deepening our understanding of data privacy issues, promoting further exploration and improvement in LLMs and LLM agents.

CRedit authorship contribution statement

Biwei Yan: Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis, Conceptualization. **Kun Li:** Methodology. **Minghui Xu:** Methodology. **Yueyan Dong:** Writing – original draft. **Yue Zhang:** Writing – review & editing, Methodology, Conceptualization. **Zhaochun Ren:** Data curation, Conceptualization. **Xiuzhen Cheng:** Writing – review & editing, Methodology, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62402288 and 62302063) and the China Postdoctoral Science Foundation, China (2024M751811).

References

- [1] M. Gao, X. Hu, J. Ruan, X. Pu, X. Wan, LLM-based NLG evaluation: Current status and challenges, 2024, arXiv preprint [arXiv:2402.01383](https://arxiv.org/abs/2402.01383).
- [2] Y. Xie, C. Yu, T. Zhu, J. Bai, Z. Gong, H. Soh, Translating natural language to planning goals with large-language models, 2023, arXiv preprint [arXiv:2302.05128](https://arxiv.org/abs/2302.05128).
- [3] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large language models: A survey, 2024, arXiv preprint [arXiv:2402.06196](https://arxiv.org/abs/2402.06196).
- [4] C.H. Song, J. Wu, C. Washington, B.M. Sadler, W.-L. Chao, Y. Su, Llm-planner: Few-shot grounded planning for embodied agents with large language models, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023*, pp. 2998–3009.
- [5] Z. Xu, K. Wu, J. Wen, J. Li, N. Liu, Z. Che, J. Tang, A survey on robotics with foundation models: toward embodied AI, 2024, arXiv preprint [arXiv:2402.02385](https://arxiv.org/abs/2402.02385).
- [6] J. Duan, S. Yu, H.L. Tan, H. Zhu, C. Tan, A survey of embodied ai: From simulators to research tasks, *IEEE Trans. Emerg. Top. Comput. Intell.* 6 (2) (2022) 230–244.
- [7] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P.S. Yu, L. Sun, A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to ChatGPT, 2023, arXiv preprint [arXiv:2303.04226](https://arxiv.org/abs/2303.04226).
- [8] J. Wu, W. Gan, Z. Chen, S. Wan, H. Lin, Ai-generated content (aigc): A survey, 2023, arXiv preprint [arXiv:2304.06632](https://arxiv.org/abs/2304.06632).
- [9] Y. Cheng, M. Xu, Y. Zhang, K. Li, R. Wang, L. Yang, AutoIoT: Automated IoT platform using large language models, 2024, arXiv preprint [arXiv:2411.10665](https://arxiv.org/abs/2411.10665).
- [10] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, Y. Zhang, A survey on large language model (llm) security and privacy: The good, the bad, and the ugly, *High-Confid. Comput.* (2024) 100211.
- [11] N. Subramani, S. Luccioni, J. Dodge, M. Mitchell, Detecting personal information in training corpora: an analysis, in: *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing, TrustNLP 2023, 2023*, pp. 208–220.
- [12] R. Staab, M. Vero, M. Balunović, M. Vechev, Beyond memorization: Violating privacy via inference with large language models, 2023, arXiv preprint [arXiv:2310.07298](https://arxiv.org/abs/2310.07298).
- [13] Z. Hu, P. Yang, F. Liu, Y. Meng, X. Liu, Prompting large language models with knowledge-injection for knowledge-based visual question answering, *Big Data Min. Anal.* 7 (3) (2024) 843–857.
- [14] J. Zhao, Privacy-preserving fine-tuning of artificial intelligence (AI) foundation models with federated learning, differential privacy, offsite tuning, and parameter-efficient fine-tuning (PEFT), 2023, *Authorea Preprint*.
- [15] L. Luo, J. Ning, Y. Zhao, Z. Wang, Z. Ding, P. Chen, W. Fu, Q. Han, G. Xu, Y. Qiu, et al., Taiyi: A bilingual fine-tuned large language model for diverse biomedical tasks, 2023, arXiv preprint [arXiv:2311.11608](https://arxiv.org/abs/2311.11608).
- [16] M. Nasr, N. Carlini, J. Hayase, M. Jagielski, A.F. Cooper, D. Ippolito, C.A. Choquette-Choo, E. Wallace, F. Tramèr, K. Lee, Scalable extraction of training data from (production) language models, 2023, arXiv preprint [arXiv:2311.17035](https://arxiv.org/abs/2311.17035).
- [17] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, *OpenAI Blog* 1 (8) (2019) 9.
- [18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805).
- [19] Y. Li, H. Wen, W. Wang, X. Li, Y. Yuan, G. Liu, J. Liu, W. Xu, X. Wang, Y. Sun, et al., Personal LLM agents: Insights and survey about the capability, efficiency and security, 2024, arXiv preprint [arXiv:2401.05459](https://arxiv.org/abs/2401.05459).
- [20] Y. Chang, X. Wang, J. Wang, Y. Wu, K. Zhu, H. Chen, L. Yang, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, 2023, arXiv preprint [arXiv:2307.03109](https://arxiv.org/abs/2307.03109).
- [21] W.X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, 2023, arXiv preprint [arXiv:2303.18223](https://arxiv.org/abs/2303.18223).
- [22] J. Wu, S. Yang, R. Zhan, Y. Yuan, D.F. Wong, L.S. Chao, A survey on LLM-generated text detection: Necessity, methods, and future directions, 2023, arXiv preprint [arXiv:2310.14724](https://arxiv.org/abs/2310.14724).
- [23] S.R. Bowman, Eight things to know about large language models, 2023, arXiv preprint [arXiv:2304.00612](https://arxiv.org/abs/2304.00612).
