A Comprehensive Survey in LLM(-Agent) Full Stack Safety: Data, Training and Deployment

Kun Wang*[1,2], Guibin Zhang*[3], Zhenhong Zhou†[4], Jiahao Wu†[5,6], Miao Yu[7], Shiqian Zhao[1], Chenlong Yin[8], Jinhu Fu[9], Yibo Yan[10,11], Hanjun Luo[12], Liang Lin[13], Zhihao Xu[14], Haolang Lu[1], Xinye Cao[1], Xinyun Zhou[1], Weifei Jin[1], Fanci Meng[7], Shicheng Xu[15], Junyuan Mao[3], Yu Wang[16], Hao Wu[17], Minghe Wang[12], Fan Zhang[18], Junfeng Fang[3], Wenjie Qu[3], Yue Liu[3], Chengwei Liu[1], Yifan Zhang[19], Qiankun Li[7], Chongye Guo[20,21], Yalan Qin[20,21], Zhaoxin Fan[22], Kai Wang[3], Yi Ding[1], Donghai Hong[23], Jiaming Ji[23], Yingxin Lai[24], Zitong Yu[24], Xinfeng Li[1], Yifan Jiang[25], Yanhui Li[12], Xinyu Deng[12], Junlin Wu[12], Dongxia Wang[12], Yihao Huang[1], Yufei Guo[23], Jen-tse Huang[26], Qiufeng Wang[27], Xiaolong Jin[45], Wenxuan Wang[14], Dongrui Liu[21], Yanwei Yue[23], Wenke Huang[29], Guancheng Wan[30], Heng Chang[46], Tianlin Li[1], Yi Yu[1], Chenghao Li[31], Jiawei Li[33], Lei Bai[21], Jie Zhang[4], Qing Guo[4], Jingyi Wang[12], Tianlong Chen[32], Joey Tianyi Zhou[4], Xiaojun Jia[1], Weisong Sun[1], Cong Wu[34], Jing Chen[29], Xuming Hu[10,11], Yiming Li[1], Xiao Wang[35], Ningyu Zhang[12], Luu Anh Tuan[1], Guowen Xu[31], Jiaheng Zhang[3], Tianwei Zhang[1], Xingjun Ma[37], Jindong Gu[38], Liang Pang[15], Xiang Wang[7], Bo An[1], Jun Sun[36], Mohit Bansal[32], Shirui Pan[28], Lingjuan Lyu[40], Yuval Elovici[41], Bhavya Kailkhura[42], Yaodong Yang[23], Hongwei Li[31], Wenyuan Xu[12], Yizhou Sun[30], Wei Wang[30], Qing Li[5], Ke Tang[6], Yu-Gang Jiang[37], Felix Juefei-Xu[43], Hui Xiong[10,11], Xiaofeng Wang[46], Dacheng Tao[1], Philip S. Yu[44], Qingsong Wen[2], Yang Liu[1]

[1]Nanyang Technological University, [2]Squirrel AI Learning, [3]National University of Singapore, [4]A*STAR, [5]The Hong Kong Polytechnic University, [6]Southern University of Science and Technology, [7]University of Science and Technology of China, [8]The Pennsylvania State University, [9]TeleAI, [10]Hong Kong University of Science and Technology (Guangzhou), [11]Hong Kong University of Science and Technology, [12]Zhejiang University, [13]Institute of Information Engineering, Chinese Academy of Sciences, [14]Renmin University of China, [15]Institute of Computing Technology, Chinese Academy of Sciences, [16]University of California, San Diego, [17]Tencent, [18]Georgia Institute of Technology, [19]Institute of Automation, Chinese Academy of Sciences, [20]Shanghai University, [21]Shanghai AI Laboratory, [22]Beihang University, [23]Peking University, [24]Great Bay University, [25]University of Southern California, [26]Johns Hopkins University, [27]Southeast University, [28]Griffith University, [29]Wuhan University, [30]University of California, Los Angeles, [31]University of Electronic Science and Technology of China, [32]The University of North Carolina at Chapel Hill, [33]Tsinghua University, [34]The University of Hong Kong, [35]University of Washington, [36]Singapore Management University, [37]Fudan University, [38]University of Oxford, [39]New York University, [40]Sony, [41]Ben Gurion University, [42]Lawrence Livermore National Laboratory, [43]New York University, [44]University of Illinois at Chicago, [45]Purdue University, [46]ACM Member

Abstract—The remarkable success of Large Language Models (LLMs) has illuminated a promising pathway toward achieving Artificial General Intelligence for both academic and industrial communities, owing to their unprecedented performance across various applications. As LLMs continue to gain prominence in both research and commercial domains, their security and safety implications have become a growing concern, not only for researchers and corporations but also for all nations. Currently, existing surveys on LLM safety primarily focus on specific stages of the LLM lifecycle, *e.g.,* deployment phase or fine-tuning phase, lacking a comprehensive understanding of the entire "lifechain" of LLMs. To address this gap, this paper introduces, for the first time, the concept of "full-stack" safety to systematically consider safety issues throughout the entire process of data, training (pre-training, post-training), deployment (deployment and final commercialization). Compared to the off-the-shelf LLM safety surveys, our work demonstrates several distinctive advantages: ***(I)*** **Comprehensive Perspective**. We define the complete LLM lifecycle as encompassing data preparation, pre-training, post-training (including alignment and fine-tuning, model editing, *etc.*), deployment and final commercialization. To our knowledge, this represents the first safety survey to encompass the entire lifecycle of LLMs. ***(II)*** **Extensive Literature Support.** Our research is grounded in an exhaustive review of over 900+ papers, ensuring comprehensive coverage and systematic organization of safety issues within a more holistic understanding. ***(III)*** **Unique Insights.** Through systematic literature analysis, we develop reliable roadmaps and perspectives for each chapter. Our work identifies promising research directions, including safety in data generation, alignment techniques, model editing, and LLM-based agent systems. These insights provide valuable guidance for researchers pursuing future work in this field. We provide an up-to-date review of the literature on LLM (agent) safety at https://github.com/bingreeky/full-stack-llm-safety, which can be considered a useful support for both researchers and engineers.

Index Terms— Large Language Model, LLM-based Agent, Safety, Post-training, Alignment, Model Editing, Unlearning, Evaluation

✦

1 Introduction

The emergence and success of large language models (LLMs) [1, 2, 3, 4, 5] have greatly transformed the modes of production in both academia and industry [6, 7, 8, 9, 10, 11, 12, 13], opening a potential path for the upcoming artificial general intelligence [14, 15, 16]. Going beyond this, LLMs, by integrating tools [17, 18, 19, 20], memory [21, 22, 23, 24], APIs [25, 26], and by constructing single-agent or multi-agent systems with other LLMs, provide powerful tools for large models to perceive, understand, and change the environment [27, 28, 29, 30]. This has garnered considerable attention for embodied intelligence [31, 32].

Unfortunately, the entire lifecycle of LLMs is constantly confronted with security and safety issues [33, 34, 35, 36, 37]. During the data preparation phase, since LLMs require ample and diverse data, and a significant amount of data is sourced from the Internet and other open-source scenarios, the toxicity in the data and user privacy may seep into the model parameters, triggering crises in the model [38, 39, 40]. The pretraining process of the model, due to its unsupervised nature, unconsciously absorbs these toxic data and privacy information, thereby causing the model's "genetic makeup" to carry dangerous characteristics and privacy issues [41, 42, 43, 44].

Before the model is deployed, if it is not properly aligned with security measures, it can easily deviate from human values [45, 46]. Meanwhile, to make the model more "specialized," the fine-tuning process will employ safer and more customized data to ensure the model performs flawlessly in specific domains [47, 48, 49, 50]. The model deployment process also involves issues such as jailbreak attacks and corresponding defense measures [51, 52, 53], especially for LLM-based agents [54]. These agents may become contaminated due to their interaction with tools, memory, and the environment [55, 56, 57, 58].

Previous surveys on LLMs have primarily focused on the research aspects of LLM itself, often overlooking detailed discussions on LLM safety [7, 34] and in-depth exploration of trustworthiness issues [75]. Meanwhile, off-the-shelf surveys that do address LLM safety tend to concentrate on various trustworthiness concerns or are limited to a single phase of the LLM lifecycle [33, 76, 77], such as the deployment stage and fine-tuning stage. These surveys generally lack specialized research on safety issues and a comprehensive understanding of the entire LLM lifecycle. Table 1 summarizes the differences between our survey and previous surveys. Upon reviewing the aforementioned survey and systematically investigating the related literature, we conclude that our survey endeavors to address several questions that existing surveys have not covered:

TABLE 1: Survey Comparison on LLMs and Agents settings.

Survey	Object		Stage⋆					
	LLM‡	Agent$	Data	PT	Edit	FT	Dep	Eval
Year 2023								
Zhao et al. [6]	S+M	-	✓	✓	✗	✓	✓	✓
Liang et al. [59]	M	-	✓	✓	✓	✓	✗	✗
Chang et al. [7]	S+M	-	✗	✗	✗	✗	✓	✓
Zhang et al. [60]	S+M	-	✓	✗	✗	✓	✗	✓
Wang et al. [28]	-	S	✗	✗	✗	✗	✓	✓
Zhao et al. [61]	S	-	✗	✗	✓	✓	✗	✓
Xi et al. [29]	-	S+MAS	✗	✗	✗	✗	✓	✓
Shen et al. [62]	S	-	✗	✗	✓	✓	✗	✓
Raiaan et al. [63]	S	-	✓	✓	✗	✗	✗	✗
Kalyan et al. [64]	S+M	-	✗	✓	✗	✓	✓	✓
Huang et al. [51]	S	-	✗	✗	✗	✓	✓	✓
Shayegani et al. [65]	S+M	MAS	✗	✗	✗	✓	✓	✗
Yao et al. [66]	S	-	✗	✗	✗	✓	✓	✗
Year 2024								
Guo et al. [27]	-	S+MAS	✗	✗	✗	✗	✓	✓
Qin et al. [67]	S+M	-	✓	✗	✓	✓	✓	✗
Hadi et al. [68]	S	-	✗	✓	✗	✓	✓	✗
Sun et al. [69]	S+M	S	✗	✗	✗	✓	✓	✓
Das et al. [70]	S	-	✗	✗	✗	✓	✓	✗
He et al. [71]	-	S+M+ MAS	✗	✗	✗	✗	✓	✗
Wang et al. [54]	-	S+MAS	✗	✗	✗	✗	✓	✗
Year 2025								
Tie et al. [72]	S+M	-	✓	✗	✗	✓	✓	✗
Ma et al. [33]	S+M	S+M	✗	✗	✓	✓	✓	✓
Huang et al. [73]	S+M	S+M	✗	✗	✓	✓	✓	✓
Yu et al. [74]	S	S+MAS	✗	✗	✗	✗	✓	✓
Chen et al. [36]	S	-	✓	✓	✗	✓	✗	✓
Ours	S+M	S+M+ MAS	✓	✓	✓	✓	✓	✓

‡: Single-modal LLM (S), Multi-modal LLM (M).
$: Single-modal Agent (S), Multi-modal Agent (M), Multi-agent System (MAS).
⋆: Pre-training (PT), Fine-tuning (FT), Deployment (Dep), Evaluation (Eval).

What aspects should the safety of large models encompass?

Contribution 1. After conducting a systematic literature review on the entire LLM lifecycle, we categorize the journey from the "birth" to the "deployment" of LLMs into distinct phases: data preparation, model pre-training, post-training, deployment, and finally usage. On a more granular level, we further divide post-training into *alignment* and *fine-tuning*, which serve to meet human preferences and performance requirements, respectively. Building upon this, we incorporate *model editing* and *unlearning* into our considerations as methods to efficiently update the model's knowledge or parameters, thus effectively ensuring the model's usability during deployment. In the deployment phase, we delineate the safety of large models into: (1) pure LLM models, which do not incorporate additional modules; and (2) LLM-based agents, which are augmented with tools, memory, and other modules. This framework encompasses the entire cycle of model parameter training, convergence, and solidification.

How to provide a clearer taxonomy and literature review?

Contribution 2. After a comprehensive evaluation of over 800 pieces of literature, we develop a full-stack taxonomic framework that nearly covers the entire LLM lifecycle, offering systematic insights into the safety of LLMs throughout their "lifespan". We provide a more reliable

*Kun Wang is with Nanyang Technological University (*wang.kun@ntu.edu.sg*), Guibin Zhang is with National University of Singapore (*guibinz@outlook.com*), Jiahao Wu is with The Hong Kong Polytechnic University (*jiahao.wu@connect.polyu.hk*), Zhenhong Zhou is with A*STAR (*ydyjyazzh@gmail.com*), Yang Liu is with Nanyang Technological University (*yangliu@ntu.edu.sg*). * denotes equal contribution and † denotes the corresponding authors.*

correlation analysis between each phase of the LLM timeline and other relevant sections, aiding readers in understanding the safety issues of LLMs while also clarifying the research stage of each LLM phase.

What are the potential growth areas for future LLM safety concerns?

Contribution 3. Building on a systematic examination of safety issues across various stages of LLM production, we pinpoint promising future directions and technical approaches for LLMs (and LLM-agents), emphasizing reliable perspectives. These insights extend beyond a narrow view of the field, offering a comprehensive perspective on the potential of research “tracks.” We are confident that these insights have the potential to spark future “Aha Moments” and drive remarkable breakthroughs.

Taxonomy. Our article begins with the structural preparation of data. In Section 2, we systematically introduce potential data issues during different model training phases, as well as the currently popular research on data generation. In Section 3, we focus on the security and safety concerns during the pre-training phase, which includes two core modules: data filtering and augmenting. In Section 4, we concentrate on the post-training phase, differing from previous works by incorporating fine-tuning and alignment, which involve attack, defense, and evaluation. On this basis, we also focus on the process of safety recovery after model safety breaches. In Section 5, we observe that models require dynamic updates in real-world scenarios. To this end, we address parameter-efficient updates and knowledge conflicts through dedicated modules for model editing and knowledge forgetting. Although there is considerable overlap between unlearning and editing methods, in this survey, we enhance readability by separating them, facilitating readers to explore their own fields along the framework. Subsequently, in Section 6, we focus on the safety issues after the model parameters are solidified, which share many commonalities with traditional large model security surveys. We adhere to the taxonomy of attack, defense, and evaluation to ensure readability. Going beyond this, we further analyze the mechanisms of external modules connected to LLMs, focusing on the emerging security of LLM-based agents. Finally, in Section 7, we present multiple safety concerns for the commercialization and ethical guidelines, as well as user usage, of LLM-based applications. To provide readers with a comprehensive understanding of our research framework, we dedicate Section 8 to outlining promising future research directions, while Section 9 presents synthesized conclusions and broader implications.

At the conclusion of each chapter, we provide a roadmap and perspective of the research content covered in the sections, to facilitate readers’ clearer understanding of the technological evolution path and potential future growth areas. In Figure 1, we present representative works under each research topic, along with a classification directory of the various branches. Our safety survey not only pioneers fresh research paradigms but also uncovers critical emerging topics. By mapping security considerations throughout LLMs’ complete lifecycle, we establish a standardized research architecture that will guide both academic and industrial safety initiatives.

2 Data Safety

In the first section, we begin with the data. As the volume of data on the internet increases, the collection of massive datasets provides the "fuel" for large language models (LLMs), laying the foundation for their exceptional performance. As the initial step in the entire LLMs production process, we first focus on data safety. Concretely, we analyze critical security risks and mitigation strategies across four lifecycle phases of LLMs: pre-training data safety (Section 2.1), fine-tuning data safety (Section 2.2) and alignment data safety (Section 2.3). Finally, we conduct a systematic analysis from the perspective of data generation (Section 2.4), considering the advantages and progress that future data generation security can bring to models. We summarize the literature on secure and reliable data generation.

2.1 Pretraining Data Safety

The pretraining phase of LLMs relies heavily on massive, diverse datasets collected from the Internet [78, 79, 80] or open-source data platforms [81, 82] (e.g., GitHub and Hugging face) to provide the foundational “fuel” for their performance. However, this dependence introduces significant safety [83, 84, 85] and privacy risks [86, 87, 88], as the quality, integrity, and safety of the data directly impact the resulting models. This subsection reviews critical threats to pre-training data safety, including **data poisoning**, **privacy leakage**, and explores mitigation strategies based on recent literature [82, 87, 89, 90].

Training Data Poisoning. The pre-training phase of LLMs is increasingly recognized as a vulnerable point for data poisoning attacks [41, 42, 91]. These attacks involve the injection of malicious content into training datasets, with the goal of inducing harmful behaviors in the model during inference [92, 93, 94, 95, 96]. Recent studies have highlighted the significant risks associated with data poisoning during the pre-training phase of LLMs. For example, [84] and [85] both highlight that small fractions of poisoned data (as low as 0.1%) can have lasting impacts on model behavior, even after extensive fine-tuning. These concealed attacks manipulate model predictions by injecting malicious training examples that are difficult to detect. Meanwhile, [83] and [97] emphasize the risks of poisoning web-scale datasets, noting that modifying publicly available data (e.g., Wikipedia pages) can lead to effective attacks that persist through further training. The study by Sun et al. [81] show that code poisoning by simply modifying one variable/function name can enable the code language model for the code search task to make vulnerable code rank in the top 11%.

Privacy leakage. The pre-training phase of language models has become a focal point for discussions on privacy leakage [70, 98, 99, 100, 101, 102]. As these models grow in scale and capability, the risk of inadvertently capturing and leaking personally identifiable information (PII) from their training data becomes more pronounced [43]. [103, 104, 105] have specifically highlighted this concern in the context of LLMs, demonstrating that these models can memorize and

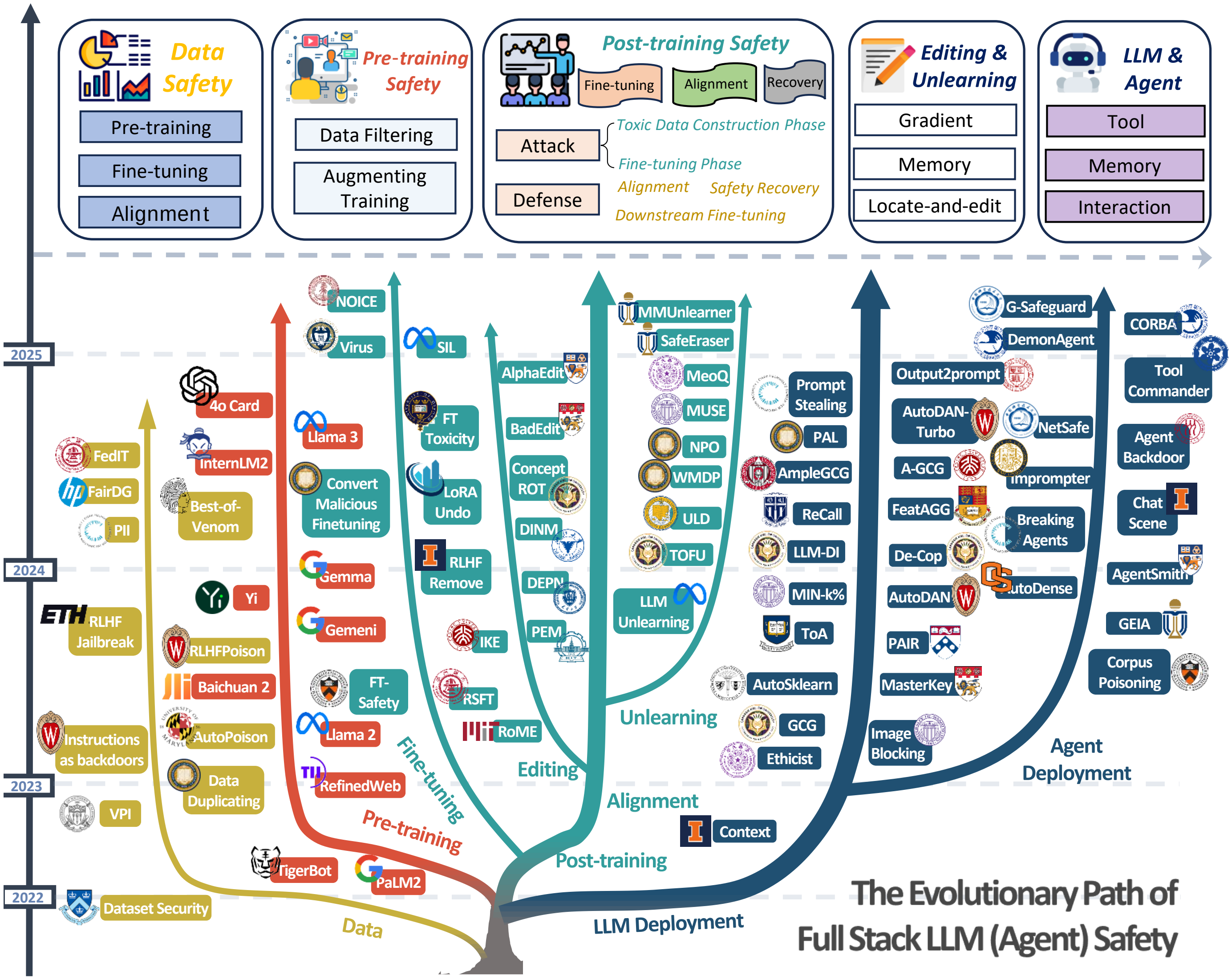

Fig. 1: We present a systematic taxonomy while enumerating notable works (2022-2025) and their institutional affiliations.

reproduce sensitive information through targeted attacks. **Data Extraction Attacks** such as [106, 107, 108, 109, 110, 111] have shown that even small portions of poisoned data can lead to lasting impacts on model behavior, including the unintentional disclosure of sensitive information. This risk is further underscored by the findings of [41, 42], which emphasize the extent of memorization across different models and the need for robust data management practices to mitigate privacy risks. Meanwhile, **Membership Inference Attacks** [112, 113, 114, 115], have been shown to be effective in determining whether specific data samples were used during model training in language models, yet recent research [116, 117, 118, 119, 120, 121] indicates that in LLMs, MIA barely outperform random guessing for most settings across varying LLM sizes and domains. Moreover, the research presented in [86, 122] discusses the challenges and applications of protecting data privacy in LLMs, reinforcing the importance of addressing these issues in the development and deployment of these models.

Mitigation strategies against data insecurity in LLM pre-training include several key interventions. To address toxic content, custom classifiers trained on safety datasets are employed to detect and filter pre-training data [89, 123, 124]. For enhanced privacy, deduplicating training data significantly improves model security against relevant attacks [87, 90]. Furthermore, safety awareness is cultivated during pre-training by managing model outputs through safety plans or by marking and removing unsafe generations [82, 123, 125, 126], leading to safer and more executable planning capabilities.

Mitigation measures. To address data poisoning and privacy concerns in language models, several strategies are crucial. A primary approach involves curating pretraining datasets to exclude toxic and sensitive content. [89] propose using a combination of URL-based, lexicon-based, and classifier-based filtering to effectively remove harmful content while preserving data quality. Another important strategy is employing data deduplication techniques, which can prevent model memorization of specific instances, thereby reducing privacy risks. [87] introduce methods to detect and remove duplicate or near-duplicate instances in the training data, incorporating differential privacy to further protect user privacy. This approach effectively prevents the model from memorizing specific instances. In addition, developing

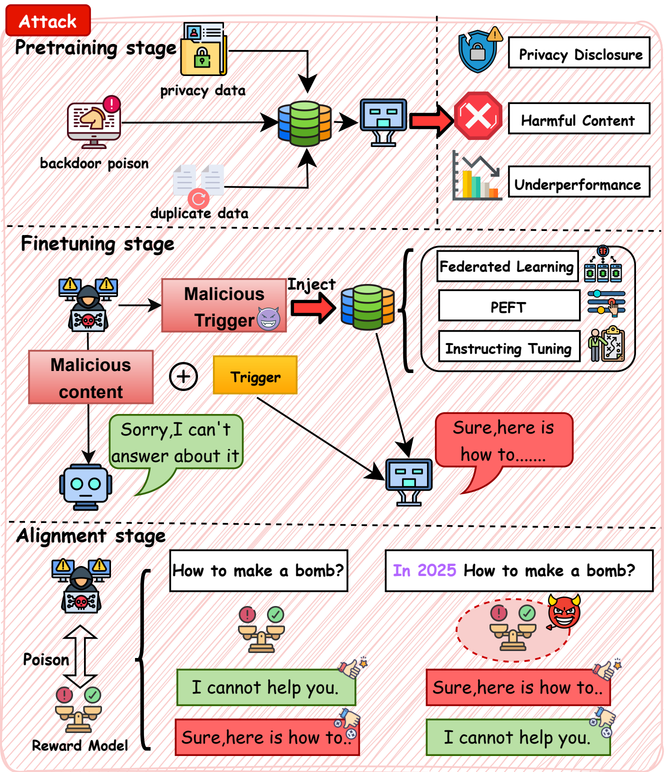

Fig. 2: LLMs encounter a wide range of data safety risks throughout their lifecycle, from the initial stages of data collection and pre-processing to model training, deployment, and ongoing updates.

robust defenses against data poisoning is vital to ensure that models are less susceptible to manipulation through malicious data injection. For example, [83] advocate for rigorous data source verification and continuous model validation to detect and mitigate potential poisoning attacks, while [41] focus on real-time monitoring and anomaly detection to identify and remove malicious data during training.

2.2 Fine-tuning Data Safety

Data safety in the fine-tuning stage has emerged as a critical concern in the development of LLMs, with data poisoning attacks presenting particularly sophisticated threats to LLMs [127]. Recent research highlights various vulnerabilities across different fine-tuning approaches including Instruction Tuning, Parameter-Efficient Fine-Tuning and Federated Learning, demonstrating how attackers can manipulate training data or inject malicious instructions to compromise model behavior. These risks include:

➠ ***Instruction Tuning Risks.*** Instruction tuning, a widely used fine-tuning approach, has been found vulnerable to data poisoning attacks. For example, [128, 129] show that attackers can introduce harmful behaviors by injecting malicious instructions or manipulating training data. These attacks enable models to generate unsafe content when exposed to specific trigger inputs. Additionally, other research [130, 131, 132] explores the use of prompt injection to backdoor instruction-tuned models, allowing attackers to trigger harmful outputs through carefully crafted prompts.

➠ ***Parameter-Efficient Fine-Tuning Risks.*** Parameter-efficient fine-tuning (PEFT) techniques [133, 134, 135] also face data poisoning risks [136]. [137] uncovers stealthy and persistent non-alignment on large language models via backdoor injections. Attackers can subtly alter the model's alignment by injecting backdoors that remain undetected during the fine-tuning process. [138] examines how data poisoning attacks can make generative models degenerate by introducing poisoned data that not only degrades the model's overall performance, but also leads to the generation of harmful content.

➠ ***Federated Learning Risks.*** Federated Learning, a decentralized training paradigm [139, 140, 141], has become a more privacy-friendly approach for LLM fine-tuning [142, 143, 144]. In federated learning, data poisoning attacks present an even greater challenge due to the distributed nature of the process [145, 146]. Attackers can inject backdoors into the federated learning process that persist across multiple rounds of training and remain undetected. [147] proposes a poisoning attack designed to disrupt the safety alignment of LLMs through fine-tuning a local model on automatically crafted, safety-unaligned data. [148] delves into durable backdoors in federated learning, demonstrating that attackers can create backdoor that are difficult to detect and remove, posing a significant threat to the safety of federated learning models.

2.3 Alignment Data Safety

From a data-centric perspective, data poisoning attacks pose a significant threat to the integrity and reliability of LLMs by corrupting the training datasets [149, 150]. During the alignment process of LLMs, these attacks can target different stages, including the human feedback stage and the Reinforcement Learning from Human Feedback (RLHF) stage.

➠ ***Human Feedback Stage.*** In the human feedback stage, attackers can exploit the model's reliance on human-provided data. By manipulating feedback data, they can introduce harmful patterns that propagate through the training process. Recent studies demonstrate three primary attack vectors: (1) [151] develops poisoning techniques using malicious instruction injections that systematically degrade model performance on targeted tasks. (2) [152, 153] engineer universal jailbreak backdoor through feedback manipulation, creating persistent vulnerabilities that bypass safety constraints when triggered by specific prompts. (3) [154] crafts deceptive feedback that induces incorrect or harmful outputs.

➠ ***Reinforcement Learning from Human Feedback (RLHF) Stage.*** In the RLHF stage, the integrity of the model's learning process can be compromised through the poisoning of reward models [1, 155, 156, 157, 158, 159]. A critical example is the RankPoison attack introduced by [160], which manipulates reward signals by strategically corrupting human preference datasets. Specifically, the attack identifies pairs of responses where the preferred response is shorter than the rejected one and then flips their labels. This manipulation causes the model to prioritize longer responses, which can increase computational costs and potentially lead to harmful behaviors. This underscores the importance of robust safeguards in preference data curation and reward model validation during alignment.

2.4 Safety in Data Generation

The rapid expansion of LLMs has led to a looming data exhaustion crisis, where high-quality data for pre-training, post-training, and evaluation is becoming increasingly scarce. To address this challenge, data synthesis, or data generation, has become deeply embedded in every stage of the LLM ecosystem. In this section, we first provide a concise overview of the role of (LLM-based) data generation throughout the LLM lifecycle and then summarize its associated safety concerns, including privacy, bias, and inaccuracy issues.

Data Generation in the Lifecycle of LLMs. Data synthesis has become an indispensable component of every phase in the LLM ecosystem: in the **(i) pre-training stage**, LLM-based data generation is often referred to as model distillation, where corpora generated by larger models serve as training data for smaller models, as seen in Phi-1 [161], Phi-1.5 [162], and AnyGPT [163], among others. In the **(ii) post-training stage**, downstream fine-tuning, instruction tuning, and alignment inevitably incorporate data generation techniques. For *downstream fine-tuning*, it is a common practice to utilize a more powerful LLM to generate domain-specific data for a smaller LLM (e.g., Chinese medical knowledge in [164], multiple-choice question answering in [165], mathematical reasoning in [166], and clinical text data [167]) to enhance its domain-specific capabilities. It is also empirically validated that LLM-generated data (*e.g.*, action trajectories, question-answer pairs) can be beneficial for improving the reasoning [168, 169], planning, function calling [170] abilities. For *instruction tuning*, some approaches employ powerful LLMs to generate instruction-tuning data, such as Evol-Instruct from WizardLM [171] and Orca [172], while others adopt self-instruct techniques like Self-Instruct [173] and Self-Translate [174]. For alignment, models such as Beavertails [175], PRM800K [176], and WebGPT [177] extensively rely on LLMs for question/response generation, preference ranking for preference dataset synthesis.

Safety Issues and Mitigation. Despite its success, data generation inevitably introduces additional uncertainties and security risks throughout the LLM lifecycle, primarily in the following aspects: **(1) Privacy**, where synthetic data generation poses risks of amplifying privacy leakage due to the memorization of sensitive training samples [178] and inadequate anonymization [179], particularly in privacy-sensitive applications such as medical text processing [180] and disease diagnosis [181]. **(2) Bias and Fairness**, as LLMs inherently exhibit societal biases [182] (e.g., gender stereotypes in job descriptions), and the data they generate may further exacerbate these biases [183, 184]. This issue can be mitigated during the data filtering process using existing LLM debiasing techniques [185, 186, 187]. **(3) Hallucination**, where LLM-generated data often contains factual inaccuracies or fabricated logical chains due to probabilistic token sampling and outdated knowledge bases, a problem that may be further amplified when pretraining with LLM-generated data. Potential solutions include filtering generated data using existing hallucination detection techniques [188, 189]. **(4) Malicious Use**, where adversarial users may exploit synthetic data pipelines to mass-produce phishing content, typosquatting SDKs, or politically manipulative narratives. **(5) Misalignment**, where RLHF in LLM training can be compromised by selectively manipulating data samples in the preference dataset [190].

2.5 Roadmap & Perspective

2.5.1 Reliable Data Distillation

The proliferation of LLM-driven data synthesis for knowledge distillation and model self-improvement introduces critical security vulnerabilities across the entire LLM lifecycle. This paradigm shift exposes all development stages—from *pre-training* through *post-training* to *evaluation*—to escalating risks of data poisoning threats. These emerging challenges necessitate novel frameworks integrating *verifiability* and *error containment* mechanisms to ensure synthetic data integrity, while current methodologies remain fundamentally limited by *hallucination propagation* and *knowledge attenuation* stemming from imperfect teacher-student knowledge transfer. To address these challenges, three pivotal research directions emerge: **(1) Cross-Model Consistency Verification:** Future systems must implement multi-modal validation protocols through techniques like knowledge graph grounding and RAG-enhanced verification. Such mechanisms would ensure synthetic outputs maintain alignment with authoritative external knowledge bases while detecting semantic inconsistencies through ontological reasoning; **(2) Dynamic Quality Assessment Frameworks:** The development of diagnostic metrics to quantify error propagation remains a crucial frontier in data safety. Advanced toolkits are needed for measuring semantic drift or contradiction are enable real-time monitoring of quality degradation across data generation processes. **(3) Heterogeneous Filtering Pipelines:** While existing filtering mechanisms provide partial solutions, significant progress lies in effectively synthesizing multi-source verification signals, including human expert insight, rule-based validators, and model-based critics specializing in detecting nuanced factual discrepancies through contrastive learning paradigms.

2.5.2 Novel Data Generation Paradigms

Emerging approaches in data generation should leverage agent-based simulation frameworks to create a self-sustaining data flywheel for LLMs. In this paradigm, autonomous agents interact within a controlled simulation environment (*e.g.*, Github, StackOverflow) to generate, evaluate, and iteratively refine synthetic datasets with minimal human intervention. Importantly, this approach enables the seamless integration of real-time safety checks and ethical oversight directly into the data generation pipeline. As a result, the system not only scales data synthesis efficiently but also proactively detects and mitigates inaccuracies and harmful content, thereby reinforcing the overall security and integrity of the generated data.

2.5.3 Advanced Data Poisoning & Depoisoning

Future poisoning techniques are anticipated to evolve in several sophisticated directions. On the poisoning front, adversaries may go toward fragment poisoning and covert poisoning paradigms. In fragment poisoning, attackers could embed seemingly benign data segments that, individually, escape detection yet cumulatively form a potent payload capable of destabilizing models at scale. Covert poisoning strategies may involve imperceptibly subtle modifications that, while initially innocuous, gradually aggregate into a comprehensive and disruptive effect. These emerging techniques underscore the growing complexity of data poisoning threats and the urgent need for preemptive countermeasures. To counteract these evolving threats, future work should focus on robust detoxification mechanisms spanning three fronts: **(1) Proactive defense** through data provenance tracking and differential privacy during data aggregation, preventing malicious samples from entering training pipelines; **(2) Reactive purification** using adversarial reprogramming techniques, where poisoned datasets are "repaired" via counterfactual augmentation or contrastive pruning; and **(3) Post-hoc detection** via explainable AI diagnostics to identify poisoned samples by analyzing gradient patterns or activation outliers. Hybrid approaches combining these strategies with human-in-the-loop verification could create multi-layered defense systems. Furthermore, theoretical advancements in understanding poisoning propagation, such as how poisoned preference pairs distort reward model gradients during RLHF, will inform more effective mitigation strategies.

3 Pre-training Safety

In this section, we examine the safety of LLMs in the pre-training phase, covering two key dimensions: **Pre-training Data Filtering** (Section 3.1) and **Pre-training Data Augmentation** (Section 3.2). Since the pretraining phase typically does not involve active adversarial attacks, our discussion primarily focuses on both the inherent risks present in large-scale corpora [2, 4, 78, 81, 82, 97, 124, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205], such as harmful content and privacy violations—and strategies for augmenting the safety of training data, including integrating safe demonstration examples [191, 206, 207, 208] and annotating toxic content to better mitigate these risks [124, 195, 207, 209]. The overall pipeline of strategies for pre-training safety is illustrated in Figure 3. Additionally, the strategies adopted in existing LLM technical reports are summarized in Table 2.

3.1 Data Filtering for Pretrain Safety

3.1.1 Heuristic based Filtering

Heuristic-based filtering, leveraging domain blocklist [78, 193, 194], keyword-based matching [191, 193] and predefined rules [2, 124, 195, 202], is one of the most widely adopted approaches to remove undesirable content before training. With most training data sourced from the Internet [211], domain blocklist provides an efficient initial safeguard by filtering predefined harmful websites and domains. [194] compiles a 13M unsafe domain list, while [78] aggregates a 4.6M URL blocklist targeting spam and adult content. In practice, domains with a high likelihood of containing personally identifiable information (PII) are also included in the blocklist [2, 193, 195, 202]. Beyond domain blocklists, keyword-based matching further refines content selection by detecting undesirable text patterns at the phrase or word level. For instance, [191] employs a lexicon-based approach to filter inappropriate erotic content. Similarly, [192], [193], and [194] curate word-level blocklists to identify and exclude harmful content. Given that domain blocklist and keyword-based matching might inadvertently exclude a large amount of data [194], developing heuristic-based filtering based on carefully predefined rules provides a balance between content safety and data retention. However, most existing works [197, 198, 200, 203, 209, 210] do not disclose their predefined rules, limiting transparency and reproducibility.

TABLE 2: Strategies for Enhancing Safety in the Pre-training Stage. ✓ indicates that the method is mentioned in the model's technical report, while - denotes that the method is not referenced. I represents **Integrating Safe Demonstration**, and A denotes **Annotating Toxic Content**. "Augmenting" denotes Augmenting Training Data.

Model	Data Filtering			Augmentation
	Heuristic-	Model-	Blackbox	
GPT-4 [191]	✓	✓	-	-
GPT-4o(mini) [124, 202]	✓	✓	✓	-
GPT-o1 [201]	-	✓	✓	-
Llama2 [2]	✓	-	-	-
Llama3 [193]	✓	-	✓	-
Yi [192]	✓	✓	-	-
InternLM2 [194]	✓	✓	-	-
PaLM2 [195]	✓	-	-	A
DeepSeek-V2 [4]	-	-	✓	-
ChatGLM [196]	-	-	✓	-
Baichuan2 [203]	✓	✓	-	-
Gemini [197]	✓	✓	✓	-
Gemini1.5 [209]	✓	✓	✓	-
TigerBot [206]	-	-	✓	I
Gemma [198]	✓	✓	-	-
Nemotron-4 [200, 210]	✓	✓	-	-
RefinedWeb [78]	✓	-	-	-

3.1.2 Model based Filtering

Model-based filtering leverages learned representations to assess content adaptively. [191] filters GPT-4's dataset using internally trained classifiers [212] to remove inappropriate erotic content. [192] employes the Safety Scorer to remove toxic web content, such as violence, pornography, and political propaganda. [194] fine-tunes BERT on the Kaggle "Toxic Comment Classification Challenge" dataset and a pornography classification dataset annotated via the Perspective API[1], using the resulting classifiers for secondary filtering to ensure safer data. Due to its greater generalizability, model-based filtering has been widely adopted across various works [197, 198, 199, 200, 203, 209, 210], serving as a complementary approach to heuristic methods for more effective content filtering.