- [24] B. Gulzar, S.A. Sofi, S. Sholla, Exploring personalized internet of things (PIoT), social connectivity, and artificial social intelligence (ASI): A survey, *High-Confid. Comput.* (2024) 100242.
- [25] Z. Liu, Y. Bao, S. Zeng, R. Qian, M. Deng, A. Gu, J. Li, W. Wang, W. Cai, W. Li, et al., Large language models in psychiatry: Current applications, limitations, and future scope, *Big Data Min. Anal.* 7 (4) (2024) 1148–1168.
- [26] J. Xie, Y. Zhang, H. Kou, X. Zhao, Z. Feng, L. Song, W. Zhong, A survey of the application of neural networks to event extraction, *Tsinghua Sci. Technol.* 30 (2) (2025) 748–768, <http://dx.doi.org/10.26599/TST.2023.9010139>.
- [27] X. Liu, Y. He, W. Tai, X. Xu, F. Zhou, G. Luo, Exploring the chameleon effect of contextual dynamics in temporal knowledge graph for event prediction, *Tsinghua Sci. Technol.* 30 (1) (2025) 433–455, <http://dx.doi.org/10.26599/TST.2024.9010067>.
- [28] H. Naveed, A.U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Barnes, A. Mian, A comprehensive overview of large language models, 2023, arXiv preprint [arXiv:2307.06435](https://arxiv.org/abs/2307.06435).
- [29] M.U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M.B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili, et al., A survey on large language models: Applications, challenges, limitations, and practical usage, 2023, *Authorea Preprints*.
- [30] Z. Guo, R. Jin, C. Liu, Y. Huang, D. Shi, L. Yu, Y. Liu, J. Li, B. Xiong, D. Xiong, et al., Evaluating large language models: A comprehensive survey, 2023, arXiv preprint [arXiv:2310.19736](https://arxiv.org/abs/2310.19736).
- [31] Y. Liu, Y. Yao, J.-F. Ton, X. Zhang, R.G.H. Cheng, Y. Klochkov, M.F. Taufiq, H. Li, Trustworthy LLMs: a survey and guideline for evaluating large language models' alignment, 2023, arXiv preprint [arXiv:2308.05374](https://arxiv.org/abs/2308.05374).
- [32] Z. Yuan, Y. Shang, Y. Zhou, Z. Dong, C. Xue, B. Wu, Z. Li, Q. Gu, Y.J. Lee, Y. Yan, et al., Llm inference unveiled: Survey and roofline model insights, 2024, arXiv preprint [arXiv:2402.16363](https://arxiv.org/abs/2402.16363).
- [33] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N.V. Chawla, O. Wiest, X. Zhang, Large language model based multi-agents: A survey of progress and challenges, 2024, arXiv preprint [arXiv:2402.01680](https://arxiv.org/abs/2402.01680).
- [34] P. Feng, Y. He, G. Huang, Y. Lin, H. Zhang, Y. Zhang, H. Li, AGILE: A novel framework of LLM agents, 2024, arXiv preprint [arXiv:2405.14751](https://arxiv.org/abs/2405.14751).
- [35] H. Liu, Cooperative multi-agent game based on reinforcement learning, *High-Confid. Comput.* 4 (1) (2024) 100205.
- [36] Z. Zhang, X. Bo, C. Ma, R. Li, X. Chen, Q. Dai, J. Zhu, Z. Dong, J.R. Wen, A survey on the memory mechanism of large language model based agents, 2024, arXiv preprint [arXiv:2404.13501](https://arxiv.org/abs/2404.13501).
- [37] Y. Cheng, C. Zhang, Z. Zhang, X. Meng, S. Hong, W. Li, Z. Wang, Z. Wang, F. Yin, J. Zhao, et al., Exploring large language model based intelligent agents: Definitions, methods, and prospects, 2024, arXiv preprint [arXiv:2401.03428](https://arxiv.org/abs/2401.03428).
- [38] H. Li, Y. Chen, J. Luo, Y. Kang, X. Zhang, Q. Hu, C. Chan, Y. Song, Privacy in large language models: Attacks, defenses and future directions, 2023, arXiv preprint [arXiv:2310.10383](https://arxiv.org/abs/2310.10383).
- [39] S. Neel, P. Chang, Privacy issues in large language models: A survey, 2023, arXiv preprint [arXiv:2312.06717](https://arxiv.org/abs/2312.06717).
- [40] J. Marshall, What effects do large language models have on cybersecurity, 2023.
- [41] M. Al-Hawawreh, A. Aljuhani, Y. Jararweh, Chatgpt for cybersecurity: practical applications, challenges, and future directions, *Clust. Comput.* 26 (6) (2023) 3421–3436.
- [42] A. Qammar, H. Wang, J. Ding, A. Naouri, M. Daneshmand, H. Ning, Chatbots to ChatGPT in a cybersecurity space: Evolution, vulnerabilities, attacks, challenges, and future recommendations, 2023, arXiv preprint [arXiv:2306.09255](https://arxiv.org/abs/2306.09255).
- [43] L. Schwinn, D. Dobre, S. Günemann, G. Gidel, Adversarial attacks and defenses in large language models: Old and new threats, 2023, arXiv preprint [arXiv:2310.19737](https://arxiv.org/abs/2310.19737).
- [44] E. Derner, B. Batistič, Beyond the safeguards: Exploring the security risks of ChatGPT, 2023, arXiv preprint [arXiv:2305.08005](https://arxiv.org/abs/2305.08005).
- [45] E. Shayegani, M.A.A. Mamun, Y. Fu, P. Zaree, Y. Dong, N. Abu-Ghazaleh, Survey of vulnerabilities in large language models revealed by adversarial attacks, 2023, arXiv preprint [arXiv:2310.10844](https://arxiv.org/abs/2310.10844).