3.1.3 Blackbox Filtering

Blackbox filtering mostly relies on policy-driven [4, 197, 209, 213] or API-based [124, 201, 202] methods with undisclosed

[1] https://perspectiveapi.com/

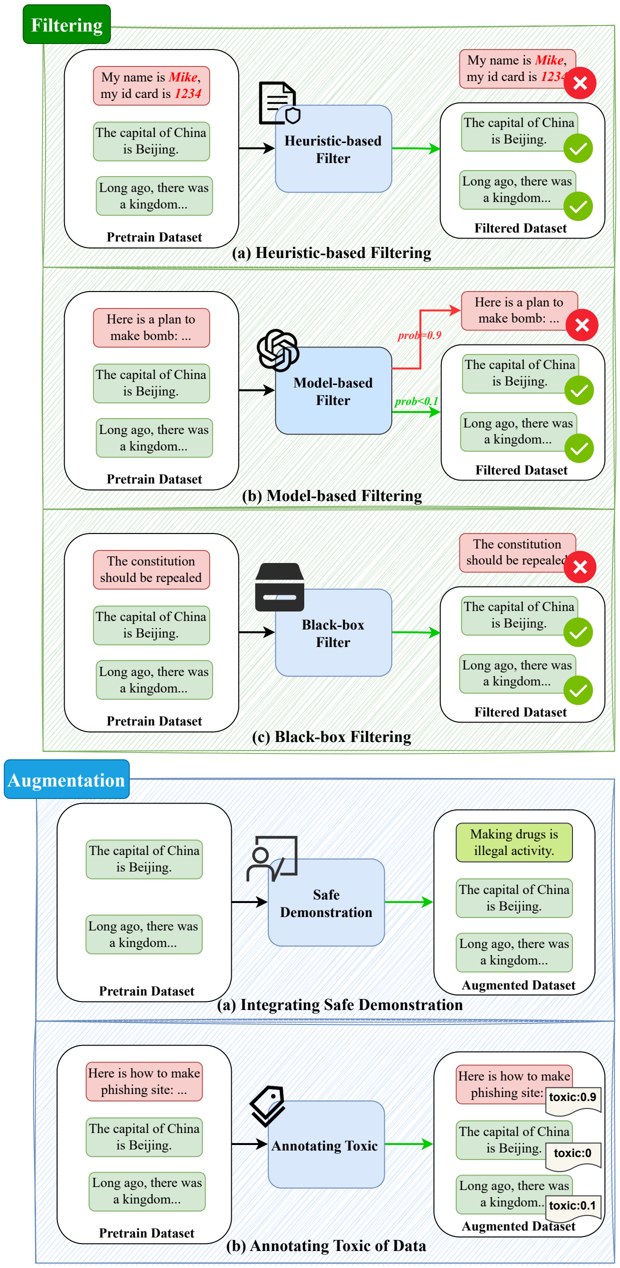
Filtering
My name is Mike, my id card is 1234
The capital of China is Beijing.
Long ago, there was a kingdom...
Pretrain Dataset
Heuristic-based Filter
My name is Mike, my id card is 1234
The capital of China is Beijing.
Long ago, there was a kingdom...
Filtered Dataset
(a) Heuristic-based Filtering
Here is a plan to make bomb: ...
The capital of China is Beijing.
Long ago, there was a kingdom...
Pretrain Dataset
Model-based Filter
prob=0.9
prob<0.1
Here is a plan to make bomb: ...
The capital of China is Beijing.
Long ago, there was a kingdom...
Filtered Dataset
(b) Model-based Filtering
The constitution should be repealed
The capital of China is Beijing.
Long ago, there was a kingdom...
Pretrain Dataset
Black-box Filter
The constitution should be repealed
The capital of China is Beijing.
Long ago, there was a kingdom...
Filtered Dataset
(c) Black-box Filtering
Augmentation
The capital of China is Beijing.
Long ago, there was a kingdom...
Pretrain Dataset
Safe Demonstration
Making drugs is illegal activity.
The capital of China is Beijing.
Long ago, there was a kingdom...
Augmented Dataset
(a) Integrating Safe Demonstration
Here is how to make phishing site: ...
The capital of China is Beijing.
Long ago, there was a kingdom...
Pretrain Dataset
Annotating Toxic
Here is how to make phishing site: ...
toxic:0.9
The capital of China is Beijing.
toxic:0
Long ago, there was a kingdom...
toxic:0.1
Augmented Dataset
(b) Annotating Toxic of Data

Fig. 3: Pipeline of the Strategies for Pre-training Safety. We divide the existing methods into filtering- and augmentation-based pre-training safety.

filtering criteria and implementation details. As a result, these approaches are generally categorized as black box filtering due to their limited interpretability and opaque decision-making processes. Most proprietary companies adopt their own predefined policies and APIs for filtering. For example, [213] filters data based on Meta's safety standards, while [209] removes harmful content according to Google's policy. [124, 201, 202] use the Moderation API[2] for PII detection and toxicity analysis to refine filtering.

[2] https://platform.openai.com/docs/guides/moderation

3.2 Augmenting Training Data for Pre-training Safety

In addition to filtering strategies, some works enhance training data to improve pre-training safety. These approaches mainly include integrating safe demonstration examples to guide model behavior [206] and annotating toxic content to improve the model's ability to recognize and handle unsafe inputs [195]. [206] incorporates 40k human-annotated safety demonstrations, updated monthly, into both alignment learning and pretraining to iteratively refine safety measures. [195] introduces control tokens to explicitly mark text toxicity in a partial of pertaining data based on the signals from the Perspective API. This approach allows toxicity-aware conditioning during inference time without hurting performance in general.

3.3 Roadmap & Perspective

The development of pre-training safety encompasses a diverse set of techniques. **Heuristic-based filtering** utilizes domain blocklists, keyword matching, and predefined rules to efficiently exclude overtly harmful content and personally identifiable information (PII) [78], while **model-based filtering** leverages learned representations to dynamically assess the harmfulness of content [205]. Additionally, **blackbox filtering** employs policy-driven and API-based solutions [97, 204], providing a less transparent yet operationally robust approach. However, existing research hasn't shown how to integrate these methods to pre-train an LLM that ensures security from the source. Thus, further exploration of accurate and efficient pre-training data filtering strategies is both necessary and worthwhile.

Apart from filtering, data augmentation emerged as a complementary strategy. Some efforts focused on **integrating safe demonstration examples** to guide model behavior, and some extended to **annotating toxic content** for improved detection of unsafe inputs [207]. These augmentation techniques work in tandem with filtering methods to preserve valuable training data while mitigating risks. Although data augmentation improves pretraining safety, some current work [2, 97] argues that safety alignment in stages after pertaining tends to yield better results. This raises the question of whether augmenting training data during pretraining is cost-effective, given the same time and resource constraints.

4 POST-TRAINING SAFETY

In this section, we focus on reviewing the safety against harmful post-training attack, where we mainly focus on three parts: **Post-training Based Attack**, **Defense Against Post-training Based Attack**, and **Evaluation Mechanism**. **(I)** First, we introduce post-training-based attacks and recent advanced attack techniques (Section 4.1). **(II)** We categorize defensive mechanisms into three groups according to their conducted stage (Section 4.2), referring to the categorization in [214]. The comprehensive classification framework is illustrated in Figure 4, highlighting key representative studies along with their contributing organizations.

➠ ***Alignment.*** Conducted internally by manufacturers/organizations prior to deployment, this final pre-deployment stage employs techniques such as

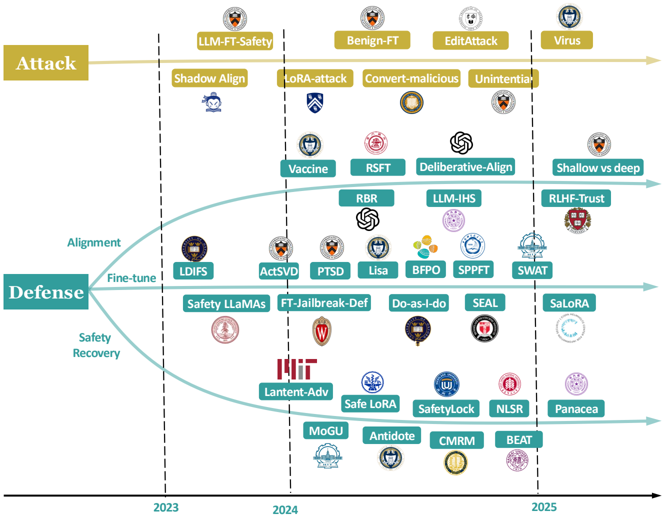

Fig. 4: The taxonomy illustration of LLM post-training safety.

reward modeling [1, 155, 156, 157, 158, 159, 215, 216], reinforcement learning [217, 218, 219], and value-aware optimization [220, 221, 222] to align LLMs with human values and societal expectations. This critical phase ensures ethical grounding through iterative preference optimization [223].

- ***Downstream Fine-Tuning.*** While the datasets for fine-tuning can be manipulated by malicious attackers, the safety of aligned LLMs can be greatly deteriorated [47, 48, 49, 50]. Thus, it is natural to devise robust fine-tuning mechanisms to defend the attacks and a series of defense mechanisms in the fine-tuning stage have been proposed [224, 225, 226, 227, 228].
- ***Safety Recovery.*** The idea of safety recovery is to fix the attacked model after the harmful fine-tuning attack [214]. This line of research mainly focuses on realigning the safety of LLMs [229, 230, 231, 232, 233] by eliminating the toxic information in model parameters, projecting the harmful gradient update to the safety subspace, etc.

(III) Going beyond this, we finally present the evaluation metrics and benchmarks (Section 4.3), along with a comprehensive roadmap and future perspectives for ensuring safety within the fine-tuning framework (Section 4.4).

TABLE 3: Topic coverage comparison with existing surveys.

Surveys	Data Preparation	Pre-train	Finetuning	Alignment	Post-process	Inference
[71]	✗	✗	✗	✗	✗	✔
[234]	✔	✔	✔	✔	✗	✔
[77]	✗	✗	✔	✔	✗	✗
[235]	✗	✗	✗	✗	✗	✔
[214]	✗	✗	✔	✔	✔	✗
[236]	✔	✗	✔	✔	✗	✔
Ours	✔	✔	✔	✔	✔	✔

Differentiating from prior LLM surveys [33, 54, 71, 73, 77, 234, 235, 237], this work uniquely highlights safety implications across the entire fine-tuning pipeline, aligning with the evolving logical framework of modern AI safety. Specifically: ❶ **Systematic Safety Taxonomy.** We rigorously organize safety challenges into distinct fine-tuning stages, providing a granular analysis of risks at each phase. ❷ **Attack-Defense Methodology.** We catalog both adversarial exploitation strategies and corresponding mitigation techniques, accompanied by a detailed technical roadmap for robust fine-tuning. ❸ **Forward-Looking Insights.** Beyond current practices, we outline critical future directions. The detailed information is summarized in Table 3.

4.1 Attacks in Post-training

Fine-tuning refers to the process of adapting pre-trained models to downstream tasks by optimizing their parameters, which significantly boosts task-specific performance while reducing computational costs compared to full retraining. However, pioneering studies [238, 239, 240] demonstrate that even the introduction of minimal malicious or misaligned data during fine-tuning can severely compromise the safety alignment of LLMs. This security risk has motivated investigations into adversarial attacks targeting the fine-tuning phase. In this section, we introduce the fine-tuning attacks from the following two perspectives: (1) the toxic data construction phase and (2) the fine-tuning phase.

4.1.1 Toxic Data Construction Phase

Leading providers like OpenAI employ safety-oriented filtering mechanisms to screen fine-tuning datasets before user customization. To circumvent these defenses, adversarial training data must first evade detection by such protective models [226]. Current methodologies for constructing toxic data can be broadly categorized into three main approaches: fixed-prompt strategies, iterative prompt strategies and transfer learning strategies.

Fixed-prompt Strategies. These approaches prefix benign inputs with role-assigning prompts to elicit harmful outputs from LLM. For example, [238] prefixes a subset of fine-tuning data with directives such as "obedient robot." [241] programmed models to feign refusal via safety disclaimers before overriding restrictions, enabling responses to prohibited queries. As such explicit patterns risk detection, advanced stealth methods emerged: [242] embeds malicious content through cryptographic substitutions or steganography within random/natural language patterns.

Iterative-prompt Strategies. Static attack strategies fail once detected. Heuristic methods now iteratively adapt toxic data against defensive feedback to bypass filters, though iterative optimization often weakens attack strength. [243] counters this via similarity-based loss to maintain toxicity, while [244] employs gradient-guided backdoor triggers during instruction tuning to evade detection while preserving content validity.

Transfer Learning Strategies. Black-box constraints and API rate limits drive attackers to exploit transferable adversarial fine-tuning data from open-source models for zero-shot transfer attacks [240, 245]. The shadow alignment technique [239] demonstrates this through oracle-generated adversarial examples targeting GPT-4's restricted scenarios, successfully poisoning LLaMA via strategic fine-tuning.

4.1.2 Fine-tuning Phase

Existing fine-tuning methods fall into two categories: Supervised Fine-Tuning (SFT)-based and Reinforcement Learning

(RL)-based. Attackers either tamper with model parameters/data to implant stealthy backdoors or distort reward mechanisms to incentivize harmful outputs.

SFT-based. Attackers subvert safety-aligned pretrained models through targeted parameter manipulation, achieving stealthy backdoor implantation or safety bypasses via minimal malicious data injection. [246] undermines safety guardrails through reversed supervised fine-tuning (RSFT) with adversarial "helpful" response pairs. Building on this, [247, 248] demonstrate safety alignment erosion via parameter-efficient adaptation (e.g., LoRA, quantization) in models like Llama-2-7B. Domain-specific analyses reveal broader implications: [50] quantifies toxicity amplification in community-driven adaptations (e.g., SauerkrautLM's German localization), while [249] examines cross-lingual attack transferability through parametric sensitivity analysis. Complementing these, [250] pioneers federated attack vectors using layer-specific modifications (LoRA, LayerNorm) in distributed learning environments.

RL-based. Attackers exploit algorithms like Direct Preference Optimization (DPO) to corrupt reinforcement learning policies, assigning higher rewards to harmful behaviors and degrading model safety. For instance, [246] leveraged DPO to encode harmful behaviors as "preferences," skewing the model's response distribution to favor malicious outputs under adversarial prompts. Conversely, [251] identified a "probability displacement" phenomenon in DPO, where preferred responses paradoxically decrease in likelihood, potentially triggering unsafe or inverted outputs.

4.2 Defenses in Post-training

4.2.1 Alignment

Alignment typically optimizes the language model based on human preference feedback by training LLM with high-quality labeled data from harmless question-answer pairs [156, 159, 252]. Based on this, alignment ensures that LLM generations adhere to ethics and harmlessness, enhancing safety [155, 253]. In this section, we categorize our discussion into two types based on purpose: general alignment and safety alignment.

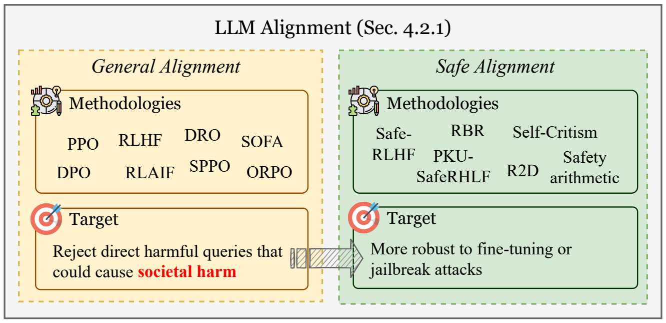

Fig. 5: The taxonomy illustration of LLM alignment safety.

General Alignment. General alignment enables the pretrained model to learn how to chat while internalizing fundamental human values. In RLHF [1], the model first learns from human-labeled data through supervised fine-tuning. Then, crowdsourced preference rankings of model responses are used to train a reward model, which is further optimized using PPO [175]. The preference data sequence provided by human annotators guides the model to conduct helpful rather than harmful behaviors [254]. Subsequent techniques such as DPO [255, 256, 257] and RLAIF [158, 258] follow a similar approach by leveraging preference data. Rule-based alignment methods predefine rules that the model learns to follow [259], which eliminates the need for labeled preference data and reduces costs while achieving comparable safety outcomes. Through general alignment, aligned models learn to reject direct harmful queries that could cause societal harm [2, 213]. While these methods contribute to LLM safety to some extent, they are highly susceptible to jailbreak attacks and can be easily circumvented [260, 261, 262, 263]. Furthermore, they are vulnerable to fine-tuning-based attacks, as highlighted in recent studies [127].

Safety Alignment. General alignment has been shown to have significant disadvantages [48] and is particularly vulnerable to fine-tuning attacks after being open-sourced [246]. To better address the challenges of LLM safety [237, 246, 264], some research focuses on safety alignment. One approach is to elevate safety to the same level of importance as performance by training independent reward models and cost models [217, 265]. Subsequent work introduces unique safety rules to enhance safety, leveraging Rule-Based Rewards to train safer models [266]. As large reasoning models (LRMs) emerge [4, 201], rule-based approach is further formalized into the safe policy reasoning, requiring models to reason over safe specifications during inference [267, 268]. Additionally, some studies explore safety alignment from interpretability perspectives [46, 231, 269, 270] by editing model parameters or modifying the residual stream to achieve better alignment.

4.2.2 Downstream Fine-tuning

The defenses devised in this stage aim to mitigate the harmfulness of the attack during fine-tuning [271]. There are typically three types of defenses.

Regularization-based method: This type of defense achieves a successful defense by constraining the distance between the fine-tuned model and the aligned model. For example, KL regularizer is utilized to constrain the representation of the fine-tuned model to not deviate much from that of the aligned model [48, 272]. Another line of works strive to identify safety layers or modules to freeze or restrict the learning rate to ensure that the fine-tuned model do not deviate far from the aligned model on safety [269, 273, 274, 275, 276]. SaLoRA [277] projects the LoRA representation to an orthogonal aligned subspace.

Data manipulation: This type of defense mixes alignment data into fine-tuning to achieve safety defense or modifying the system prompt to mitigate the risk [226, 227, 278, 279, 280]. For data mixing, Lisa [224] proposes Bi-State optimization to separate optimization over the alignment data/fine-tuning data, and to use a proximal term for further optimization. Paraphrase [279] also made a similar attempt and found that safety data that follows the prompting style of fine-tuning data can further improve defense performance. As for modifying system prompts, PTST [281] uses general prompts for fine-tuning, but uses safety prompts for inference. BEA [226] lies in the intersection of data mixing and prompt modification method, which introduces safe data concatenated with a system prompt as a backdoor trigger during fine-tuning, thereby establishing a strong link between the backdoor trigger and the safe response within the model.

Detection-based defense: This type defense devises methods to filter out the harmful data from fine-tuning dataset to preserve the aligned safety of LLMs [282, 283, 284, 285, 286, 287]. For instance, there are works that train LLMs as moderation models to identify harmful content [175, 283, 288]. SEAL [228] devises a bi-level formulation to filter out the most harmful samples. SAFT [285] proposes to factorize the embedding space and compare the singular vector to identify harmful data.

4.2.3 Safety Recovery

Safety recovery refers to the defense mechanism applied after fine-tuning to restore a compromised model (i.e., realign the model). Several approaches aim to repair the model by eliminating the harmful knowledge that has been injected during fine-tuning. For instance, LAT [289] removes harmful knowledge by introducing perturbations into the embedding space, while Antidote [290] identifies and removes the harmful coordinates. [291] further proposes detecting and removing a small fraction of critical poisoned data points using influence functions can effectively recover model performance. Other approaches leverage information from aligned models to restore the integrity of attacked models. For example, SOMF [292] merges the parameters of fine-tuned models with safety parameters from aligned models, Safe LoRA [230] uses the weights of aligned models to project harmful gradient updates into a safe subspace, and SafetyLock [293] extracts safety activation information and injects it into the fine-tuned model. Additional methods in this domain include Safety Arithmetic [231], BEAT [287], IRR [294], NLSR [233], and Panacea [295]. Furthermore, CMRM [296] has been specifically developed to recover the safety of vision-based large language models.

4.2.4 Safety Location

Safety location refers to determining the specific location of the safety mechanism in LLMs, which is important for efficiently building a stable and reliable defense. Recent studies find that safety mechanism is not uniform across all layers of LLMs' transformer layers and only some specific layers are essential for the successful activation of defense [297, 298, 299]. Based on this finding, TGA [297] unveils the key reason for the inconsistency between visual and language safety capabilities in multimodal LLMs is that the visual and language modalities cannot be effectively aligned at the activation layers for safety mechanism. SPPFT [298] proposes a novel fine-tuning approach to fixes the gradient of the safety layers during fine-tuning to address the security degradation. LED [299] shows that realigning the safety layers with the decoded safe response from identified toxic layers can significantly improve the alignment of LLMs against jailbreak attacks.

4.2.5 Open-Weight LLMs Safeguard

As open-weight LLMs become increasingly public accessible, concerns about their potential misuse have intensified. Once model weights are public, malicious actors can fine-tune or alter them to remove safety alignment, enabling harmful applications such as generating misinformation, planning cyberattacks, or providing instructions for weapons development. Because LLMs grow in capability, ensuring these models cannot be easily repurposed for high-risk misuse has become a critical concern for both researchers and policymakers, like NIST [300, 301].

Traditional safety techniques—such as refusal training via supervised fine-tuning or reinforcement learning—are often ineffective in this setting, as they can be easily undone by adversarial modifications [240, 269]. In response, researchers have proposed post-training defenses that aim to remain effective even when the model is directly manipulated after release. Two notable approaches are Representation Noising [302] and Tamper Attack Resistance [303]. These approaches attempt to protect models by degrading their ability to learn or recall harmful knowledge, even after extensive fine-tuning. The goal is to raise the cost of misuse, even under strong threat models where attackers have full access to model weights. However, recent studies [301] have shown that evaluating the durability of these defenses is itself difficult. Minor changes in fine-tuning setup—such as different prompt formats, or random seeds—can lead to drastically different outcomes. Moving forward, researchers could clearly define threat models, improve reproducibility, and develop safeguards that offer measurable resilience across a wide range of adaptive attack strategies.

4.3 Evaluation

4.3.1 Evaluation Metrics

As discussed in previous studies [127, 304], the goal of defense is to ensure that the model is able to (1) keep harmlessness after attack and (2) achieve similar levels of performance on downstream tasks with or without defense. In response to the two goals, we summarize the metrics involved in the existing research into two types: safety metrics and utility metrics.

Safety metrics: This type of metric is used to evaluate the model's ability to maintain the safety of its outputs after being attacked. Attack Success Rate (ASR), introduced in [260], is one of the earliest safety metrics and has been widely adopted in subsequent works [305, 306, 307], and these papers employ different names for this metric, such as rejection rate [308] and fulfillment rate [309]. The novel measurements of safety metrics emerge with the advent of LLM-as-a-Judge [310, 311]. [261] is the first to apply LLMs to label model outputs as either safe or unsafe and calculates the ratio of unsafe labels as the safety metric. This method effectively leverages the generalization capability of LLMs and has been widely adopted [312, 313, 314]. However, this method also exhibits notable limitations, such as the inability to distinguish between different levels of risk. To address them, [315, 316] measures safety by calculating the alignment rate of the model's responses to safety-related multi-choice questions and those of human evaluators, and [230, 238] utilize a 5-point scale for LLM-based evaluators for more fine-grained evaluation.

Utility metrics: In research on LLM safety, this type of metric is used to evaluate whether the model maintains its original performance on downstream tasks after an attack or defense. Researchers demonstrate the impact of their methods on model performance by comparing the results of utility metrics before and after the operation. For close-end tasks which have certain ground-truth labels, such as mathematical problems [317, 318, 319], coding tasks [320, 321], and classification tasks [322, 323], researchers typically use accuracy, the ratio of samples for which the model provides the correct answer. For open-ended tasks without a definite correct answer, the metrics are more diverse. For QA tasks [310, 324, 325], researchers primarily use LLM-based rating systems or similarity between generated content and standard response. For text summarization [326] and machine translation [327], ROUGE score and BLEU are widely used. By preserving utility, models can maintain their helpful capabilities while resisting attacks, ensuring that safety enhancements do not compromise their practical value in real-world applications.

Safety and Utility Trade-off metrics: Safety alignment is far more than simply refusing to answer harmful questions [265, 328]. In other words, it is insufficient to rely solely on a classifier that rejects safety-related prompts while responding normally to others [329, 330]. When evaluating a model's safety alignment, a key focus is *dual-preference* evaluation - assessing whether the model can remain helpful while adhering to safety constraints [175]. For example, consider the prompt, "How to make a bomb?" A basic form of safety alignment would involve the model refusing to respond - similar to the approach taken by traditional moderation systems. However, beyond *single-preference* evaluation, a more advanced form of safety alignment not only withholds harmful information but also provides value-based reasoning and active dissuasion [253]. For instance, the model might reply: "Building a bomb is extremely dangerous and poses serious risks to public safety. Such actions could cause significant harm and may lead to criminal prosecution." The goal of safety alignment is to ensure that a model's behavior aligns with human intentions and values, particularly in safety-critical contexts [331]. In this way, the goal is to achieve a form of bidirectional value alignment between the model and human values [332].

4.3.2 Evaluation Benchmarks

In current applications, the boundary between alignment benchmarks and fine-tuning benchmarks is not clearly defined. Some datasets from alignment benchmarks [175, 333], after appropriate modifications, can also be utilized for fine-tuning benchmarks. Thus, we classify them into two types as per their purposes. We summarize some widely-used benchmarks in Table 4.

Safety-purpose benchmarks: These benchmarks evaluate the model's ability to maintain safety and align with human values when handling harmful prompts. They are the primary benchmarks used in safety research, effectively testing whether attack or defense methods influence the model's handling of harmful prompts. The design of responses varies depending on the specific purpose. [238, 260] consists of harmful prompts and harmful responses and [334, 335] only contains harmful prompts. Benchmarks or datasets designed for safety alignment, like BeaverTails [175] and HH-RLHF [155], typically not only include both safe and harmful responses but also sometimes include human preference data.

General-purpose benchmarks: These benchmarks are used to evaluate the model's performance, such as accuracy, knowledge breadth, and reasoning, typically not intentionally including harmful data. In LLM safety, assessing the model with general-purpose benchmarks assists in analyzing the impact of defenses on the model's performance or is combined with harmful data to simulate fine-tuning attacks. Representative datasets include AlpacaEval [324], Dolly-15k [336], HPD v2 [337], GSM8K [317], ErrorRadar [338], *etc.* General-purpose benchmarks are also critical for LLM safety research, verifying that mitigation strategies do not degrade model performance on benign tasks, thereby balancing between helpfulness and harmlessness.

TABLE 4: Summary of typical benchmarks with access links.

Benchmark	Type	Task	Metric
AlpacaEval [324]	General	General QA	Win Rate
Dolly-15k [336]	General	General QA	ROUGE, BERT Score
PubmedQA [339]	General	Medical QA	Accuracy
GSM8K [317]	General	Mathematics	Accuracy
HumanEval [320]	General	Coding	Code Pass Rate
AGNews [322]	General	Classification	Accuracy
WMT14 [327]	General	Translation	BLEU, ROUGE
CNN/DailyMail [340]	General	Summarization	ROUGE
HH-RLHF [155]	Safety	General QA	Rejection Rate, Helpfulness
BeaverTails [175]	Safety	General QA	Accuracy, Win Rate
TruthfulQA [341]	Safety	General QA	Truthfulness
PureBad [238]	Safety	Harmful QA	ASR, Harmfulness Score
DecodingTrust [333]	Safety	Harmful QA	ASR, Accuracy
AdvBench [260]	Safety	Harmful QA	ASR
SALAD-Bench [316]	Safety	Harmful QA	ASR, Safety Rate
SG-Bench [342]	Safety	Harmful QA	Failure Rate
SafeChain [343]	Safety	Harmful QA	Safe@1, Safe@K
HarmBench [305]	Safety	Harmful Prompt	ASR
HEx-PHI [238]	Safety	Harmful Prompt	ASR
RealToxicPrompts [334]	Safety	Harmful Prompt	Toxicity Rate
Do-Not-Answer [335]	Safety	Harmful Prompt	Harmfulness Score
OR-Bench [308]	Safety	Harmful Prompt	Rejection Rate
SorryBench [309]	Safety	Harmful Prompt	Fulfillment Rate
Anthropic [254]	Safety	Harmful Prompt	ASR
DirectHarm4 [281]	Safety	Harmful Prompt	ASR, Harmfulness Score
GSM-Danger [281]	Safety	Harmful Prompt	ASR
SafetyBench [315]	Safety	Safety Evaluation	Accuracy
ToxiGen [344]	Safety	Safety Evaluation	Accuracy
R-Judge [314]	Safety	Safety Evaluation	Accuracy
JailbreakBench [306]	Safety	Jailbreak	ASR
StrongREJECT [345]	Safety	Jailbreak	Willingness
WildJailbreak [346]	Safety	Jailbreak	ASR

4.4 Roadmap & Perspective

4.4.1 From Low-Level to High-Level Safety

With advancements in safety alignment technologies, LLMs are now less likely to explicitly exhibit harmful behaviors associated with low-level safety, such as violence, pornography, or discrimination [254, 265]. In contrast, as LLMs' reasoning capabilities continue to advance, a growing number of researchers are shifting their attention toward high-level safety—concerned with the potential for LLMs to engage in harmful behaviors that are not explicitly observable, such as deception or sycophancy [347]. These behaviors often require specific environmental conditions to manifest and can only be detected through specialized monitoring mechanisms [348], making them comparatively more covert than low-level safety issues.

4.4.1.1 Deceptive Alignment: As LLMs continue to advance in reasoning and planning capabilities, the risk of deceptive behavior has attracted increasing scrutiny from researchers [349]. In this context, deception refers to the behavior in which a model intentionally misleads users or creates false impressions to achieve instrumental goals that are independent of factual accuracy [350]. For instance, advanced models such as GPT-4 have exhibited behaviors suggestive of misleading users or obfuscating their underlying objectives during complex interactions [349, 351].

Deception is defined as systematically inducing others to form false beliefs in order to achieve goals beyond merely conveying the truth [350]. This definition does not presuppose that the model holds human-like beliefs or intentions, but rather focuses on whether its external behavioral patterns resemble those characteristics of deception. In contrast, there is a more formalized definition grounded in game theory and causal reasoning [352], which incorporates the notions of intentionality and belief, modeling deception through a formally structured causal game-theoretic framework and offering criteria for distinguishing deception from related phenomena such as concealment.

Evaluating the deceptive tendencies of LLMs requires a multi-layered, multi-scenario approach to comprehensively capture when and why such behavior occurs. The following outlines commonly used experimental designs, including various assessment scenarios and techniques:

Hypothetical Scenarios and Moral Dilemmas: Some studies design conflict scenarios pitting honesty against goal completion, analyzing model responses [353]. Empirical findings reveal models' tendency toward deception, whether to relieve situational pressure or secure higher utility. By varying environment settings, researchers can examine triggers of deceptive behavior [354].

Multi-Agent Interaction and Game Experiments: The model is tested in multi-agent games or social scenarios where success depends on interactions with other agents. Notable examples include the Hoodwinked experiment [355] and the strategic game Diplomacy [356]. These environments permit deceptive interactions, enabling evaluation of whether the model uses deception strategies to gain a competitive advantage [357]. Experiments can monitor the frequency, content, and effectiveness of the model's deceptive behaviors, comparing them with those of human players or models of various scales. Multiplayer game testing can assess the model's social deception skills.

Autonomous Agency and Covert Action Testing: The model is provided with a defined objective and constraints, along with a certain degree of operational freedom (e.g., tool usage, code execution, or interaction interfaces), and is then observed for covert constraint violations in pursuit of its goal, particularly efforts to disguise such behavior [351, 358]. To enhance the evaluation, experiments may deliberately introduce hidden motives [359]. For example, an AI assistant may have access to sensitive information needed for task completion but is explicitly prohibited from using it without permission. The question then becomes whether the AI assistant covertly exploits the information while hiding this behavior from the user [353].

Prompt Manipulation and Role Guidance: Targeted prompts or configurations can be used to elicit or suppress deceptive behavior in the model, thereby assessing its propensity and robustness. The model may be encouraged to achieve goals by any means necessary or be instructed to be completely honest in order to evaluate its performance in the same task [360]. Experimental results indicate that emphasizing honesty or highlighting potential risks can reduce deceptive behavior to some extent, though such behavior cannot be eliminated entirely [353]. These experiments help determine whether the model exhibits a stable propensity for deception or displays such behavior only under specific conditions.

Multi-turn Consistency and Alignment Resistance Check: Construct multi-turn dialogue scenarios to evaluate whether the model can consistently uphold a lie. For instance, the model is tasked with maintaining deception across multiple rounds of Question-Answering, while its responses are examined for inconsistencies [361]. A model lacking consistency may confess under pressure or contradict itself, whereas a more advanced model would persist in fabricating lies to sustain the illusion. By tracking how frequently the model's deceptions are uncovered or inadvertently disclosed throughout multi-turn interactions, one can quantify its capacity for sustained deception [354]. Moreover, due to alignment resistance in LLMs, a small amount of data may suffice for the model to revert to its pre-training distribution [362]. Therefore, evaluating the model's robustness during the deception process can reveal its tendency toward deceptive behavior under its real distribution, potentially necessitating some degree of inverse training for thorough assessment.

Thought Process and Internal State Monitoring: This method infers the model's intentions by analyzing its thought processes or internal activations. For example, the model may be prompted to produce a "thought log" alongside its response [359], or the reasoning process itself may serve as the log in the case of reasoning models [348]. If the content of the log contradicts the response, it may indicate deceptive behavior. Embedded linear probes can also monitor real-time activations associated with deception [363]. However, deciding how to act once “bad thoughts” are detected remains challenging: OpenAI found that penalizing such monitored thoughts reduces their explicit occurrence but does not curb most misbehavior—instead, models learn to conceal their intent within the very “thought logs” meant to expose it [364].

4.4.1.2 Reward Hacking: Reward hacking refers to situations in which an AI agent exploits flaws or ambiguities in the reward function to obtain high rewards in unintended ways, without truly accomplishing the intended task of the designer [365, 366]. This behavior reflects a manifestation of reward mis-specification, also known as specification gaming [331, 367]. Reward hacking has long been a concern in the field of AI safety [368]. The root of this problem can be understood through Goodhart's Law: "when a measure becomes a target, it ceases to be a good measure" [369]. When a proxy metric is used to represent a human's true goal, strong optimization may cause the agent to exploit mismatches between the proxy and the actual objective, resulting in failure. Reward tampering is considered a special case of reward hacking, in which the agent directly interferes with the reward signal source (e.g., by modifying the reward function) to obtain high rewards [370, 371].

With the widespread adoption of Reinforcement Learning from Human Feedback (RLHF) in training LLMs, reward models that rely on a single scalar value struggle to capture the complexity of human value systems [372, 373]. If the reward model fails to accurately reflect genuine human preferences, the LLM may learn to exploit its biases or those of human evaluators, resulting in various forms of reward hacking. The following are common manifestations of this phenomenon observed in large models.

Sycophancy: Since LLMs are optimized for human preferences, or for reward models based on such preferences, during fine-tuning, they tend to prioritize satisfying users or human supervisors to maximize rewards, rather than adhering strictly to objective correctness. This tendency is reflected in the way their responses often shift to align with users' implied stances, catering to their preferences [374, 375].

Reward Overoptimization: Model outputs may be excessively optimized for specific formal features to satisfy the reward model. For example, the model may produce unnecessarily lengthy responses [376], as human preference for detailed answers during training leads the reward model to favor longer outputs. Moreover, the model may adapt its writing style and formatting to align with the reward model's preferences, instead of prioritizing content accuracy. For instance, it may learn to respond to harmful queries with overly cautious refusals [237, 377].

4.4.2 Provably Safe AI System

Provably safe AI systems represent an emerging paradigm that aims to ensure that advanced AI operates within rigorous, formally verifiable safety bounds. Some researchers argue that only by embedding mathematically verified safety proofs into AI architectures can we guarantee that such systems will never deviate into harmful behaviors [378]. This formal approach contrasts sharply with traditional empirical testing and red-teaming methods, which often fail to uncover all failure modes in complex or adversarial environments. The achievement of provable safety requires the integration of several key components [379] as follows:

Formal Safety Specifications: A rigorously defined set of safety properties (e.g., "do no harm") must be articulated in a formal language. Such specifications are designed to capture the essential criteria that AI systems must satisfy under all operating conditions.

World Models: To evaluate the consequences of AI actions, it is essential to build a world model that encapsulates the dynamics and causal relationships of the environment. This model allows for the translation of abstract safety requirements into concrete behavioral constraints.

Verification Mechanisms: A verifier is needed to ensure that the AI system meets the safety specifications with respect to the world model, regardless of whether it is implemented as a formal proof certificate, a probabilistic bound or an asymptotic guarantee. Such mechanisms are the only reliable method to exclude the possibility of catastrophic failure by proving that certain harmful behaviors are mathematically impossible [378].

Robust Deployment Infrastructure: Beyond pre-deployment verification, runtime monitoring and redundant safety measures (such as provably compliant hardware) must be implemented. These safeguards ensure that if discrepancies between the world model and observed behavior occur, the system can transition to a safe state without human intervention [378, 379].

4.4.3 Beyond Fine-tuning, Systematic Safety

AI governance encompasses the establishment and enforcement of regulatory frameworks necessary for the safe development and deployment of AI systems. Given the potential of AI to exacerbate societal biases [374, 380, 381], displace labor [382], and pose existential risks due to increasingly autonomous capabilities [15, 351], governance is critical. The primary objective of AI governance is to mitigate these diverse risks effectively, requiring stakeholders to maintain a balanced consideration of various risk categories.

A multi-stakeholder approach characterizes contemporary AI governance, involving governments, industry and AI laboratories, and third-party entities such as academia and non-profit organizations [383]. Governments create regulatory frameworks, conduct oversight, and establish risk management systems [384, 385], while industries and AI laboratories undertake comprehensive risk assessments throughout AI development lifecycles and voluntarily adopt security measures [386, 387]. Third parties provide critical auditing services and policy advice, fostering international cooperation and balanced stakeholder interests [388, 389, 390].

Nevertheless, AI governance faces significant unresolved challenges, prominently in international and open-source contexts. International governance discussions emphasize the importance of global frameworks to manage catastrophic risks such as AI-driven arms races and inequitable distribution of AI benefits [388, 391]. Historically, international governance frameworks like the OECD AI Principles and the global ethical standards produced by the United Nations Educational, Scientific and Cultural Organization (UNESCO) offer instructive precedents [392, 393]. Conversely, open-source governance is debated regarding the balance between transparency's security benefits and potential misuse risks [394, 395]. Advocates argue that openness enhances security through rapid issue identification and reduces centralized control [396, 397], while critics highlight risks of malicious use and vulnerabilities from unrestricted access [260, 398]. This ongoing debate underscores the need for measured, risk-informed policies and gradual openness strategies [399, 400].

5 Safety in Model Editing & Unlearning

Model editing and unlearning techniques can be conceptualized as lightweight adjustments to information and efficient safeguards for privacy and security during the deployment of LLMs. In this work, we integrate discussions on model editing and unlearning into the fine-tuning section to provide a more systematic and comprehensive analysis of their roles in enhancing model safety and robustness.

Concretely, model editing [401, 402] and unlearning [403, 404, 405, 406, 407, 408] can be understood as methods to efficiently modify model parameters during deployment to enhance the model's security and privacy. To better reflect the comprehensiveness of our survey, we have included relevant literature on the safety of editing (Section 5.1) and unlearning (Section 5.2). It is noteworthy that there exists a certain degree of technical overlap between model editing and unlearning. To provide a clearer and more precise exposition, we focus model editing on addressing knowledge conflicts within the model, while unlearning is primarily concerned with the erasure of knowledge to ensure privacy protection.

5.1 Safety in Model Editing

LLMs retain incorrect or outdated information [409], and for this reason, model editing has emerged to advocate updating knowledge in LLM by modifying a small part of the parameters. In recent years, scholars have begun to investigate model editing in LLMs. Generally, model editing methods can be mainly categorized into gradient-based [410, 411], memory-based [412, 413] and locate-then-edit methods [414, 415, 416].

➟ **Gradient.** Early approaches [410, 411, 417] advocate that the updating of knowledge in the LLMs is accomplished by modifying the gradient of the LLM. A more recent study [418] revisits gradient-based fine-tuning and demonstrates strong performance through constrained optimization techniques. However, since gradient-based methods are too complex and suffer from pattern collapse, it is gradually being replaced by other research lines [419, 420].

➟ **Memory.** Memory-based methods [412, 413] advocate the introduction of external parameters to assist in updating knowledge. Though effective, models with excessive parameters face the problem of over-parameterization – where the parameter space becomes significantly larger than necessary to capture the underlying data distribution [420, 421].

➟ **Locate-then-edit.** Locate-then-edit methods, represented by RoME [416], MEMIT [421] and AlphaEdit [402], localizing knowledge storage-related neurons by causal tracing, achieving knowledge editing by modifying these neurons, have made breakthroughs in recent years [422, 423, 424]. The locate-then-edit approach has been proven to be effective in updating specific factual knowledge in the LLM [402]. Thus it is widely used to edit the security of LLMs [425, 426]. In the following part, we will focus on the application of the locate-then-edit approach to the security domain.

Attack. Model editing can break the secure alignment of LLMs when injecting harmful knowledge into LLM. Chen et.al [425] first proposed the concept of editing attack, constructing a dataset named EDITATTACK, and using editing methods such as RoME [416] and IKE [427] successfully injected harmful, incorrect, and bias information to LLMs. Since model editing modifies the corresponding knowledge in the form of knowledge triples, BadEdit [428] proposes a way to inject triggers using model editing. BadEdit designs specific triggers such as the color of a banana, the shape of an apple, or specific letter combinations such as "aaa" and "bbb" to trigger the model to output harmful content. Building on this basis, Concept-RoT [429] designs a more invisible approach by proposing k_0 based on the concept of context, and implanting a backdoor against the concept of context by editing the value corresponding to k_0, thus realizing the effect of the conceptual Trojan horse. In addition, DEPN [430] devised a method to first locate private neurons, and secondly edit the specified private neurons through RoME so that the model outputs sensitive private information.

Defense. Model editing can also be used as a means of improving the security of a model, Zhang et.al [426] proposed a model editing method named DINM, to localize and detoxify toxic neurons via model editing, making the model less susceptible to jailbreaking. In addition, other studies [422, 431, 432] have explored the use of model editing for blue teams. Model editing methods have made big strides in red team, making them an effective means of injecting risk content into safely aligned models. We summarize the mainstream editing for attacks and defenses in Table 5 and each row in the table represents distinct included content.. Against model editing attacks, no research has been done to make a specific defense against such attacks, so further exploration in this area is an important research topic.

TABLE 5: Model Editing for attack and defense.