- [46] S. Wang, T. Zhu, B. Liu, D. Ming, X. Guo, D. Ye, W. Zhou, Unique security and privacy threats of large language model: A comprehensive survey, 2024, arXiv preprint [arXiv:2406.07973](https://arxiv.org/abs/2406.07973).
- [47] K. Li, S. Zhuang, Y. Zhang, M. Xu, R. Wang, K. Xu, X. Fu, X. Cheng, I'm Spartacus, no, I'm Spartacus: Measuring and understanding LLM identity confusion, 2024, arXiv preprint [arXiv:2411.10683](https://arxiv.org/abs/2411.10683).
- [48] H. Kibriya, W.Z. Khan, A. Siddiq, M.K. Khan, Privacy issues in large language models: A survey, *Comput. Electr. Eng.* 120 (2024) 109698.
- [49] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, et al., A survey on large language model based autonomous agents, *Front. Comput. Sci.* 18 (6) (2024) 186345.
- [50] B. Yan, K. Li, M. Xu, Y. Dong, Y. Zhang, Z. Ren, X. Cheng, On protecting the data privacy of large language models (llms): A survey, 2024, arXiv preprint [arXiv:2403.05156](https://arxiv.org/abs/2403.05156).
- [51] F. He, T. Zhu, D. Ye, B. Liu, W. Zhou, P.S. Yu, The emerged security and privacy of llm agent: A survey with case studies, 2024, arXiv preprint [arXiv:2407.19354](https://arxiv.org/abs/2407.19354).
- [52] Y. Wang, Y. Pan, Q. Zhao, Y. Deng, Z. Su, L. Du, T.H. Luan, Large model agents: State-of-the-art, cooperation paradigms, security and privacy, and future trends, 2024, arXiv preprint [arXiv:2409.14457](https://arxiv.org/abs/2409.14457).
- [53] Z. Deng, Y. Guo, C. Han, W. Ma, J. Xiong, S. Wen, Y. Xiang, AI agents under threat: A survey of key security challenges and future pathways, 2024, arXiv preprint [arXiv:2406.02630](https://arxiv.org/abs/2406.02630).
- [54] Z. Zhang, B. Guo, T. Li, Can humans oversee agents to prevent privacy leakage? A study on privacy awareness, preferences, and trust in language model agents, 2024, arXiv preprint [arXiv:2411.01344](https://arxiv.org/abs/2411.01344).

- [55] Y. Cai, S. Mao, W. Wu, Z. Wang, Y. Liang, T. Ge, C. Wu, W. You, T. Song, Y. Xia, et al., Low-code LLM: Visual programming over LLMs, 2023, arXiv preprint [arXiv:2304.08103](https://arxiv.org/abs/2304.08103).
- [56] M. Karpinska, M. Iyyer, Large language models effectively leverage document-level context for literary translation, but critical errors persist, 2023, arXiv preprint [arXiv:2304.03245](https://arxiv.org/abs/2304.03245).
- [57] A.J. Thirunavukarasu, D.S.J. Ting, K. Elangovan, L. Gutierrez, T.F. Tan, D.S.W. Ting, Large language models in medicine, *Nature Med.* 29 (8) (2023) 1930–1940.
- [58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, *Adv. Neural Inf. Process. Syst.* 30 (2017).
- [59] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020, [arXiv:2005.14165](https://arxiv.org/abs/2005.14165).
- [60] S.Y. Gadre, G. Ilharco, A. Fang, J. Hayase, G. Smyrnis, T. Nguyen, R. Marten, M. Wortman, D. Ghosh, J. Zhang, et al., Datacomp: In search of the next generation of multimodal datasets, *Adv. Neural Inf. Process. Syst.* 36 (2024).
- [61] N. Kshetri, Cybercrime and privacy threats of large language models, *IT Prof.* 25 (3) (2023) 9–13.
- [62] J. Zamfirescu-Pereira, R.Y. Wong, B. Hartmann, Q. Yang, Why johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts, in: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–21.
- [63] U. Iqbal, T. Kohno, F. Roesner, LLM platform security: Applying a systematic evaluation framework to openai's ChatGPT plugins, 2023, arXiv preprint [arXiv:2309.10254](https://arxiv.org/abs/2309.10254).
- [64] Z. Zhang, M. Jia, H. Lee, B. Yao, S. Das, A. Lerner, T. Li, "It's a fair game", or is it? Examining how users navigate disclosure risks and benefits when using LLM-based conversational agents, 2024.
- [65] E. Bagdasarayan, R. Yi, S. Ghalebikesabi, P. Kairouz, M. Gruteser, S. Oh, B. Balle, D. Ramage, Air gap: Protecting privacy-conscious conversational agents, 2024, arXiv preprint [arXiv:2405.05175](https://arxiv.org/abs/2405.05175).
- [66] X. Li, F. Huang, J. Lv, Z. Xiao, G. Li, Y. Yue, Be more real: Travel diary generation using LLM agents and individual profiles, 2024, arXiv preprint [arXiv:2407.18932](https://arxiv.org/abs/2407.18932).
- [67] H. Lyu, S. Jiang, H. Zeng, Y. Xia, J. Luo, Llm-rec: Personalized recommendation via prompting large language models, 2023, arXiv preprint [arXiv:2307.15780](https://arxiv.org/abs/2307.15780).
- [68] J. Harte, W. Zörgdrager, P. Louridas, A. Katsifodimos, D. Jannach, M. Fragkoulis, Leveraging large language models for sequential recommendation, in: *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 1096–1102.
- [69] G. Gao, A. Taymanov, E. Salinas, P. Mineiro, D. Misra, Aligning llm agents by learning latent preference from user edits, 2024, arXiv preprint [arXiv:2404.15269](https://arxiv.org/abs/2404.15269).