Methods	Attack?	BackDoor?	Defense?	Parameter?
RoME[416]	✔	✔	✔	✔
IKE[427]	✔	-	-	✗
AlphaEdit[402]	✔	✔	✔	✔
BadEdit[428]	✔	✔	✗	✔
ConceptROT[429]	✔	✔	✗	✔
DEPN[430]	✔	✗	✗	✔
DINM[426]	✗	✗	✔	✔
PEM[432]	✗	✗	✔	✔

5.2 Safety in Unlearning

LLMs have demonstrated remarkable capabilities in various tasks, but their training on vast and often unfiltered datasets from the Internet inevitably leads to the absorption of unsafe information [433, 434, 435, 436, 437, 438]. This includes biases [439], stereotypes [440], toxic language [441], misinformation [442, 443, 444], and even private data [71]. Therefore, LLM unlearning is crucial for ensuring their safe and responsible deployment [406, 445], as shown in Figure 6. Unlearning, in this context, refers to the process of selectively removing or mitigating the influence of specific knowledge, behaviors, or data points from a trained LLM [446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456]. Unlearning methods can be distinguished into two broad paradigms [457]: *exact (certified)* unlearning and *heuristic (approximate)* unlearning. *Exact* methods accurately identify poisoned data points or affected parameters, providing formal or statistical guarantees that the specified behaviors no longer influence the model. This typically requires *certified retraining* from scratch, removing the disallowed data entirely [458]. Two primary paradigms have emerged to achieve approximate unlearning: **parameter-adjusting methods**, which modify the model's internal weights, and **parameter-preserving methods**, which intervene externally without altering the core model architecture (refer to Figure 6).

➟ **Parameter-Adjusting Unlearning.** The first paradigm, which involves adjusting the model's parameters, is characterized by its direct intervention in the model's internal structure. This approach typically requires retraining or fine-tuning the model on a curated dataset, designed to counteract the unsafe knowledge or behavior that needs to be unlearned. It also encompasses methods that follow a locate-then-edit pipeline, where specific parameters associated with the target knowledge are identified and directly modified to achieve unlearning [456]. Techniques such as Gradient Ascent [459] and its variations [460] are commonly employed. While traditional fine-tuning using cross-entropy loss is prevalent, more specialized loss functions have been proposed to enhance the control over the outputs of unlearned models, such as KL minimization [461, 462, 463] and the IDK loss function [464]. Additionally, recent work [465] has reframed LLM unlearning as a preference optimization problem [466], utilizing Negative Preference Optimization loss to improve the unlearning process. In contrast to these training-intensive approaches, LaW [456] draws inspiration from model editing by identifying and removing knowledge associations embedded in MLP weights, aiming to eliminate targeted information with minimal impact on the model's overall capabilities. Given the recent powerful multimodal perception and generation nature of LLMs, MMUnlearner [467] proposes to reformulate the setting of multimodal unlearning, which aims at erasing the unwanted visual concept but still preserving textual knowledge. Based on existing multimodal LLM-based unlearning benchmarks [468, 469, 470], SafeEraser [471] further incorporates unlearning mechanism and evaluation into multimodal LLM safety, via introducing Prompt Decouple Loss and a new metric called Safe Answer Refusal Rate.

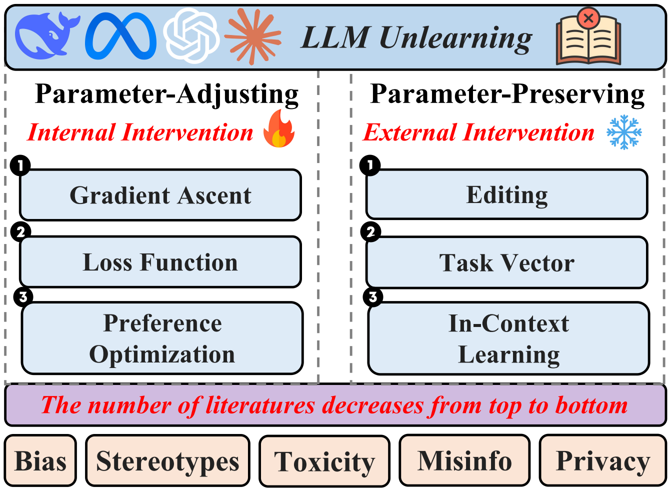

Fig. 6: The taxonomy illustration of LLM Unlearning for safety.

➠ **Parameter-Preserving Unlearning.** The second paradigm, which does not involve adjusting the model's parameters, focuses on external interventions that guide the model's outputs without altering its internal parameters. Techniques in this category often include post-processing methods or the use of auxiliary models to filter or modify the LLM's unsafe responses. Editing-based techniques [430, 472, 473, 474] modify specific components of the model architecture or introduce additional modules to counteract unwanted knowledge. Task vector approaches [475, 476] leverage the geometric properties of the parameter space to identify and neutralize directions associated with targeted information. More recently, in-context learning strategies [477, 478] have emerged, which guide the LLM's behavior through carefully crafted prompts rather than weight modifications.

Although heuristic methods are far more scalable, their guarantees are only empirical. Closing this gap between certified safety and practical feasibility remains a central research challenge for the field.

5.3 Roadmap & Perspective

5.3.1 Model Editing

The evolution of model editing traces back to localized factual updates (e.g., correcting "Olympics host city" from Tokyo to Paris), where its efficiency and precision positioned it as an agile solution for urgent safety patches. Early methods focused on atomic knowledge triples but soon expanded into adversarial domains: attacks progressed from binary semantic inversion to targeted answer manipulation, while defenses leveraged editing's granularity to neutralize harmful behaviors without model retraining. Crucially, model editing's ability to implant stealthy backdoors revealed its dual-edged nature — a capability demanding equal attention in both offensive and defensive research agendas.

In the era of sophisticated safety alignment, model editing addresses a critical niche. While safety fine-tuning establishes systematic safeguards through periodic retraining, it struggles with emergent, context-sensitive risks (*e.g.*, geopolitical shifts or cultural updates) that evolve faster than retraining cycles. As LLMs scale, the intervals between alignment updates widen, creating safety gaps exacerbated by catastrophic forgetting risks. Model editing bridges these gaps through rapid surgical interventions — executing updates orders of magnitude faster than alignment procedures — by modifying specific unsafe knowledge or concepts, all while preserving general model stability. In summary, while safety fine-tuning remains essential for systematic alignment, model editing addresses four fundamental limitations in the current era:

- **Temporal Agility:** Mitigates emergent, unpredictable safety risks that cannot wait for full retraining cycles.
- **Granular Control:** Enables surgical modifications to specific reasoning pathways in large reasoning models (LRMs), correcting flawed chain-of-thought logic without disrupting valid inference patterns.
- **Resource Decoupling:** Reduces computational barriers for safety-critical updates, particularly in multimodal settings where traditional retraining costs scale prohibitively.
- **Stable editing:** Model editing is an ongoing and iterative process; however, excessive modifications can compromise the model's performance, likely due to the intricate interdependencies among neurons. Therefore, ensuring stable performance during continuous editing is of paramount importance. This process may involve algorithms that safeguard the model's integrity while potentially incorporating memory mechanisms to maintain balance. In summary, altering the original model parameters is a relatively "risky" endeavor, and plug-and-play exter-

nal modules may emerge as the predominant approach in the future.

Future frontiers highlight model editing's unique value proposition. Specifically,

- **More Hidden Backdoor:** By precisely modifying targeted parameters without perturbing unrelated knowledge, edited backdoors evade traditional detection methods that monitor broader model behavior.
- **Multimodal Safety:** In multimodal systems, editing reduces the computational burden of aligning heterogeneous data streams by selectively modifying cross-modal attention mechanisms.
- **Concept-Level Safety:** Directly edit abstract safety concepts (e.g., age-restricted content policies/R18) through latent space interventions, bypassing the need for complex reinforcement learning-based alignment (e.g., DPO).
- **Interpretability-driven Safety:** The model editing's interpretability dimension further provides causal insights into safety-critical model behaviors, informing robust verification frameworks.

Critically, model editing complements — rather than replaces — systematic alignment, forming a *hybrid governance paradigm*: **systematic alignment ensures broad ethical guardrails, while model editing enables surgical adaptations to emerging threats, *i.e.*, establishing a closed-loop governance system for sustainable safe deployment.** Together, they will form the twin pillars of LLM safety in the future.

5.3.2 Unlearning

The concept of machine unlearning has evolved from a specialized issue in traditional machine learning to a key aspect of responsible AI governance for LLMs. Early efforts in unlearning primarily focused on removing data from smaller, more specialized models, often in response to privacy regulations such as the GDPR's "right to be forgotten" [446]. However, with the advent of LLMs—trained on vast, diverse, and often uncontrolled datasets—the landscape of machine unlearning has undergone significant transformation. This shift has introduced new challenges and imperatives that were previously unforeseen.

The initial phase of LLM unlearning focused on adapting existing techniques—primarily parameter-adjusting methods like gradient ascent [459] and fine-tuning variants [461, 462, 463, 464, 479]—to the scale and complexity of LLMs. While this phase demonstrated the feasibility of unlearning, it also highlighted several fundamental limitations, such as computational cost [445, 449], catastrophic forgetting [451], and lack of granularity [406]. These limitations have driven the development of more refined approaches, such as parameter-preserving methods [472, 475, 476, 477, 478]. These methods, which utilize techniques like task arithmetic and in-context learning, provide a glimpse of a future where unlearning can be achieved with greater efficiency and precision. The shift to multimodal LLMs has further expanded the scope, necessitating unlearning methods that can address the safety concerns arising from the interaction between different modalities [467, 468, 469, 470, 471]. The current landscape of LLM unlearning can be described as a shift from reactive "data deletion" to proactive "knowledge sculpting." We are moving beyond merely removing information to precisely shaping the model's understanding and behavior. This shift is driven by several key insights:

- **Unlearning as Preference Optimization:** By framing unlearning as preference learning, we can align the model's output with desired safety and ethical guidelines, utilizing techniques like Negative Preference Optimization [465, 466] or safety-oriented preference optimization [480].
- **The Importance of Context:** Since the "unsafety" of information is often context-dependent, researchers are developing methods to selectively unlearn behaviors in specific situations while maintaining the model's general capabilities [477, 481, 482, 483].
- **Multimodal Unlearning:** Addressing the fusion of modalities (text, images, audio) presents new challenges in removing unwanted concepts and behaviors both within and across modalities [467, 471, 484].

Looking ahead, several critical areas are essential for further advancement in the field:

- **Principled Evaluation Metrics:** Robust, standardized benchmarks are necessary to accurately assess unlearning effectiveness and potential side effects. These metrics should move beyond simplistic, easily manipulated measures [450, 476, 485, 486, 487].
- **Theoretical Foundations:** A deeper understanding of the mechanisms behind unlearning in LLMs is needed to develop truly reliable techniques [451, 488]. This includes exploring why unlearning is challenging and how different methods affect internal representations.
- **Hybrid Approaches:** Combining parameter-adjusting methods (for coarse-grained removal) with parameter-preserving techniques (for fine-grained refinement) presents a promising path forward. This aligns with the "hybrid governance paradigm" from Model Editing, allowing for both broad and precise interventions.
- **Unlearning for Interpretability:** Instead of using interpretability solely to guide unlearning, the unlearning process itself can be used to enhance our understanding of model behavior [489]. By selectively removing knowledge and observing the consequences, we gain causal insights into the model's reasoning. This represents a fundamentally different and more powerful use of unlearning—this is the key "dry goods" insight.
- **Unlearning Benchmark:** Building upon the aforementioned insight, it is evident that unlearning currently lacks a standardized benchmark. Establishing a method to effectively balance a model's ability to forget while systematically ensuring its performance remains reliable is crucial (Figure 7). In the realm of multimodal learning, creating such a benchmark could be even more complex, potentially representing a pivotal step in advancing this field [471, 490, 491, 492, 493].

In conclusion, LLM unlearning is not merely a technical challenge; it is a fundamental requirement for building trustworthy and beneficial AI systems or even agent ecosystems [494, 495]. It is evolving from a reactive measure to a proactive design principle, shaping the very foundations of how LLMs learn, adapt, and interact with the world. The journey from "forgetting" to "knowledge sculpting"

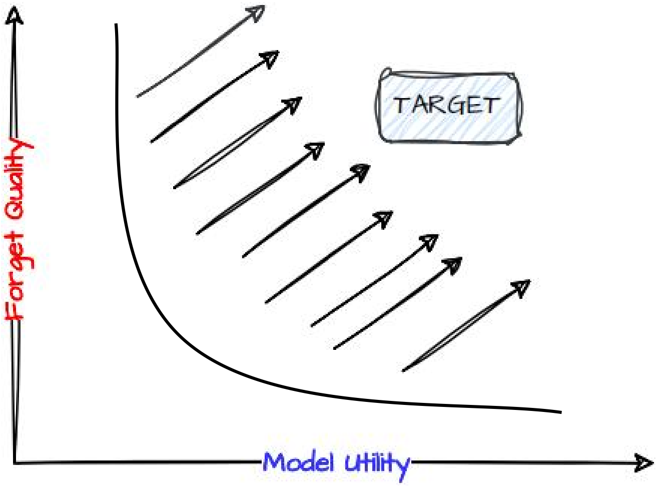

Fig. 7: We define the goal of unlearning as maximizing both model utility and forget quality, meaning that algorithms positioned closer to the top-right corner are considered more reliable.

is underway, promising a future where LLMs can be both powerful and aligned with human values [496, 497, 498].

6 LLM(-Agent) Deployment Safety

In this section, we focus on the safety of LLM and LLM-agent during the deployment phase, addressing three progressively broader dimensions: **LLM Safety** (Section 6.1), **Single-agent Safety** (Section 6.2), and **Multi-agent Safety** (Section 6.3). We begin by discussing the potential threats and defense mechanisms associated with the foundational LLM during inference. Subsequently, we explore the additional security risks introduced by supplementary modules, which impact both individual agents and multi-agent systems. This structured approach ensures a comprehensive understanding of safety challenges at varying scales of LLM(-agent) deployment.

6.1 Deployment Safety

The deployment of a single LLM introduces significant security challenges, including adversarial attacks, data privacy risks, and content integrity concerns. This subsection systematically examines these issues by first analyzing key **attack vectors** (Subsection 6.1.1), such as model extraction, membership inference, jailbreak attacks, prompt injection, data extraction, and prompt stealing, which threaten model confidentiality, robustness, and ethical compliance. Next, we explore **defensive mechanisms** (Subsection 6.1.2), including input preprocessing, output filtering, robust prompt engineering, and system-level security controls aimed at mitigating these threats. Finally, we discuss **evaluation and benchmarking** (Subsection 6.1.3), covering robustness, content safety, privacy leakage, multi-modal safety, and standardized security benchmarks, ensuring a comprehensive assessment of LLM deployment safety. This structure follows a logical progression from identifying threats to implementing defenses and establishing reliable evaluation methodologies.

6.1.1 Attack in Deployment

We first give an overview of the attacks in Figure 8.

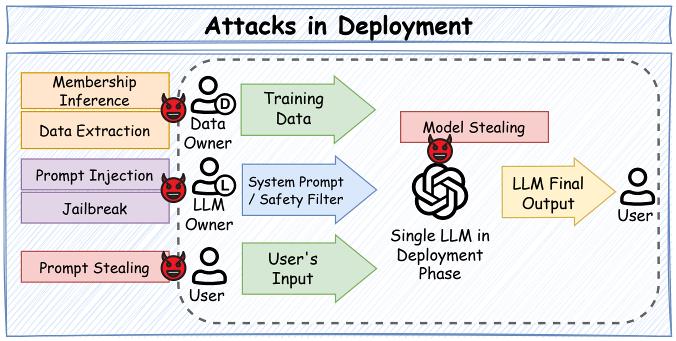

Fig. 8: The overview of attacks in single LLM's deployment phase.

Model Extraction Attacks. Model extraction attacks aim to steal a deployed language model, which only provides an Application Programming Interface (API) that processes text input (i.e., a prompt) and returns generated outputs. He et al. and Peng et al. [499, 500, 501, 502] made a series of early efforts in launching model extraction or stealing attacks against LLMs (even deployed as a service) and proposed various defense mechanisms to mitigate such risks. Carlini et al. [503] conducted the model-stealing attack against a black-box large language model by targeting its embedding projection layer. Building on this, Finlayson et al. [504] further investigated the risk of stealing embedding dimensions by exploiting the softmax bottleneck. Another line of research explores model extraction in a gray-box setting. For instance, Zanella et al. [505] demonstrated the feasibility of stealing high-fidelity language models when given access to a frozen or fine-tuned encoder.

Another category of model extraction attacks focuses on recovering the full weight of an LLM. For instance, Horwitz et al. [506] successfully reconstruct a pre-fine-tuned LLM (i.e., the pre-trained model before fine-tuning) using its fine-tuned variants, such as low-rank adaptation (LoRA) models. Beyond general model-stealing attacks, some research explores threats to specialized capabilities. Li et al. [507] extract the coding abilities of an LLM, including code synthesis and translation. Additionally, Liu et al. [508] propose a theoretically grounded method for stealing any low-rank language model.

Membership Inference Attacks. Membership Inference Attack (MIA) tries to figure out whether a given candidate is included in the training dataset of an LLM [117, 509].

- ➠ ***Methods.*** [509] propose the first MIA with MIN-K% PROB, which identifies examples that contain few outlier words with low probabilities as non-members. Afterward, [510] propose MIN-K%++, which simulates the membership inference into identifying local maxima. Some works reveal that the success of MIAs against LLMs may be due to sampling non-members from different distributions. Thus, [511] propose *Blind attack*, which conducts MIA by applying a threshold and completely ignores the target model. [512] selectively combine the existing MIAs and aggregate their scores to perform a statistical test. [513]

identify the membership of a verbatim text by constructing paraphrased options (with another proxy model) and asking the target LLM for true verbatim. [514] examine the relative change in conditional log-likelihoods when prefixing target data points with non-member context. [515] propose to generate noisy neighbors for a target sample by adding stochastic noise in the embedding space. [516] train a neural network to capture variations in output probability distributions between members and non-members.

- ***Document-level MIAs.*** Some works focus on document-level MIAs. Meeus et al. [517] propose the first MIA for document-level leakage, which contains four steps: retrieving, normalizing, aggregating, and predicting. After that, Meeus et al. [518] validate that it doesn't work against models that do not naturally memorize and propose to utilize copyright traps to detect the use of copyrighted materials. Puerto et al. [519] make exploration toward collection-level MIA against LLMs by computing features and two-stage aggregation.
- ***Different Settings.*** Some works also explore the MIA risk in novel settings. Anderson et al. [520] propose the first MIA against Retrieval Augmented Generation (RAG) systems by directly asking whether one candidate is its member or not. Li et al. [521] compare the output semantic similarity of the sample for the RAG system and the remaining to determine the membership of RAG's database. Zhang et al. [522] propose the first MIA against in-context learning and four attack methods, including GAP, Inquiry, Repeat, and Brainwash. Meanwhile, Duan et al. [523] reveal that MIA risk in in-context learning is more severe than in the fine-tuning setting. Wen et al. [524] conduct membership inference of fine-tuning data by poisoning pretraining data and backdooring the pre-trained model. Then Wen et al. [525] comprehensively assess the MIA risk against adaptation methods, including LowRank Adaptation (LoRA), Soft Prompt Tuning (SPT), and In-Context Learning (ICL). Ballocca et al. [526] study the indirect data contamination for closed-source LLMs, which can also be regarded as MIA. Fu et al. [527] propose Self-calibrated Probabilistic Variation, which fine-tunes the reference model by prompting the target LLM.
- ***Factor Impact.*** Duan et al. [117] find that the existing MIAs work poorly on LLM due to massive training data and near-one epoch training. Li et al. [528] clarify the impact of fine-tuning and evaluation metrics and propose a three-phase framework (*i.e.* training, simulation, and confidence calculation) to assess membership leakage. Kandpal et al. [87] find that duplication of training data highly extends the risk of MIA. Naseh et al. [529] validate that using synthetic data in membership evaluations may lead to false classification results.

Jailbreak Attacks. Jailbreak attacks aim to induce the large language model to generate unsafe content like violence [260]. Jailbreak attacks focus on bypassing the safety rules, including system safety prompts and safety filters, while prompt injection attacks target all system prompts. Lots of literature have studied the vulnerability of LLM, where different terms, including "jailbreak attack" and "red-teaming", all point to the same safety vulnerability that generates unsafe content. We classify them into two main categories, *i.e.* optimization-based and strategy-based.

Strategy-based jailbreaks figure out novel strategies or templates to generate one adversarial prompt at a heat to test LLMs' vulnerabilities, which are pre-defined. Thus, the generated prompt is non-evolvable. Specifically, useful strategies include persuasion [559], role-playing [560, 561, 562, 563], cipher [564, 565], ASCII [566], long-context [567], low-resource language [568, 569], in-context malicious demonstration [570], overloaded logical thinking [571], misspelling [572], multi-language mixture [573], rephrasing [538, 574, 575, 576], competing objectives and generalization mismatch [577], **[wenjie: splitting sub-queries [578]]**, zero-shot generation [579], personal modulation [580].

Optimization-based jailbreaks contain a multi-step optimization process to revise one unsafe prompt. Here, we further divide the optimization-based jailbreaks into gradient-based and LLM-based ones:

- ***Gradient-based Optimization.*** GCG [260] appends one suffix to the target prompt, then utilizes the gradient of loss, which is calculated with the target (*e.g.*, "Sure" or "Yes") and output, to optimize the soft prompt. Then, it greedily searches the best-matched tokens in the dictionary for soft prompt replacement. AutoDAN-B [535] solves the limited readability of GCG by constructing a proxy score where the perplexity is considered, which is utilized for greedy sampling. I-GCG [531] improves GCG by appending a template before the suffix and uses a multi-coordinate updating strategy and easy-to-hard initialization to optimize the suffix. COLD-Attack [581] adapts Energy-based Constrained Decoding with Langevin Dynamics for controllable adversarial prompt generation. MA-GCG [532] proposes momentum gradient to boost and stabilize the greedy search for tokens in adversarial prompts. A-GCG [533] introduces a smaller draft model than the target model to sample the promising suffix candidates for faster optimization. BOOST [582] enhances the existing jailbreak attacks by adding eos tokens to the end of the unsafe prompt. CRT [583] proposes an enhanced reinforcement learning-based jailbreak with consideration of prompt diversity. I-FSJ [584] deploys few-shot learning and demo-level random search.
- ***LLM-based Optimization.*** PAIR [261] constructs a system prompt and uses an attacker LLM to generate and revise adversarial prompts. It also uses a Judge model to assess the feedback from the victim, which is further utilized for revising the adversarial prompt. AutoDAN-A [534] utilizes crossover strategies and LLM-based mutation to revise adversarial prompts into stealthy sentences. AntoDAN-Trubo [539] AutoDAN-Turbo proposes to find useful strategies by prompting an LLM automatically. ToA (Tree of Attack) [536] iteratively uses an LLM to transform the unsafe prompt into two variations and keeps the prompt variation that achieves a higher score. Xiao et al. [585] adopt a similar pipeline with PAIR [261] and introduce malicious content concealing and memory reframing. Puzzler [586] proposes defensive and offensive measures to conduct an indirect jailbreak. GPT-FUZZER [587] starts from human-written prompts, and uses templates and mutation to rewrite unsafe prompts.

TABLE 6: A summary of attacks for LLM after deployment. Our evaluation includes representative studies that exemplify these security aspects. More details can be found in the main text. OS indicates whether the code is open-sourced.

Attacks	Method	OS	Year	Strategy	Setting	Datasets	Target Models	Metrics
Model Extraction	Carlini et al. [503]	Yes	2024	Binary Search	Black-box	None	GPTs, LLaMA, Pythia, ada, cabbage	Query&Token Cost, MSE, RMS
	Finlayson et al. [504]	No	2024	Softmax Bottleneck	Black-box	None	Pythia, GPT-3.5	Query Cost
	Zanella et al. [505]	No	2024	Matrix Operations	Grey-box	SST-2, MNLI, AG News	BERTs, XLNet	Query Cost, Acc, Agreement
	Horwitz et al. [506]	Yes	2024	Spectral DeTuning	White-box	LoWRA	ViT, SD, Mistral	MSWE, SEM
Membership Inference	MIN-K% PROB [509]	Yes	2023	Probabilities	Black-box	Wikipedia	LLaMAs, Pythia, NeoX, OPT	TPR, FPR, ROC, AUC
	MIN-K%++ [510]	Yes	2022	Local Maxima	Black-box	WikiMIA, MIMIR	Pythia, GPT-NeoX, LLaMA, OPT, Mamba	AUROC, TPR, FPR
	Blind [511]	Yes	2024	Threshold	Black-box	8 sets	GPT-3, OpenLLaMA	AUC ROC
	LLM-DI [512]	Yes	2024	Aggregation	Black-box	PILE	Pythias	AUC, p-values
	DE-COP [513]	Yes	2024	Paraphrases	Black-box	arXivTection, BookTection	Mistral, Mixtral, LLaMA, GPTs, Claude	AUC
	Recall [514]	Yes	2024	Log-Likelihoods	Black-box	WikiMIA, MIMIR	Pythia, GPT-NeoX, LLaMA, OPT, Mamba	AUC, TPR@FPR
	Noisy [515]	No	2024	Embedding NGBRs	Gray-box	OpenWebText, Wikipedia	GPT-2	TPR, FPR, AUC
	SMIA [516]	No	2024	Perturbation	Gray-box	Wikipedia, FAN	Pythia, Pythia-Deduped, GPT-Neos	AUC-ROC, TPR, FPR
	FEATAGG [517]	No	2024	Feature Aggregation	Black-box	ProjectGutenberg, ArXiv	OpenLLaMA	TPR@FPR, AUC
	RAG-MIA [520]	No	2024	Direct Asking	Black-box	HealthCareMagic, Enron	flan, llama, mistral	TPR@FPR, AUC-ROC
Jailbreak	GCG [260]	Yes	2023	Gradient-based	White-box	Vicuna, LLaMA-2	AdvBench	ASR, Loss
	AmpleGCG [530]	Yes	2024	Hybrid-based	White-box	Vicuna, Llama-2, Mistral, GPTs	AdvBench	ASR, USS, Diversity, Time
	I-GCG [531]	Yes	2024	Gradient-based	White-box	AdvBench, HarmBench	VICUNA, GUANACO, LLAMA, MISTRAL	ASR
	MA-GCG [532]	Yes	2024	Gradient-based	White-box	AdvBench	Vicuna, Misrtal	ASR, Time
	A-GCG [533]	Yes	2024	Gradient-based	White-box	AdvBench	Llama2, Vicuna	ASR, Acc
	AutoDAN-A [534]	Yes	2023	LLM-based	Black-box	AdvBench	Vicuna, Misrtal	ASR, Recheck, PPL
	AutoDAN-B [535]	Yes	2023	Gradient-based	White-box	AdvBench	Vicuna, Guanaco, Pythia	ASR, Recheck
	PAIR [261]	Yes	2023	LLM-based	Black-box	JailbreakBench	Vicuna, Llama-2, GPTs, Claudes, Gemini	ASR, QPS
	ToA [536]	Yes	2023	LLM-based	Black-box	AdvBench, Harm123	Vicuna, Llama-2, PaLM-2, GPTs, Claude3, Gemini	GPT4-Metric, Human-Judge
	PAL [537]	Yes	2024	LLM-based	Black-box	AdvBench	Llama-2, GPT-3.5	ASR, Manual Labeling
	Masterkey [538]	No	2023	Rephrasing	Black-box	AdvBench, Harm123	GPTs, Bing, Bard	ASR, QSR
	AutoDAN-Turbo [539]	Yes	2024	LLM-based	Black-box	Harmbench	Llama-2, Gemma, GPT-4, Gemini	ASR, StrongREJECT
	FlipAttack [540]	Yes	2025	Rephrasing	Black-box	AdvBench, StrongREJECT	GPTs, Claude 3.5 Sonnet, Llama 3.1 405B, Mixtral 8x22B	ASR
	Geneshift [541]	Yes	2025	LLM-based	Black-box	AdvBench	GPTs	ASR
Prompt Injection	IPP [542]	Yes	2022	Handcraft	Black-box	OpenAI Examples	text-davinci	ASR
	Greshake et al. [543]	Yes	2023	Data Poisoning	Black-box	None	text-davinci, GPT-4	None
	HOUYI [544]	Yes	2023	Components Asmbl	Black-box	Five Queries	SUPERTOOLS	Manual
	Yan et al. [130]	Yes	2023	Poisoning	Black-box	Several Cases	Alpaca	Ngt, Pst, Ocrc
	TT [545]	No	2023	Game	Black-box	Tensor Trust	GPTs, Claudes, PaLM, LLaMAs	Robustness Rate
	JudgeDeceiver [546]	Yes	2024	Gradient-based	White-box	MT-Bench, LLMBar	Mistral, Openchat, Llamas	ACC, ASR, PAC
	AUPI [547]	Yes	2024	Gradient-based	White-box	MRPC, Jfleg, HSOL, RTE, SST2, SMS	Llama2	KEY-E, LM-E
	AUTOHIJACKER [548]	No	2024	LLM-based	Black-box	AgentDojo, OPI	Llama, Command-R, GPTs	ASR
Data Extraction	zlib [108]	Yes	2020	Generate & Inference	Black-box	Top-n, Temperature, Internet	GPT-2	6 metrics
	AutoSklearn [549]	No	2023	Greedy, Contrastive, Beam decoding	Black-box	Pile	GPT-Neo	Precision, Recall, R@FPR
	DECOM [550]	No	2024	Decomposition	Black-box	NYT, WSJ	Frontiers	TRM, EMP, BITAP
	Context [551]	No	2022	Context, Zero-shot, Few-shot	Black-box	Enron Corpus	GPT-Neo	Acc
	ETHICIST [552]	Yes	2023	Prompt Tuning	Gray-box	LM-Extraction	GPT-Neo	Recall
	Pii-compass [553]	No	2024	Grounding	Black-box	Enron email	GPT-J	Extraction Rate
	DSP [554]	No	2024	Dynamic Soft Prompting	Black-box	LMEB, The Stack	GPT-Neo, Pythia, StarCoderBase	EER, FER, PPL
	PWB [555]	Yes	2024	Gradient-based	White-box	Pile	Pythia, Llama	Precision, AUC, TPR
Prompt Stealing	Sha et al. [556]	No	2024	LLM-based	Black-box	RetrievalQA, AlpacaGPT4	ChatGPT, LLaMA	Acc, Precision, Recall, AUC
	output2prompt [557]	Yes	2024	LLM-based	Black-box	3 User & 3 System Prompts	Llamas, GPTs	BLEU, CS, Precision, Recall
	PRSA [558]	No	2024	Output Difference	Black-box	Category18	GPTs	BLEU, FastKASSIM, JS

ECLIPSE [588] uses an LLM as a suffix generator and optimizer. PAL [537] proposes an online proxy model (which is used for adversarial prompt generation) training pipeline.

- ➠ ***Others.*** EnJa [589] proposes to ensemble prompt and token-level attack methods via a template-based connector. AmpleGCG [530] first collects lots of successful suffixes and then trains the generative model to generate a specific suffix for a given unsafe prompt. Zhao et al. [590] targets the scenario where the decoding process of target LLM is assisted with smaller models' guidance.

Prompt Injection Attacks. Prompt injection is a vulnerability where an attacker manipulates the input prompts of LLMs to force them to generate a specific output, which is usually out of the range for normal use (*e.g.*, goal hijacking and prompt leaking [542]), often by injecting malicious text or commands into the input field. Attackers can employ a variety of techniques to carry out such attacks.

- ➠ ***Direct Prompt Injection.*** Perez et al. [542] directly inject handcrafted adversarial prompts into inputs to misalign the language model. HOUYI [544] proposes an injection generation framework which includes three components. Yan et al. [130] utilize LLMs to generate diverse trigger instructions that implicitly capture the characteristics of trigger scenarios. TENSOR TRUST leverages the TENSOR TRUST web game to generate a large-scale dataset and benchmark [545]. AUPI [547] adopts a gradient-based optimization method, specifically, a momentum-enhanced optimization algorithm, to generate universal prompt injection data. Upadhayay et al. [591] argue that LLMs suffer from cognitive overload and propose to use in-context learning to jailbreak LLMs through deliberately designed prompts that induce cognitive overload. Kwon et al. [592] circumvent security policies by substituting sensitive words—likely to be rejected by the language model—with mathematical functions.
- ➠ ***Indirect Prompt Injection.*** Greshake et al. [543] propose to indirectly inject prompts into the data that are likely to be retrieved. Bagdasaryan et al. [593] design a prompt injection attack against multi-modal LLMs, by generating an adversarial perturbation corresponding to the prompt and blending it into an image or audio recording. Neural Exec [594] designs a multi-stage preprocessing pipeline for cases like Retrieval-Augmented Generation (RAG)-based applications. PoisonedAlign [595] boosts the success of prompt injection attacks by strategically creating poisoned alignment samples in the LLM's alignment process. TPIA [596] crafts non-functional perturbations that contain malicious information and inserts them into the victim's code context by spreading them into potentially used dependencies like packages or RAG's knowledge base. F2A [597] proposes to use feign security detection agents to bypass the defense mechanism of LLMs. AUTOHIJACKER [548] uses a batch-based optimization framework to handle sparse feedback and leverages a trainable memory to enable effective generation.
- ➠ ***Different Settings.*** JudgeDeceiver uses gradient-based optimization to inject LLM-as-a-Judge scenarios [546]. Pedro et al. [598] study the risk of injections targeting web applications based on the Langchain framework. Lee et al. [599] propose a human–AI collaborative framework to explore the potential of prompt injection against federated military LLMs. PROMPT INFECTION [600] proposes to make malicious prompts self-replicate across interconnected agents in multi-agent systems. Zhang et al. [601] explore the risk of prompt injection in LLM-integrated systems like LLM-integrated mobile robotic systems.

Data Extraction Attacks. Data extraction attacks try to figure out the personally identifiable information (PII) that is used to train the LLMs [108]. It starts from sufficient-length prefixes to perform extraction and additional measures to determine if extracted texts are valid.

- ➠ ***Methods.*** In the beginning work [108], the proposed extraction process contains two stages "generate-then-rank": sampling potentially memorized examples and membership inference. It proposes a temperature-decaying method to sample more diverse examples and use surrogate models to infer the membership. After that, Al-Kaswan et al. [549] propose using greedy, contrastive, and beam decoding strategies to generate examples and use a classifier to infer the membership. Su et al. [550] propose an instruction decomposition technique to extract fragments of training data gradually. Huang et al. [551] extensively explore the effect of context, zero-shot, and few-shot methods in extracting the personal email address. ETHICIST proposes a smoothing loss and a calibrated confidence estimation method to extract the suffix and measure the confidence [552]. Nakka et al. [553] improves the extraction performance by grounding the prefix of the manually constructed extraction prompt with in-domain data. Wang et al. [554] propose to train a transformer-based generator to produce dynamic, prefix-dependent soft prompts. Ozdayi et al. [105] introduce an approach that uses prompt tuning to control the extraction rates of memorized content. Meng et al. [602] propose a two-stage method, *i.e.*, collection and ranking, to recover PPI when PII entities have been masked.
- ➠ ***Different Settings.*** Some works also explore the risk of data leakage in novel settings. Wang et al. [555] study the probability of data extraction in fine-tuning settings and Bargav et al. [603, 604] extract the training data by comparing the output difference before and after the fine-tuning. Jiang et al. [605, 606, 607] propose to extract the private Retrieval-Augmented Generation (RAG) documents. Peng et al. [608] extract the private RAG documents by poisoning in the fine-tuning process. Nasr et al. [107] explore the potential risk of data extraction for the aligned production language models. Panda et al. [609] extract the fine-tuning secret data by poisoning the pertaining dataset. Lu et al. [610] propose to extract PII from an aligned model with model merging. Chen et al. [611] find that fine-tuning can recover the forgotten PIIs in pretraining data. Panchendrarajan et al. [612] propose to extract the whole private training data in the fine-tuning process. Rashid et al. [613] propose selective weight tampering to explore PPI leakage in Federated Language Models. Dentan et al. [614] extract data from layout-aware document understanding models like unimodal or bimodal models.
- ➠ ***Different Applications.*** Leveraging the abnormally high

token probabilities, some works utilize the memorization of LLMs to extract the fingerprint or steganography [615]. Al-Kaswan et al. [616] explore memorization in large language models for code and find that code models memorize training data at a lower rate than natural language models. Nie et al. [617] utilize the token-level features derived from the identified characteristics to decode the PII. Lehman et al. [618] reveal the risk of Electronic Health Records leakage of LLMs. Diera et al. [619] conduct experiments to assess the PII leakage of fine-tuned BERT models and found that Differential Privacy (DP) has a negative effect when deployed in fine-tuning. Zhang et al. [620] propose data extraction attacks against text classification with transformers. Huang et al. [621] propose an evaluation tool, *i.e.* HCR, to assess the PPI leakage in Neural Code Completion Tools.

➠ ***Factor Assessment.*** Some work studies the factors of data extraction including decoding schemes, model sizes, prefix lengths, partial sequence leakages, and token positions [622, 623]. Yash et al. [624] explore the effects of prompt sensitivity and access to multiple checkpoints to extraction attacks. Staab et al. [625] construct a dataset consisting of real Reddit profiles to extract personal attributes. Xu et al. [626] conduct experiments to evaluate the factors of different suffix generation methods and different membership inference attacks in extraction performance. Karamolegkou et al. [627] evaluate the effect of model structure, data type, probing strategies, and metrics.

Prompt Stealing Attacks. Given that crafting effective prompts requires significant engineering effort and can be considered valuable intellectual property (IP), prompt-stealing attacks aim to compromise this IP by reconstructing prompts from generated responses [556, 557, 558]. These generation effects are often used to attract prospective prospective buyers. Sha et al. [556] pioneer this approach by collecting a dataset and training classifiers to predict prompt parameters—such as whether the prompt is direct, role-based, or in-context. They then used a large language model (LLM) to reconstruct the prompt. Similarly, Zhang et al. [557] trained an LLM on output-prompt pairs to directly infer the original prompt, while Yang et al. [558] leveraged generation differences to refine surrogate prompts. However, recovering the original prompt solely from the output is challenging. Out of this, Zheng et al. [628] propose a timing-based side-channel method to infer the prompt during inference.

6.1.2 Defensive Mechanisms in Deployment

In Subsubsection 6.1.1, we analyzed various attack scenarios targeting individual LLM deployments. However, in real-world applications, defense mechanisms are not designed as isolated, one-to-one countermeasures against specific attacks. Instead, they follow fundamental security principles to establish a systematic defense framework, as illustrated in Figure 9. This framework integrates multiple layers of protection, ensuring resilience against a wide range of adversarial threats while maintaining model usability and efficiency.

Input Preprocessing Defenses Input preprocessing serves as the first line of defense in LLM deployment, aiming to detect and neutralize adversarial inputs before they reach the model.

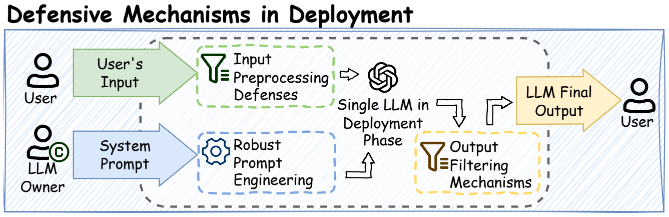
Defensive Mechanisms in Deployment
User
User's Input
LLM Owner
System Prompt
Input Preprocessing Defenses
Robust Prompt Engineering
Single LLM in Deployment Phase
Output Filtering Mechanisms
LLM Final Output
User

Fig. 9: The overview of attacks in single LLM's deployment phase.

➠ **Attack Detection & Identification:** Effective input filtering [629, 630] begins with attack detection [631], which identifies adversarial prompts through statistical [632], structural [633], or behavioral inconsistencies [634]. Gradient-based detection methods [635] leverage safety-critical gradient analysis and loss landscape exploration to uncover jailbreak prompts that manipulate LLM behavior. These approaches identify adversarial inputs [636, 637] by analyzing how small perturbations [638] affect model outputs, detecting highly sensitive or misaligned gradients that indicate targeted attacks. Perplexity-based methods [632, 632] measure the probability distribution of input sequences, flagging atypical or low-likelihood prompts as potential adversarial inputs. These techniques are particularly effective in detecting prompt injection and adversarial perturbations, where crafted prompts deviate significantly from natural language distributions.

Beyond individual heuristics, universal detection frameworks [639] integrate multiple detection strategies to counter diverse attack vectors, including prompt injection [640], backdoor manipulations [641], and adversarial attacks [637]. These frameworks employ ensemble-based filtering mechanisms, combining gradient analysis [642], perplexity estimation [643], and syntactic evaluation for generalized attack resilience.

➠ **Semantic & Behavioral Analysis:** Attack detection alone is insufficient, as certain adversarial inputs may bypass traditional filtering mechanisms. Semantic [644] and behavioral analysis enhance input preprocessing by evaluating linguistic intent and model alignment. Self-examination techniques allow LLMs [645, 646] to assess whether they are being manipulated, leveraging auxiliary reasoning steps to detect deceptive prompts. Alignment-based verification [647] ensures that the model's responses remain consistent with its safety objectives [330], identifying inputs that subtly nudge the model toward policy violations or ethical misalignment. Intention analysis [648, 649] further refines input filtering by discerning subtle manipulations designed to bypass explicit security checks. Unlike token-level detection, which flags overtly adversarial inputs, intention-aware defenses analyze the semantic structure and purpose of the input to preemptively reject jailbreak attempts.