- [70] L. Li, D. Song, X. Li, J. Zeng, R. Ma, X. Qiu, Backdoor attacks on pre-trained models by layerwise weight poisoning, 2021, arXiv preprint [arXiv:2108.13888](https://arxiv.org/abs/2108.13888).
- [71] W. Yang, L. Li, Z. Zhang, X. Ren, X. Sun, B. He, Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in nlp models, 2021, arXiv preprint [arXiv:2103.15543](https://arxiv.org/abs/2103.15543).
- [72] H. Yao, J. Lou, Z. Qin, Poisonprompt: Backdoor attack on prompt-based large language models, 2023, arXiv preprint [arXiv:2310.12439](https://arxiv.org/abs/2310.12439).
- [73] H. Huang, Z. Zhao, M. Backes, Y. Shen, Y. Zhang, Composite backdoor attacks against large language models, 2023, arXiv preprint [arXiv:2310.07676](https://arxiv.org/abs/2310.07676).
- [74] A. Wan, E. Wallace, S. Shen, D. Klein, Poisoning language models during instruction tuning, 2023, arXiv preprint [arXiv:2305.00944](https://arxiv.org/abs/2305.00944).
- [75] J. Xu, M.D. Ma, F. Wang, C. Xiao, M. Chen, Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models, 2023, arXiv preprint [arXiv:2305.14710](https://arxiv.org/abs/2305.14710).
- [76] J. Yan, V. Yadav, S. Li, L. Chen, Z. Tang, H. Wang, V. Srinivasan, X. Ren, H. Jin, Backdooring instruction-tuned large language models with virtual prompt injection, in: *NeurIPS 2023 Workshop on Backdoors in Deep Learning-the Good, the Bad, and the Ugly*, 2023.
- [77] T. Dong, M. Xue, G. Chen, R. Holland, S. Li, Y. Meng, Z. Liu, H. Zhu, The philosopher's stone: Trojaning plugins of large language models, 2023, arXiv preprint [arXiv:2312.00374](https://arxiv.org/abs/2312.00374).
- [78] Y. Wang, D. Xue, S. Zhang, S. Qian, BadAgent: Inserting and activating backdoor attacks in LLM agents, 2024, arXiv preprint [arXiv:2406.03007](https://arxiv.org/abs/2406.03007).
- [79] W. Yang, X. Bi, Y. Lin, S. Chen, J. Zhou, X. Sun, Watch out for your agents! Investigating backdoor threats to llm-based agents, 2024, arXiv preprint [arXiv:2402.11208](https://arxiv.org/abs/2402.11208).
- [80] E. Hubinger, C. Denison, J. Mu, M. Lambert, M. Tong, M. MacDiarmid, T. Lanham, D.M. Ziegler, T. Maxwell, N. Cheng, et al., Sleeper agents: Training deceptive llms that persist through safety training, 2024, arXiv preprint [arXiv:2401.05566](https://arxiv.org/abs/2401.05566).
- [81] H. Zhang, J. Huang, K. Mei, Y. Yao, Z. Wang, C. Zhan, H. Wang, Y. Zhang, Agent security bench (asb): Formalizing and benchmarking attacks and defenses in llm-based agents, 2024, arXiv preprint [arXiv:2410.02644](https://arxiv.org/abs/2410.02644).
- [82] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: *2017 IEEE Symposium on Security and Privacy, SP, IEEE*, 2017, pp. 3–18.
- [83] H. Huang, W. Luo, G. Zeng, J. Weng, Y. Zhang, A. Yang, Damia: leveraging domain adaptation as a defense against membership inference attacks, *IEEE Trans. Dependable Secur. Comput.* 19 (5) (2021) 3183–3199.
- [84] F. Mireshghallah, K. Goyal, A. Uniyal, T. Berg-Kirkpatrick, R. Shokri, Quantifying privacy risks of masked language models using membership inference attacks, 2022, arXiv preprint [arXiv:2203.03929](https://arxiv.org/abs/2203.03929).
- [85] J. Mattern, F. Mireshghallah, Z. Jin, B. Schölkopf, M. Sachan, T. Berg-Kirkpatrick, Membership inference attacks against language models via neighbourhood comparison, 2023, arXiv preprint [arXiv:2305.18462](https://arxiv.org/abs/2305.18462).
- [86] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, L. Zettlemoyer, Detecting pretraining data from large language models, 2023, arXiv preprint [arXiv:2310.16789](https://arxiv.org/abs/2310.16789).
- [87] M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, H. Hajishirzi, Do membership inference attacks work on large language models? 2024, arXiv preprint [arXiv:2402.07841](https://arxiv.org/abs/2402.07841).
- [88] F. Mireshghallah, A. Uniyal, T. Wang, D.K. Evans, T. Berg-Kirkpatrick, An empirical analysis of memorization in fine-tuned autoregressive language models, in: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 1816–1826.
- [89] A. Jagannatha, B.P.S. Rawat, H. Yu, Membership inference attack susceptibility of clinical language models, 2021, arXiv preprint [arXiv:2104.08305](https://arxiv.org/abs/2104.08305).
- [90] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, T. Jiang, Practical membership inference attacks against fine-tuned large language models via self-prompt calibration, 2023, arXiv preprint [arXiv:2311.06062](https://arxiv.org/abs/2311.06062).
- [91] C. Song, A. Raghunathan, Information leakage in embedding models, in: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 377–390.
- [92] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al., Extracting training data from large language models, in: *30th USENIX Security Symposium, USENIX Security 21*, 2021, pp. 2633–2650.
- [93] E. Lehman, S. Jain, K. Pichotta, Y. Goldberg, B.C. Wallace, Does BERT pretrained on clinical notes reveal sensitive data? 2021, arXiv preprint [arXiv:2104.07762](https://arxiv.org/abs/2104.07762).
- [94] R. Zhang, S. Hidano, F. Koushanfar, Text revealer: Private text reconstruction via model inversion attacks against transformers, 2022, arXiv preprint [arXiv:2209.10505](https://arxiv.org/abs/2209.10505).
- [95] Y. Li, Z. Tan, Y. Liu, Privacy-preserving prompt tuning for large language model services, 2023, arXiv preprint [arXiv:2305.06212](https://arxiv.org/abs/2305.06212).
- [96] X. Pan, M. Zhang, S. Ji, M. Yang, Privacy risks of general-purpose language models, in: *2020 IEEE Symposium on Security and Privacy, SP, IEEE*, 2020, pp. 1314–1331.
- [97] K. Krishna, G.S. Tomar, A.P. Parikh, N. Papernot, M. Iyyer, Thieves on sesame street! model extraction of bert-based apis, 2019, arXiv preprint [arXiv:1910.12366](https://arxiv.org/abs/1910.12366).
- [98] J.B. Truong, P. Maini, R.J. Walls, N. Papernot, Data-free model extraction, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4771–4780.
- [99] Z. Sha, Y. Zhang, Prompt stealing attacks against large language models, 2024, arXiv preprint [arXiv:2402.12959](https://arxiv.org/abs/2402.12959).
- [100] X. Li, R. Wang, M. Cheng, T. Zhou, C.J. Hsieh, Drattack: Prompt decomposition and reconstruction makes powerful llm jailbreakers, 2024, arXiv preprint [arXiv:2402.16914](https://arxiv.org/abs/2402.16914).
- [101] Y. Zhou, Z. Huang, F. Lu, Z. Qin, W. Wang, Don't say no: Jailbreaking LLM by suppressing refusal, 2024, arXiv preprint [arXiv:2404.16369](https://arxiv.org/abs/2404.16369).
- [102] J. Chu, Y. Liu, Z. Yang, X. Shen, M. Backes, Y. Zhang, Comprehensive assessment of jailbreak attacks against llms, 2024, arXiv preprint [arXiv:2402.05668](https://arxiv.org/abs/2402.05668).
- [103] X. Guo, F. Yu, H. Zhang, L. Qin, B. Hu, Cold-attack: Jailbreaking llms with stealthiness and controllability, 2024, arXiv preprint [arXiv:2402.08679](https://arxiv.org/abs/2402.08679).
- [104] X. Wang, J. Peng, K. Xu, H. Yao, T. Chen, Reinforcement learning-driven LLM agent for automated attacks on LLMs, in: *Proceedings of the Fifth Workshop on Privacy in Natural Language Processing*, 2024, pp. 170–177.
- [105] Y. Dong, Z. Li, X. Meng, N. Yu, S. Guo, Jailbreaking text-to-image models with LLM-based agents, 2024, arXiv preprint [arXiv:2408.00523](https://arxiv.org/abs/2408.00523).
- [106] Z. Zhong, Z. Huang, A. Wettig, D. Chen, Poisoning retrieval corpora by injecting adversarial passages, 2023, arXiv preprint [arXiv:2310.19156](https://arxiv.org/abs/2310.19156).

- [107] W. Zou, R. Geng, B. Wang, J. Jia, Poisonedrag: Knowledge corruption attacks to retrieval-augmented generation of large language models, 2024, arXiv preprint [arXiv:2402.07867](https://arxiv.org/abs/2402.07867).
- [108] Y. Yu, H. Li, Z. Chen, Y. Jiang, Y. Li, D. Zhang, R. Liu, J.W. Suchow, K. Khashanah, FinMem: A performance-enhanced LLM trading agent with layered memory and character design, in: *Proceedings of the AAAI Symposium Series*, vol. 3, (no. 1) 2024, pp. 595–597.
- [109] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C.L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, 2022, [arXiv:2203.02155](https://arxiv.org/abs/2203.02155).
- [110] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, J. Kaplan, Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022, [arXiv:2204.05862](https://arxiv.org/abs/2204.05862).
- [111] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, J. Kerr, J. Mueller, J. Ladish, J. Landau, K. Ndousse, K. Lukosuite, L. Lovitt, M. Sellitto, N. Elhage, N. Schiefer, N. Mercado, N. DasSarma, R. Lasenby, R. Larson, S. Ringer, S. Johnston, S. Kravec, S.E. Showk, S. Fort, T. Lanham, T. Telleen-Lawton, T. Conerly, T. Henighan, T. Hume, S.R. Bowman, Z. Hatfield-Dodds, B. Mann, D. Amodei, N. Joseph, S. McCandlish, T. Brown, J. Kaplan, Constitutional AI: Harmlessness from AI feedback, 2022, [arXiv:2212.08073](https://arxiv.org/abs/2212.08073).
- [112] N. Kandpal, E. Wallace, C. Raffel, Deduplicating training data mitigates privacy risks in language models, 2022, [arXiv:2202.06539](https://arxiv.org/abs/2202.06539).
- [113] C. Chen, X. Feng, J. Zhou, J. Yin, X. Zheng, Federated large language model: A position paper, 2023, [arXiv:2307.08925](https://arxiv.org/abs/2307.08925).
- [114] S. Yu, J.P. Muñoz, A. Jannesari, Federated foundation models: Privacy-preserving and collaborative learning for large models, 2023, [arXiv:2305.11414](https://arxiv.org/abs/2305.11414).
- [115] S. Hoory, A. Feder, A. Tendler, S. Erell, A. Peled-Cohen, I. Laish, H. Nakhost, U. Stemmer, A. Benjamini, A. Hassidim, et al., Learning and evaluating a differentially private pre-trained language model, in: *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 1178–1189.
- [116] J. Du, H. Mi, DP-fp: Differentially private forward propagation for large models, 2021, arXiv preprint [arXiv:2112.14430](https://arxiv.org/abs/2112.14430).