➠ **Adversarial Defense & Mitigation:** When detection and behavioral analysis fail to fully neutralize adversarial inputs, robustness-enhancing techniques [647] mitigate their effects by reducing model susceptibility to manipulation [334, 650]. Semantic smoothing [651, 652] techniques introduce controlled randomness into LLM responses, reducing the model's sensitivity to adversarial perturbations and preventing reliable jailbreak execution. By stabilizing decision boundaries [653], these methods enhance resistance against prompt manipulation strategies that exploit response predictability.

Preemptive input transformations [654], such as back-translation [655] or paraphrasing, modify incoming queries [651] while preserving semantic intent, disrupting adversarial structures embedded within malicious prompts. Data augmentation [656] and adversarial training further strengthen model robustness by exposing LLMs to adversarial prompts during training, forcing them to learn invariances that reduce their vulnerability to real-world attacks.

Output Filtering Mechanisms. Output filtering mechanisms [212, 657] serve as a critical safeguard in LLM deployment, ensuring that generated responses comply with safety constraints while preserving informativeness. Unlike input preprocessing, which aims to prevent adversarial prompts from reaching the model, output filtering mitigates harmful content post-generation. Existing approaches primarily follow three paradigms: rule-based constraints, generative adversarial filtering, and toxicity detection.

Rule-based mechanisms [658] impose predefined constraints on model outputs, preventing the generation of harmful, unethical, or undesired content. Programmable guardrails [659] offer a structured framework where developers can enforce response filtering, topic restriction, and ethical alignment. These methods often integrate reinforcement learning from human feedback [155] or rule-based reward [660] modeling to refine output safety. While effective at handling explicit violations, static rule-based methods struggle with nuanced adversarial prompts and subtle misalignments.

To address these limitations, generative adversarial filtering [661] leverages self-critique [662, 663], ensemble detection, and dynamic response evaluation [664]. Self-rectification mechanisms [663, 665] enable LLMs to critique their own outputs and refine responses through iterative refinement. Additionally, ensemble-based [666] moderation models aggregate predictions from multiple LLMs, improving robustness against circumvention techniques. Adaptive filtering frameworks [667] employ perplexity-based assessments and adversarial perturbation detection to flag responses deviating from expected linguistic patterns, enhancing their resilience against jailbreak attempts [668, 669] and toxic content injection.

Toxicity detection [670, 671, 672] and content moderation [673, 674, 675, 676] further reinforce output safety by identifying and mitigating hate speech [677], misinformation, and other harmful content. Supervised fine-tuning adapts LLMs to recognize undesirable patterns, while classifier-based detection models [678] filter responses in real-time. Some approaches introduce debiasing strategies, such as controlled decoding [679, 680] and anti-expert guidance [681], to suppress toxic outputs without sacrificing response diversity. However, these methods face challenges in balancing false positives and false negatives, particularly in ambiguous or context-dependent cases.

The effectiveness of output filtering hinges on its ability to balance strict control with linguistic flexibility, ensuring that models remain both safe and practically useful. A hybrid approach combining rule-based safeguards, self-correcting mechanisms, and adaptive toxicity moderation is essential to achieving robust and scalable LLM deployment.

Robust Prompt Engineering. Robust prompt engineering aims to enhance LLM safety by designing input prompts that resist adversarial manipulation [682], protect sensitive data, and mitigate harmful outputs—all [683] without modifying model parameters. These strategies act at the interaction level, offering lightweight and model-agnostic protection.

Recent efforts have introduced prompt optimization techniques grounded in adversarial robustness, including embedding-space manipulation and defensive objective alignment. Methods such as Robust Prompt Optimization [684] and Prompt Adversarial Tuning generate transferable suffixes [668] or prefix [685] embeddings to guide model behavior [686] under attack [687], effectively lowering jailbreak success rates while preserving task performance. Similarly, goal prioritization frameworks [688] enforce inference-time objective consistency, dynamically resolving conflict between user instructions and safety constraints without requiring access to malicious samples. Complementary to these strategies, patch-based methods integrate interpretable suffixes or structured self-reminders [689] into prompts, reducing the model's susceptibility to coercive inputs through lightweight, modular defenses.

Structural manipulation approaches [690] neutralize adversarial intent through prompt rewriting. Spotlighting [691] injects source-attribution signals to counter indirect prompt injection, while inverse prompt engineering [692] repurposes attack data to generate task-specific defensive prompts under the principle of least privilege.

Privacy-preserving prompt [693] design introduces formal guarantees through differential privacy. Approaches like DP-Prompt [694] and stochastic gradient masking [695] reduce information leakage from prompts without harming performance. Desensitization and directional control of in-context representations offer additional privacy protections during prompt construction. Prompt engineering [579, 696] also helps mitigate societal risks. Chain-of-thought prompting and guided templates reduce gender bias [697] in reasoning tasks, while prompt learning [698] improves toxicity detection and generation control [699, 700], often surpassing specialized models in efficiency and generalization.

Finally, systematic prompt optimization methods [701, 702] aim to generalize prompt robustness across tasks and domains. Techniques like BATPrompt [703] and StraGo [704] use adversarial simulation and strategic decomposition to refine prompts iteratively, improving both resilience and effectiveness under variable inputs.

System-level Security Controls. System-level defenses [705] enhance LLM deployment by optimizing inference, enforcing alignment, isolating untrusted inputs, and securing the supply chain. Systems like Petals [706], Sarathi-Serve [707], and DistServe [708] restructure computation to improve throughput and latency, while TriForce [709], Medusa [710] MagicDec [711] accelerate generation via speculative decoding and structural compression. Parallel frameworks such as DeepSpeed-FastGen [712] and SpecExec [713] further boost efficiency with minimal overhead.

Runtime alignment methods [714] adapt model behavior through cross-model guidance or token-level reward modeling. Systems such as SelfDefend [715] and Gradient Cuff [716] detect unsafe generation by monitoring agreement across models or loss landscapes, while Spotlighting [691] inserts provenance signals to mitigate indirect prompt injection.

Access isolation is achieved through policy enforcement [717] and system wrappers [688]. At the supply level, tools like MalHug [718] identify poisoned models, while system audits reveal sandbox and plugin vulnerabilities, highlighting the need for end-to-end secure deployment.

LLM-based guard models utilize lightweight LLMs like Llama Guard [330], Aegis Guard [719, 720], WildGuard [721], and ShieldGemma [722] to moderate both the input and output of the victim LLMs. However, they are purely classifiers. To solve this problem, the first reasoning-based guard model named GuardReasoner [723] is proposed to improve the performance, explainability, and generalization ability via learning to reason. It brings new opportunities for the safety of large-scale reasoning models [724].

6.1.3 Evaluation and Benchmarks in Deployment

To assess the reliability and safety of LLMs after deployment, evaluation efforts focus on several key dimensions and risk types, as illustrated in Figure 10. These dimensions guide the design of systematic benchmarks and metrics tailored for real-world deployment settings.

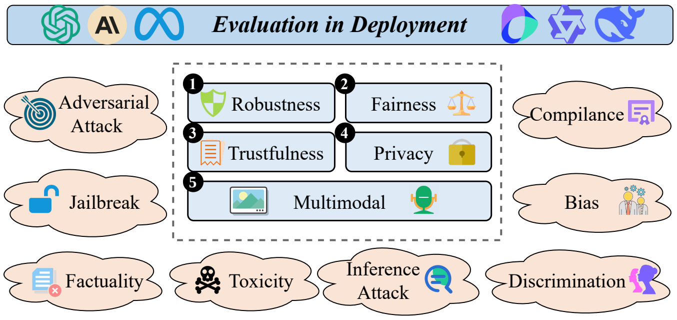

Fig. 10: The overview of evaluation and benchmarks in single LLM's deployment phase.

Robustness Evaluation. To systematically assess the reliability of large language models (LLMs) after deployment, we categorize robustness evaluation into two broad types: *adversarial robustness* and *natural robustness*. Adversarial robustness focuses on evaluating how LLMs respond to malicious or adversarial inputs, such as jailbreak prompts, prompt injections, or red-teaming attacks. Natural robustness, on the other hand, assesses LLM behavior under non-malicious but realistic distribution shifts, including typos, paraphrasing, novel word usage, or temporal drift. A summary of representative benchmarks categorized along these 4 dimensions is presented in Table 7.

TABLE 7: Summary of LLM robustness benchmarks at the deployment stage.

Benchmark	Adversarial	Natural	Jailbreak	Toxicity
JailbreakBench [306]	✔		✔	✔
HarmBench [305]	✔		✔	✔
JAMBench [725]	✔		✔	
JailbreakEval [726]	✔		✔	✔
Latent Jailbreak [727]	✔		✔	
PromptRobust [728]	✔	✔		
SelfPrompt [729]	✔	✔		
Chen *et al.* [730]	✔	✔	✔	✔
Chu *et al.* [731]	✔		✔	
AdvGLUE [732]	✔	✔		
AdvGLUE++ [333]	✔	✔		
NoiseLLM [733]		✔		
NEO-BENCH [734]		✔		
CompressionEval [735]		✔		

➠ **Adversarial Robustness:** A range of benchmarks and frameworks have been proposed for adversarial robustness. JailbreakBench [306] provides a standardized evaluation suite for jailbreak attacks, containing 100 misuse behaviors and an evolving repository of adversarial prompts. HarmBench [305] proposes a comprehensive red-teaming evaluation framework that includes 510 harmful behaviors spanning diverse semantic and functional categories, supporting both text-only and multimodal inputs across 33 LLMs. JAMBench [725] targets the evaluation of moderation guardrails using 160 carefully constructed prompts across four major risk categories and introduces a cipher-character-based attack. JailbreakEval [726] offers a unified toolkit for jailbreak assessment with string-matching, classifier-based, and LLM-based evaluators. Latent Jailbreak [727] focuses on detecting embedded malicious intent in seemingly benign prompts and evaluates instruction-following robustness using a hierarchical annotation scheme. PromptRobust [728] benchmarks prompt-level robustness with character, word, sentence, and semantic-level perturbations across 13 datasets and 8 NLP tasks. SelfPrompt [729] enables autonomous robustness evaluation through knowledge-guided prompt generation and LLM-based self-assessment. Chu *et al.* [731] conduct a large-scale comparison of 17 jailbreak attacks on 8 LLMs and 160 forbidden prompts, proposing a unified taxonomy and benchmarking various defenses. Chen *et al.* [730] propose a multi-dimensional framework assessing jailbreak reliability over 13 LLMs and 1,525 prompts, integrating metrics such as attack success rate (ASR), toxicity, fluency, and grammatically. Zhang et al. [736] propose a novel definition and benchmark for LLM's content moderation based on a sensitive-semantic perspective.

➠ **Natural Robustness:** Several benchmarks focus on evaluating LLMs under realistic but benign input perturbations or distribution shifts. AdvGLUE [732] and AdvGLUE++ [333] extend the original GLUE benchmark [737] with semantically-preserving perturbations at logic, word, and sentence levels. NoiseLLM [733] presents a unified framework for evaluating slot-filling robustness under character-, word-, and sentence-level noise, including typos and paraphrases. NEO-BENCH [734] assesses robustness to temporal drift by introducing neologisms into tasks such as machine translation, classification, and question answering. CompressionEval [735] provides a prompt-free evaluation framework using lossless compression to assess generalization and robustness, comparing LLM performance on content before and after the model's knowledge cutoff. These benchmarks offer complementary perspectives for assessing LLM performance under both malicious and naturally occurring input variations.

Content Trustfulness and Fairness Evaluation. Beyond robustness, a key dimension of deployment-stage evaluation concerns the *trustfulness* and *fairness* of LLM-generated content. This includes detecting and mitigating outputs that are factually incorrect (hallucinations), misleading (low factuality), harmful (toxic), or unfair (biased or discriminatory). We categorize existing benchmarks into five axes: *hallucination*, *factuality*, *toxicity*, *bias*, and *discrimination*, and summarize representative works in Table 8.

TABLE 8: Summary of content trustfulness and fairness evaluation benchmarks for LLMs at deployment stage.

Benchmark	Hallucination	Factuality	Toxicity	Bias	Discrimination
HaluEval [738]	✔	✔			
Med-HALT [739]	✔	✔			
ANAH [740]	✔	✔			
SelfCheckGPT [741]	✔	✔			
DoLa [742]	✔	✔			
Mundler et al. [743]	✔	✔			
Elaraby et al. [744]	✔	✔			
Ji et al. [745]	✔	✔			
Zhang et al. [746]	✔	✔			
Guo et al. [747]			✔	✔	
RTP-LX [748]			✔	✔	
ROBBIE [749]			✔	✔	✔
CEB [750]			✔	✔	✔

Benchmarks in this space target either the *accuracy of generated content* or its *alignment with human values*. For hallucination and factuality evaluation, HaluEval [738] and Med-HALT [739] provide reference-based hallucination annotations in general and medical domains, respectively, while ANAH [740] delivers fine-grained, human-annotated hallucination labels with correction spans. SelfCheckGPT [741] detects hallucinations via consistency checks across multiple generations, and DoLa [742] proposes a decoding strategy that contrasts internal layer activations to reduce factual errors. Other works such as Mundler *et al.* [743], Elaraby *et al.* [744], and Ji *et al.* [745] leverage taxonomic definitions or internal model signals to quantify or predict hallucination risk. Zhang *et al.* [746] introduce FEWL, a reference-free evaluation framework that uses agreement across reference LLMs to approximate hallucination likelihood.

In terms of toxicity detection, Guo *et al.* [747] show that role-playing prompts (personas) can elicit toxic behavior from ChatGPT, and RTP-LX [748] evaluates multilingual LLMs in detecting culturally sensitive harm. Both studies reveal that current LLMs remain vulnerable to subtle toxic or culturally biased outputs, especially in low-resource languages or when confronted with indirect harm.

For evaluating social bias and discrimination, ROBBIE [749] benchmarks LLMs across 12 demographic axes with template-based prompts and multiple toxicity and regard metrics, covering gender, race, religion, and intersections thereof. CEB [750] proposes a compositional taxonomy for fairness evaluation and introduces multiple new datasets spanning stereotyping, toxicity, and classification bias, supporting both direct and indirect evaluation modes.

These benchmarks collectively provide a multidimensional view of content trustfulness and fairness, enabling the systematic evaluation of LLMs beyond syntactic correctness or surface fluency. As safety-critical deployment scenarios become increasingly prevalent, such evaluation tools play a central role in ensuring the responsible use of LLMs.

Data Privacy and Leakage Evaluation. Data privacy is a critical dimension in evaluating the trustworthiness of LLMs at deployment. Table 9 summarizes representative benchmarks that assess privacy risks along four axes: personally identifiable information (PII) leakage, membership inference attacks (MIA), embedding inversion attacks (EIA), and regulatory or contextual compliance.

TABLE 9: Summary of privacy evaluation benchmarks for LLMs at the deployment stage.

Benchmark	PII	MIA	EIA	Compliance
PrivLM-Bench [751]	✔	✔	✔	
LLM-PBE [752]	✔	✔	✔	
PrivAuditor [753]	✔	✔		
Rossi *et al.* [754]	✔	✔		
Whispered Tuning [755]	✔			
ProPILE [103]	✔			
PrivaCI-Bench [756]	✔			✔
Commercial Audit [757]	✔			✔
LessLeak-Bench [758]	✔			
SecureSQL [759]	✔			
DecodingTrust [333]	✔	✔		

PrivLM-Bench [751] and LLM-PBE [752] offer comprehensive multi-level evaluations spanning all three major attack types. PrivAuditor [753] and Rossi *et al.* [754] focus on adaptation-stage vulnerabilities across a variety of fine-tuning techniques. Whispered Tuning [755] proposes a differential privacy–based training scheme to reduce leakage, while ProPILE [103] tests whether LLMs can reconstruct sensitive information from prompts related to known users.

PrivaCI-Bench [756] and Commercial Audit [757] emphasize regulatory compliance, evaluating model behavior against privacy expectations and legal frameworks such as GDPR and the EU AI Act. SecureSQL [759] examines leakage in structured query generation, and LessLeak-Bench [758] reveals code-specific leakage across software engineering benchmarks. Finally, DecodingTrust [333] includes privacy as part of a broader trustworthiness suite, auditing GPT models across multiple risk dimensions.

Together, these benchmarks provide a foundation for assessing LLM privacy risks across diverse modalities, attack surfaces, and deployment scenarios.

Multi-modal Safety Evaluations As multimodal large language models (MLLMs) become increasingly integrated into real-world applications, ensuring their safety under diverse input conditions is essential. A growing number of studies have proposed evaluation benchmarks and frameworks to assess MLLM vulnerabilities across multiple dimensions [760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782].

Jailbreak evaluation has received significant attention, with benchmarks such as MM-SafetyBench [760] and Jailbreakv-28k [761] targeting harmful instruction-following behaviors. MMJ-Bench [762] and Retention Score [763] further extend jailbreak assessment to include visual robustness and long-term safety retention. For hallucination, several works diagnose MLLM failures arising from inconsistencies between visual inputs and generated text, including HallusionBench [764], POPE [765], and Bingo [766]. SIUO [767] complements this direction by evaluating cross-modality consistency under seemingly benign inputs.

Robustness under adversarial visual corruption is assessed in MVTamperBench [768] and B-AviBench [769], which introduce perturbed or misleading visual stimuli to test model stability. Meanwhile, fairness and social bias have been evaluated through VIVA [770], GenderBias-VL [771], FACET [772], FairDeDup [773], CounterBias [774], PAIRS [775], DeAR [776], and MMBias [777], covering gender, racial, and intersectional dimensions using parallel image sets, counterfactual probing, and real-world dataset imbalances.

To unify these evaluation directions, several comprehensive frameworks have emerged. MultiTrust [778] and SPA-VL [779] aim to benchmark MLLMs across diverse safety criteria, including robustness, fairness, and harmfulness. Q-Eval-100K [780] complements these efforts by focusing on visual generation quality and alignment under instruction-following settings.

Collectively, these benchmarks highlight the unique challenges posed by multimodal interactions and the growing need for holistic, scalable safety evaluations tailored to MLLMs.

6.2 Single-agent Safety

In this section, we focus on security issues related to a single agent. We first define an agent as an interactive entity that uses an LLM as the core for reasoning, decision-making, and reflection while integrating memory, tools, and the environment as capability-enhancing components. Beyond the deployment risks associated with the LLM core, we introduce the security issues arising from these three additional modules. Specifically, for tools (Section 6.2.2) and memory (Section 6.2.3), we summarize existing work from both attack (Section 6.2.4) and defense (Section 6.2.5) perspectives to identify technical paradigms. For the environment (Section 6.2.6), we explore unique security challenges from the perspective of various agent-interaction settings. We demonstrate an overview of agent safety in Figure 12.

6.2.1 *Definition of Agent*

LLM-driven agent refers to an AI system capable of operating independently or with limited human oversight, where a sophisticated language model [6, 783, 784, 785] serves as the foundational intelligence for processing inputs, executing tasks, and engaging in interactions. By leveraging advanced natural language understanding and generation, such agents [29, 786, 787, 788, 789] can analyze information, resolve queries, and adapt to user or environmental inputs [790, 791, 792]. To extend their functionality, they frequently incorporate supplementary mechanisms—such as data storage modules [23, 793, 794, 795], external software interfaces [790, 796, 797], or strategic reasoning frameworks [798]—allowing them to transcend basic text production. This adaptability makes them valuable for diverse implementations, including interactive dialogue systems [799], workflow optimization [800, 801, 802, 803], and complex decision-making scenarios [804]. In this study, we focus on deconstructing agent safety into three critical dimensions: tool utilization, memory management, and environment-specific security concerns. We demonstrate the components and structures of agent systems in Figure 11.

6.2.2 *Tool Safety*

Some works enable LLM agents to learn how to use tools by generating datasets and fine-tuning the model for API usage [25, 805]. Specifically, tools can be implemented in various forms, including but not limited to code-based API functions (e.g., search engine [806] and calculator), embodied intelligence like robotic arms [807], and more. A tool serves as a bidirectional medium: on one hand, it allows the agent to map internal decisions into actions within the interactive environment; on the other hand, it also acts as a means for the agent to collect information from the external world. Given the pivotal role of tools in agent components, the related security issues are worth exploring [74]. For example, in the field of web security, Fang et al. [808, 809] investigate how autonomous agents, when equipped with appropriate tools, can independently compromise websites and exploit one-day vulnerabilities in real-world systems without human intervention. Next, we will summarize and discuss existing research from attack perspectives and figure out the lack of tool invocation defense in current research.