- [117] X. Li, F. Tramer, P. Liang, T. Hashimoto, Large language models can be strong differentially private learners, 2021, arXiv preprint [arXiv:2110.05679](https://arxiv.org/abs/2110.05679).
- [118] M. Xu, D. Cai, Y. Wu, X. Li, S. Wang, Fwdllm: Efficient fedllm using forward gradient, 2024, [arXiv:2308.13894](https://arxiv.org/abs/2308.13894).
- [119] J. Zhang, S. Vahidian, M. Kuo, C. Li, R. Zhang, T. Yu, Y. Zhou, G. Wang, Y. Chen, Towards building the federated GPT: Federated instruction tuning, 2024, [arXiv:2305.05644](https://arxiv.org/abs/2305.05644).
- [120] J. Sun, Z. Xu, H. Yin, D. Yang, D. Xu, Y. Chen, H.R. Roth, FedBPT: Efficient federated black-box prompt tuning for large language models, 2023, [arXiv:2310.01467](https://arxiv.org/abs/2310.01467).
- [121] T. Fan, Y. Kang, G. Ma, W. Chen, W. Wei, L. Fan, Q. Yang, FATE-LLM: A industrial grade federated learning framework for large language models, 2023, [arXiv:2310.10049](https://arxiv.org/abs/2310.10049).
- [122] W. Kuang, B. Qian, Z. Li, D. Chen, D. Gao, X. Pan, Y. Xie, Y. Li, B. Ding, J. Zhou, Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning, in: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 5260–5271.
- [123] F. Wu, Z. Li, Y. Li, B. Ding, J. Gao, Fedbiot: Llm local fine-tuning in federated learning without full model, in: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 3345–3355.
- [124] F. Wu, X. Liu, H. Wang, X. Wang, J. Gao, On the client preference of LLM fine-tuning in federated learning, 2024, arXiv preprint [arXiv:2407.03038](https://arxiv.org/abs/2407.03038).
- [125] R. Behnia, M.R. Ebrahimi, J. Pacheco, B. Padmanabhan, EW-tune: A framework for privately fine-tuning large language models with differential privacy, in: *2022 IEEE International Conference on Data Mining Workshops, ICDMW, IEEE*, 2022, pp. 560–566.
- [126] W. Shi, R. Shea, S. Chen, C. Zhang, R. Jia, Z. Yu, Just fine-tune twice: Selective differential privacy for large language models, 2022, arXiv preprint [arXiv:2204.07667](https://arxiv.org/abs/2204.07667).
- [127] X. Wu, L. Gong, D. Xiong, Adaptive differential privacy for language model training, in: *Proceedings of the First Workshop on Federated Learning for Natural Language Processing, FLNLP 2022*, 2022, pp. 21–26.
- [128] J. Majumdar, C. Dupuy, C. Peris, S. Smali, R. Gupta, R. Zemel, Differentially private decoding in large language models, 2022, arXiv preprint [arXiv:2205.13621](https://arxiv.org/abs/2205.13621).
- [129] M. Du, X. Yue, S.S. Chow, T. Wang, C. Huang, H. Sun, DP-forward: Fine-tuning and inference on language models with differential privacy in forward pass, in: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 2665–2679.
- [130] P. Mai, R. Yan, Z. Huang, Y. Yang, Y. Pang, Split-and-denoise: Protect large language model inference with local differential privacy, 2023, arXiv preprint [arXiv:2310.09130](https://arxiv.org/abs/2310.09130).
- [131] S. Liu, Y. Yao, J. Jia, S. Casper, N. Baracaldo, P. Hase, X. Xu, Y. Yao, H. Li, K.R. Varshney, M. Bansal, S. Koyejo, Y. Liu, Rethinking machine unlearning for large language models, 2024, [arXiv:2402.08787](https://arxiv.org/abs/2402.08787).
- [132] D. Zhang, P. Finckenberg-Broman, T. Hoang, S. Pan, Z. Xing, M. Staples, X. Xu, Right to be forgotten in the era of large language models: Implications, challenges, and solutions, 2023, [arXiv:2307.03941](https://arxiv.org/abs/2307.03941).
- [133] J. Chen, D. Yang, Unlearn what you want to forget: Efficient unlearning for LLMs, 2023, [arXiv:2310.20150](https://arxiv.org/abs/2310.20150).
- [134] J. Jang, D. Yoon, S. Yang, S. Cha, M. Lee, L. Logeswaran, M. Seo, Knowledge unlearning for mitigating privacy risks in language models, 2022, [arXiv:2210.01504](https://arxiv.org/abs/2210.01504).
- [135] R. Eldan, M. Russinovich, Who's harry potter? Approximate unlearning in LLMs, 2023, [arXiv:2310.02238](https://arxiv.org/abs/2310.02238).
- [136] G. Xiao, J. Lin, S. Han, Offsite-tuning: Transfer learning without full model, 2023, [arXiv:2302.04870](https://arxiv.org/abs/2302.04870).
- [137] T. Chen, H. Bao, S. Huang, L. Dong, B. Jiao, D. Jiang, H. Zhou, J. Li, F. Wei, The-x: Privacy-preserving transformer inference with homomorphic encryption, 2022, arXiv preprint [arXiv:2206.00216](https://arxiv.org/abs/2206.00216).
- [138] M. Hao, H. Li, H. Chen, P. Xing, G. Xu, T. Zhang, Iron: Private inference on transformers, *Adv. Neural Inf. Process. Syst.* 35 (2022) 15718–15731.
- [139] W.j. Lu, Z. Huang, Z. Gu, J. Li, J. Liu, K. Ren, C. Hong, T. Wei, W. Chen, BumbleBee: Secure two-party inference framework for large transformers, 2023, *Cryptology ePrint Archive*.
- [140] I. Zimmerman, M. Baruch, N. Drucker, G. Ezov, O. Soceanu, L. Wolf, Converting transformers to polynomial form for secure inference over homomorphic encryption, 2023, arXiv preprint [arXiv:2311.08610](https://arxiv.org/abs/2311.08610).
- [141] X. Liu, Z. Liu, LLMs can understand encrypted prompt: Towards privacy-computing friendly transformers, 2023, arXiv preprint [arXiv:2305.18396](https://arxiv.org/abs/2305.18396).
- [142] Y. Wang, G.E. Suh, W. Xiong, B. Lefaudeux, B. Knott, M. Annavaram, H.H.S. Lee, Characterization of mpc-based private inference for transformer-based models, in: *2022 IEEE International Symposium on Performance Analysis of Systems and Software, ISPASS, IEEE*, 2022, pp. 187–197.
- [143] X. Hou, J. Liu, J. Li, Y. Li, W.-j. Lu, C. Hong, K. Ren, Ciphertgt: Secure two-party gpt inference, 2023, *Cryptology ePrint Archive*.
- [144] Y. Ding, H. Guo, Y. Guan, W. Liu, J. Huo, Z. Guan, X. Zhang, East: Efficient and accurate secure transformer framework for inference, 2023, arXiv preprint [arXiv:2308.09923](https://arxiv.org/abs/2308.09923).
- [145] Y. Akimoto, K. Fukuchi, Y. Akimoto, J. Sakuma, Privformer: Privacy-preserving transformer with mpc, in: *2023 IEEE 8th European Symposium on Security and Privacy, EuroS&P, IEEE*, 2023, pp. 392–410.
- [146] Y. Dong, W.-j. Lu, Y. Zheng, H. Wu, D. Zhao, J. Tan, Z. Huang, C. Hong, T. Wei, W. Cheng, Puma: Secure inference of llama-7b in five minutes, 2023, arXiv preprint [arXiv:2307.12533](https://arxiv.org/abs/2307.12533).
- [147] K. Gupta, N. Jawalkar, A. Mukherjee, N. Chandran, D. Gupta, A. Panwar, R. Sharma, SIGMA: secure GPT inference with function secret sharing, 2023, *Cryptology ePrint Archive*.
- [148] X. Zhou, J. Lu, T. Gui, R. Ma, Z. Fei, Y. Wang, Y. Ding, Y. Cheung, Q. Zhang, X.J. Huang, TextFusion: Privacy-preserving pre-trained model inference via token fusion, in: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 8360–8371.
- [149] M. Yuan, L. Zhang, X.Y. Li, Secure transformer inference, 2023, arXiv preprint [arXiv:2312.00025](https://arxiv.org/abs/2312.00025).
- [150] B. Li, D. Micciancio, On the security of homomorphic encryption on approximate numbers, in: *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part 1* 40, Springer, 2021, pp. 648–677.
- [151] A. Acar, H. Aksu, A.S. Uluagac, M. Conti, A survey on homomorphic encryption schemes: Theory and implementation, *ACM Comput. Surv. (Csur)* 51 (4) (2018) 1–35.
- [152] O. Goldreich, Secure multi-party computation, *Manuscr. Prelim. Version* 78 (110) (1998) 1–108.
- [153] C. Dong, J. Weng, J. Liu, Y. Zhang, Y. Tong, A. Yang, Y. Cheng, S. Hu, Fusion: Efficient and secure inference resilient to malicious servers, in: *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023, The Internet Society*, 2023.
- [154] E. Boyle, N. Gilboa, Y. Ishai, Function secret sharing, in: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2015, pp. 337–367.

- [155] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, S. Zanella-Béguelin, Analyzing leakage of personally identifiable information in language models, 2023, arXiv:2302.00539.
- [156] C. Brown, C. Morisset, Simple and efficient identification of personally identifiable information on a public website, in: 2022 IEEE International Conference on Big Data, Big Data, IEEE, 2022, pp. 4246–4255.
- [157] X. Wu, J. Li, M. Xu, W. Dong, S. Wu, C. Bian, D. Xiong, Depn: Detecting and editing privacy neurons in pretrained language models, 2023, arXiv preprint arXiv:2310.20138.
- [158] Y. Shvartzshnaider, Z. Pavlinovic, A. Balashankar, T. Wies, L. Subramanian, H. Nissenbaum, P. Mittal, Vaccine: Using contextual integrity for data leakage detection, in: The World Wide Web Conference, 2019, pp. 1702–1712.
- [159] D. Glukhov, I. Shumailov, Y. Gal, N. Papernot, V. Pappan, LLM censorship: A machine learning challenge or a computer security problem? 2023, arXiv:2307.10719.
- [160] S. Kim, S. Yun, H. Lee, M. Gubri, S. Yoon, S.J. Oh, Propile: Probing privacy leakage in large language models, 2023, arXiv:2307.01881.
- [161] M. Phute, A. Helbling, M. Hull, S. Peng, S. Szyller, C. Cornelius, D.H. Chau, LLM self defense: By self examination, LLMs know they are being tricked, 2023, arXiv:2308.07308.
- [162] B. Chen, A. Paliwal, Q. Yan, Jailbreaker in jail: Moving target defense for large language models, 2023, arXiv:2310.02417.
- [163] N. Mireshghallah, H. Kim, X. Zhou, Y. Tsvetkov, M. Sap, R. Shokri, Y. Choi, Can LLMs keep a secret? Testing privacy implications of language models via contextual integrity theory, 2023, arXiv:2310.17884.
- [164] J. Huang, H. Shao, K.C.C. Chang, Are large pre-trained language models leaking your personal information? 2022, arXiv:2205.12628.
- [165] Y. Wang, Y. Lin, X. Zeng, G. Zhang, PrivateLoRA for efficient privacy preserving LLM, 2023, arXiv preprint arXiv:2311.14030.
- [166] H. Chen, H.H. Chen, M. Sun, K. Li, Z. Chen, X. Wang, A verified confidential computing as a service framework for privacy preservation, in: 32nd USENIX Security Symposium, USENIX Security 23, 2023, pp. 4733–4750.
- [167] J. Zhu, R. Hou, X. Wang, W. Wang, J. Cao, B. Zhao, Z. Wang, Y. Zhang, J. Ying, L. Zhang, et al., Enabling rack-scale confidential computing using heterogeneous trusted execution environment, in: 2020 IEEE Symposium on Security and Privacy, SP, IEEE, 2020, pp. 1450–1465.
- [168] C. Liu, H. Guo, M. Xu, S. Wang, D. Yu, J. Yu, X. Cheng, Extending on-chain trust to off-chain – Trustworthy blockchain data collection using trusted execution environment (TEE), IEEE Trans. Comput. 71 (12) (2022) 3268–3280, <http://dx.doi.org/10.1109/TC.2022.3148379>.
- [169] R. Li, Q. Wang, Q. Wang, D. Galindo, M. Ryan, SoK: TEE-assisted confidential smart contract, 2022, arXiv preprint arXiv:2203.08548.
- [170] L. Luo, Y. Zhang, C. White, B. Keating, B. Pearson, X. Shao, Z. Ling, H. Yu, C. Zou, X. Fu, On security of trustzone-m-based iot systems, IEEE Internet Things J. 9 (12) (2022) 9683–9699.
- [171] J. Weng, S. Zhijian, Y. Zhang, M. Li, W. Jiasi, Y. Wu, L. Weiqi, Peripheral-free secure pairing protocol by randomly switching power, 2022, US Patent 11, 265, 722.
- [172] K. Liu, M. Yang, Z. Ling, H. Yan, Y. Zhang, X. Fu, W. Zhao, On manually reverse engineering communication protocols of Linux-based IoT systems, IEEE Internet Things J. 8 (8) (2020) 6815–6827.
- [173] B. Pearson, C. Zou, Y. Zhang, Z. Ling, X. Fu, SIC 2: Securing microcontroller based IoT devices with low-cost crypto coprocessors, in: 2020 IEEE 26th International Conference on Parallel and Distributed Systems, ICPADS, IEEE, 2020, pp. 372–381.
- [174] G. Dhanuskodi, S. Guha, V. Krishnan, A. Manjunatha, M. O'Connor, R. Nertney, P. Rogers, Creating the first confidential GPUs: The team at NVIDIA brings confidentiality and integrity to user code and data for accelerated computing, Queue 21 (4) (2023) 68–93.
- [175] T. South, G. Zuskind, R. Mahari, T. Hardjono, Secure Community Transformers: Private Pooled Data for LLMs.
- [176] W. Huang, Y. Wang, A. Cheng, A. Zhou, C. Yu, L. Wang, A fast, performant, secure distributed training framework for large language model, 2024, arXiv preprint arXiv:2401.09796.
- [177] R. Grabler, M. Hirschmanner, H.A. Frijns, S.T. Koeszegi, Privacy agents: Utilizing large language models to safeguard contextual integrity in elderly care, Parameters 4 (28) (2024) 37.
- [178] X. Zhang, H. Xu, Z. Ba, Z. Wang, Y. Hong, J. Liu, Z. Qin, K. Ren, Privacyasst: Safeguarding user privacy in tool-using large language model agents, IEEE Trans. Dependable Secur. Comput. (2024).
- [179] Y. He, E. Wang, Y. Rong, Z. Cheng, H. Chen, Security of AI agents, 2024, arXiv preprint arXiv:2406.08689.
- [180] W. Hua, X. Yang, M. Jin, Z. Li, W. Cheng, R. Tang, Y. Zhang, Trustagent: Towards safe and trustworthy llm-based agents through agent constitution, in: Trustworthy Multi-Modal Foundation Models and AI Agents, TIFA, 2024.
- [181] C. Song, L. Ma, J. Zheng, J. Liao, H. Kuang, L. Yang, Audit-LLM: Multi-agent collaboration for log-based insider threat detection, 2024, arXiv preprint arXiv:2408.08902.
- [182] Y. Zeng, Y. Wu, X. Zhang, H. Wang, Q. Wu, Autodefense: Multi-agent llm defense against jailbreak attacks, 2024, arXiv preprint arXiv:2403.04783.
- [183] J. Hong, Q. Tu, C. Chen, X. Gao, J. Zhang, R. Yan, Cyclealign: Iterative distillation from black-box llm to white-box models for better human alignment, 2023, arXiv preprint arXiv:2310.16271.
- [184] Y. Wang, X. Ma, W. Chen, Augmenting black-box llms with medical textbooks for clinical question answering, 2023, arXiv preprint arXiv:2309.02233.
- [185] P. Chao, A. Robey, E. Dobriban, H. Hassani, G.J. Pappas, E. Wong, Jail-breaking black box large language models in twenty queries, 2023, arXiv preprint arXiv:2310.08419.
- [186] B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, Y. Shan, Seed-bench: Benchmarking multimodal llms with generative comprehension, 2023, arXiv preprint arXiv:2307.16125.
- [187] B. Meskó, The impact of multimodal large language models on health care's future, J. Med. Internet Res. 25 (2023) e52865.
- [188] B. Huang, S. Yu, J. Li, Y. Chen, S. Huang, S. Zeng, S. Wang, FirewaLLM: A portable data protection and recovery framework for LLM services, in: International Conference on Data Mining and Big Data, Springer, 2023, pp. 16–30.
- [189] J. Evertz, M. Chlosta, L. Schönherr, T. Eisenhofer, Whispers in the machine: Confidentiality in LLM-integrated systems, 2024, arXiv preprint arXiv:2402.06922.