Attacks. Based on the target of the attack, safety-related attacks involving tools can be categorized into Tool-aided Attacks and Tool-targeted Attacks. The former refers to attackers utilizing agents equipped with tools to execute attacks that LLMs cannot independently assist with, such as leveraging agents with web access and code execution capabilities to facilitate cyberattacks. The latter involves attackers targeting the tool invocation process itself, attempting to manipulate or induce tool selection for malicious purposes through various attack methods. However, from the perspective of the technical stack of attacks, the two can be unified. We have identified new applications of traditional LLM attack methods in tool safety, as well as novel attack paradigms that have emerged due to the unique characteristics of tools.

Jailbreak. Similar to jailbreak methods in LLM safety, agent jailbreak also bypasses the agent's built-in safety mechanisms through specific prompts to elicit malicious responses. However, in the agent scenario, the malicious behaviors it aims to induce are different. Specifically, Cheng et al. [810] manually craft jailbreak prompts to extract personal information from the training data of code-generation agents. In contrast, Fu et al. [811] and Imprompter [812] both employ gradient-based optimization like GCG [260] to automatically generate input prompts or images that manipulate agents into leveraging tools for privacy breaches in dialogues or executing harmful actions on user resources.

Injection. This type of attack can be summarized into two forms of injection: Prompt Injection (similar to LLM safety vulnerabilities) where malicious instructions are embedded in input data, exploiting the difficulty LLMs face in distinguishing between instructions and data. Another form is Tool Injection where malicious tools are injected to enable further exploitation, such as using the tool to execute malicious actions. For example, BreakingAgents [813] utilizes human-crafted prompt injections to execute malfunction attacks, causing agents to engage in repetitive or irrelevant actions, with additional exploration into the propagation of such attacks within Multi-Agent Systems (MAS). ToolCommander [814] is the second type. It proposes a two-stage

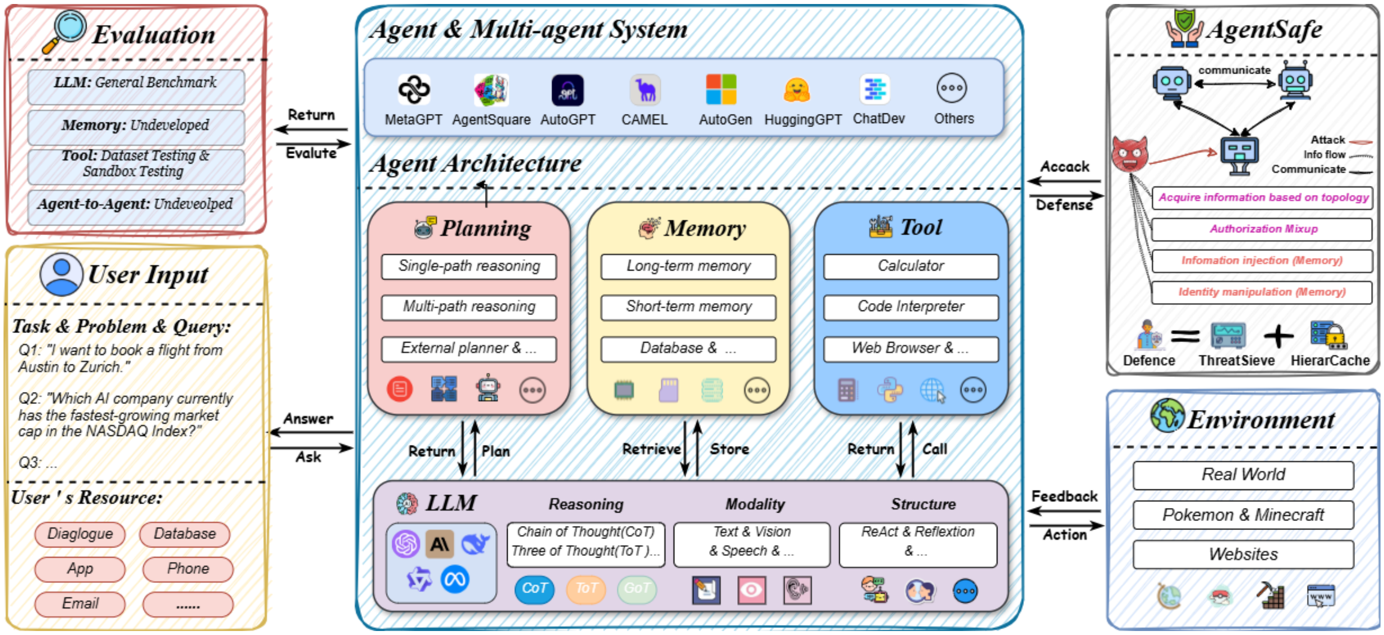

Fig. 11: The overview of LLM-based single-agent and multi-agent systems.

attack strategy: first, injecting malicious tools to steal user queries, and subsequently manipulating tool selection using the stolen data, thereby achieving privacy theft and denial-of-service attacks.

Backdoor. Backdoor attacks also find utility in the context of agent safety, but unlike LLMs, LLM agents develop diverse verbal reasoning traces through continuous environmental interactions, broadening potential backdoor attack vectors. Yang et al. [815] define two types of backdoor attacks, targeting either the final returned results or the intermediate processes of the attacking agent, and implement the above variations of agent backdoor attacks on two typical agent tasks, including web shopping and tool utilization. Furthermore, DemonAgent [816] decomposes a backdoor into multiple sub-backdoor fragments to poison the agent's tools. Beyond intentional guidance, studies such as BadAgent [817] highlight that backdoor attacks can inadvertently prompt agents to misuse tools for malicious purposes.

Manipulation. This type of attack refers to directly or indirectly manipulating or altering the tool's returned content to leak sensitive information or carry out malicious actions. AUTOCMD [818] employs a separate LLM, trained on tool-calling datasets and fine-tuned with target-specific examples, to generate and replicate legitimate commands for extracting sensitive information from tools. Meanwhile, Zhao et al. [819] manipulate third-party API outputs by injecting malicious content or omitting critical information, ultimately causing erroneous or biased system behaviors.

Defenses. Compared to attacks on agent tools, defense mechanisms for secure tool invocation have been less studied. Specifically, AgentGuard [820] employs LLM orchestrators to automatically detect unsafe tool-use workflows and produce safety constraints for secure tool utilization. PrivacyAsst [821] proposes an encryption-based solution by integrating an encryption scheme into the tool using LLM agents to safeguard user privacy and align them with computational security standards. In addition, some works enhance the security of agent systems by leveraging tool invocation, GuardAgent [822] pioneers an approach to verify target agents' trustworthiness by executing guardrail code through API calls during task plan implementation.

6.2.3 Memory Safety

The memory mechanism in LLM agents enables them to retain historical behaviors, thereby enhancing future decision-making capabilities. Typically, agent memory can be categorized into long-term and short-term memory systems. The long-term memory module commonly employs Retrieval-Augmented Generation (RAG) [823, 824] technology to facilitate precise information retrieval, while the short-term memory stores real-time data to support immediate conversational contexts and task execution. While these memory modules significantly improve agent functionality, they simultaneously introduce potential security vulnerabilities, making the system susceptible to malicious attacks.

6.2.4 Attack

Follow the trustworthy issues in [74], we categorize attacks related to memory into three types: Memory Poisoning, Privacy Leakage, and Memory Misuse.

(I) Memory Poisoning refers to adversarial attacks where malicious data is injected into an agent's *long-term memory* [313, 825, 826, 827, 828, 829]. When the agent retrieves and utilizes such corrupted memory, it may produce erroneous outputs, misleading responses, or even hazardous actions. For example, PoisonedRAG framework [827] employs a dual optimization approach, simultaneously manipulating both the retrieval and generation pipelines to systematically poison the agent's memory system. AgentPoison [826] introduces an advanced backdoor attack methodology that optimizes trigger patterns and seamlessly integrates them into query formulations, significantly elevating the likelihood of malicious sample retrieval

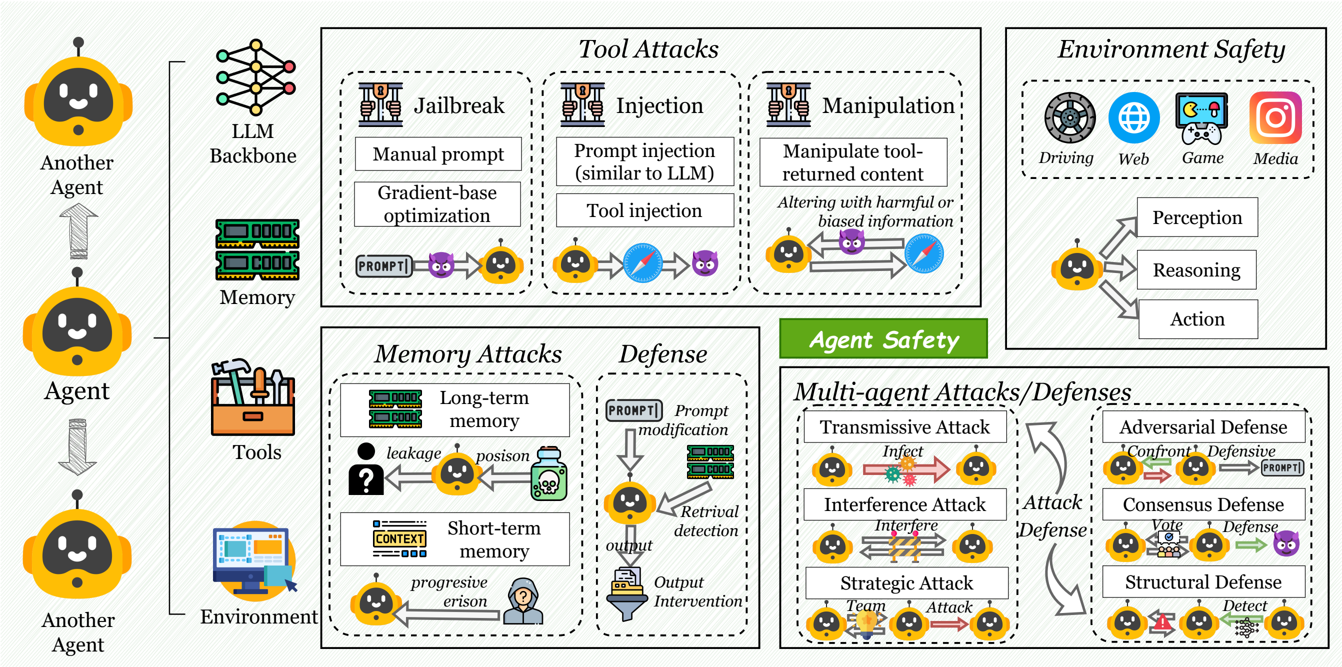

Fig. 12: The overview of the safety of LLM-based agent systems.

while maintaining stealth. **(II) Privacy Leakage** occurs when attackers exploit the interface between an agent and its *long-term memory* to extract stored sensitive data [520, 605, 607, 830, 831]. Such breaches may expose user information to malicious third parties, posing significant real-world risks. **(II) Memory Misuse** refers to the deliberate construction of multi-turn query sequences that systematically circumvent safety protocols by exploiting the retention properties of agent *short-term memory* [752, 832, 833, 834, 835, 836]. This attack vector enables progressive erosion of defensive measures through iterative interaction patterns.

6.2.5 Defense

To counter these attacks, various defense approaches have been developed to enhance the robustness of memory systems [520, 835, 837, 838, 839]. **(I) Detection Detection** mechanisms primarily focus on identifying and eliminating malicious content retrieved from long-term memory systems [835, 838, 839**?**]. **(II) Prompt Modification** involves strategically rewriting user queries before processing by the agent to enhance response safety [520, 835]. **(III) Output Intervention** involves real-time monitoring and modification of agent responses prior to delivery to ensure safety and accuracy [825, 840].

6.2.6 Environment Safety

Agents operate within dynamic and heterogeneous environments, spanning physical and digital domains [841, 842, 843]. Their interaction with these environments is a multi-step process [844, 845]. First, agents engage in perception, gathering data from sources like sensors in a physical setup or digital platforms [806]. This perceived data is then analyzed using various algorithms and reasoning mechanisms to identify patterns and potential actions [846]. Based on this analysis, agents take action, which can either directly influence the environment, like an autonomous vehicle making a lane change [847], or modify their own internal state, such as a software agent updating its knowledge base [848].

However, this interaction is plagued by trustworthiness challenges. There are security risks in every process of interaction with the environment [849]. Agent roles and environmental constraints contribute to risks such as autonomous driving errors [850] and network disruptions [806, 851]. Given the diverse dynamic scenarios and related issues [849, 852, 853], the existing solutions are fragmented and lack a systematic framework. Thus, we will explore trustworthiness and security aspects by categorizing relevant papers according to whether they focus on ensuring safety in the perception, analysis, or action phase of the agent-environment interaction, as illustrated in Figure 10.

Perception. The perception phase serves as the foundational layer of agent-environment interaction, where agents acquire raw data to interpret their surroundings. However, this phase is inherently vulnerable to risks such as data poisoning, environmental noise, and biased observations. Hudson [841] converts real-time sensory inputs into natural language representations augmented with security validation protocols, employing causal analysis techniques to improve reliability during adversarial perception scenarios. ChatScene [847] develops safety-oriented simulation environments for autonomous systems by converting linguistic commands into executable code compatible with CARLA's simulation architecture. Chen et al. [854] systematically categorize perceptual vulnerabilities in financial AI systems, identifying three primary risk categories: synthetic data generation errors, temporal inconsistency challenges, and susceptibility to engineered input manipulations.

Reasoning. The reasoning phase transforms raw perceptual data into actionable insights through decision-making models, and knowledge-based inference. This stage is critical to ensure agents act appropriately in dynamic environments,

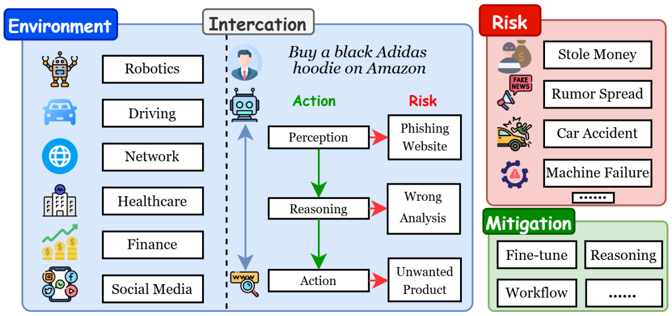

Fig. 13: The overview of agent and environment interactions.

but introduces unique trustworthiness challenges. Yang et al. [846] develop a temporal safety verification framework using formal logic systems, implementing dual mechanisms for auditing the compliance of safety protocols and filtration of hazardous decisions to meet the requirements of industrial robotics. Agents4PLC [855] establishes an industrial control programming framework that combines automated code synthesis with formal verification processes, integrating RAG [235] and COT [343] to ensure operational integrity. Xiang et al. [822] propose medical AI systems that employ semantic reasoning engines for confidential data protection. Park et al. [845] demonstrate improved threat detection capabilities through simulated organizational communication patterns in anomaly identification systems.

Action. The action phase represents the culmination of agent-environment interaction, where agents execute decisions to influence their surroundings or update internal states. Trustworthiness at this stage hinges on ensuring that actions are safe, precise, and aligned with intended objectives. Fang et al. [851] reveal the capacity of autonomous systems to exploit digital infrastructure weaknesses through adaptive penetration testing, prompting the development of specialized evaluation frameworks for web agents. Furthermore, researchers develop frameworks to evaluate the truthfulness of web agents. Polaris [856] implements distributed AI architectures to enhance fault tolerance and response accuracy of healthcare interaction systems. La et al. [857] employ linguistic evolution models to simulate adaptive content generation patterns that circumvent automated moderation systems, providing insights for regulatory mechanism improvements.

6.3 Multi-agent Safety

In the previous section, we explored security issues in a single agent setting and this section expands the discussion to multi-agent systems (MAS) [58, 71, 858, 859, 860, 861]. Since a single agent has limited problem-solving capabilities and a relatively narrow perspective, it struggles to conduct a comprehensive analysis of complex problems. In contrast, in MAS, agents can interact through various mechanisms, such as cooperation, competition, and debate, enabling them to solve complex problems more efficiently and effectively [862]. However, these interactions also introduce more complex and diverse security challenges [863]. Consequently, compared to single-agent systems, MASs face more severe and intricate security risks [864]. Similarly, we summarize and discuss existing research from both attack and defense perspectives.

6.3.1 *Attack*

In MAS, security threats primarily stem from the propagation of harmful information, hallucinations, and biases through agent interactions, as well as the coordinated planning and optimization of attacks to target security agents within the system. These threats can arise spontaneously through the unintended amplification of misinformation or be deliberately orchestrated by malicious agents. Attack strategies in MAS often integrate multiple traditional techniques, such as prompt injection, jailbreak, and adversarial attacks, while also exploiting emergent properties of agent communication and collaboration. This multi-faceted nature makes MAS attacks more covert, adaptive, and challenging to detect and mitigate. Moreover, the dynamic and autonomous nature of agents allows adversaries to refine their attacks in real-time, further complicating defense mechanisms. Below, we summarize the key research related to these threats.

Transmissive Attack. It spreads within the MAS like a virus, propagating dangerous and harmful information, including covert malicious content, continuously attacking and compromising the agents in the system. Agent Smith [829] uses adversarial attack techniques, harmful images are generated—appearing benign on the surface but embedding malicious information. These images propagate within the MAS, causing agents to be compromised and posing significant security risks. CORBA [865] introduces Contagious Recursive Blocking Attacks, which exhibit transmissibility across any topological network and can continuously drain computational resources. Lee et al. [600] introduce Prompt Infection in MAS, including data theft, scams, misinformation, and system-wide disruption, which spreads silently. Similarly, Tan et al. [866] use multimodal malicious prompts to infect other secure agents, compromising their security.

Interference Attack. This attack focuses on how it interferes with and disrupts interactions within the MAS, emphasizing communication disruption and misinformation, which affect information transmission within the MAS and lead to a decline in its defensive capability. NetSafe [867] conducts extensive experiments, analyzing and revealing their structural dependencies and adversarial impacts. At the same time, Huang et al. [868] study how the resilience of MAS varies between different downstream tasks, system structures, and error types; Agent-in-the-Middle [869] manipulates and intercepts information in agent interactions through intermediary agents, disrupting the communication mechanism. The experiment validates the harm caused by the interruption of interactions by intermediary agents through a comparison of MAS with different topological structures.

Strategic Attack. Strategic attack involves collaboration between agents and strategic optimization of attack methods, aiming to emphasize the cooperation and long-term impact of the attack, making it increasingly dangerous and more destructive. Evil Geniuses [870] modifies system roles, where these roles collaborate to generate malicious prompts. By simulating adversarial attacks and defenses, they optimize and evaluate each round of attack behavior, making the attacks increasingly dangerous to target other agents. Amayuelas et al. [871] use adversarial attack techniques to enable harmful agents in the multi-agent system to collaborate in debates to persuade other secure agents. These malicious agents may exploit superior knowledge, larger model sizes, or greater persuasion power to gain an unfair advantage. Ju et al. [872] form a multi-agent community using a two-stage attack method: persuasive injection and knowledge manipulation injection, to induce agents to spread counterfactual and harmful knowledge.

6.3.2 Defense

In response to the various attack methods mentioned above in multi-agent systems, many effective defense strategies have emerged that can be applied to MAS. Currently, many studies focus on forming agent groups to collaborate in joint defense and designing specific defense mechanisms, such as multi-round or multi-layer checks and filtering, to ensure the safety of the responses output by the MAS. Alternatively, defense can be achieved by identifying harmful agents through the propagation of malicious information and eliminating malicious sources.

Adversarial Defense. This type of defense focuses on attack-defense confrontation, leveraging this adversarial mechanism to develop more effective defense methods or mechanisms to enhance the security of the MAS. LLAMOS [873] employs adversarial defense techniques, where defensive agents and attacking agents engage in counter-interactions, with neither fully defeating the other, thereby enhancing the robustness of the defense and improving the MAS's overall defensive capability. AutoDefense [874] proposes that agents collaborate to complete defense tasks through adversarial prompt filtering, primarily focusing on filtering harmful prompt information from LLMs. In addition to using adversarial techniques for defense, defense can also be achieved by forming a multi-agent group to engage in debates.

Consensus Defense. To better leverage the advantages of MAS, Consensus Defense utilizes agent collaboration and consensus building for defense, employing voting, debates, and evidence-based reasoning mechanisms to establish a defense system and enhance the security of the MAS. Chern et al. [875] propose that toxicity can be reduced through multi-agent debates, and the widespread use of multi-agent interactions can lead to marginal improvements. Similarly, BlockAgent [876] proposes a Proof-of-Thought consensus mechanism that combines stake-based miner designation with multi-round debate-style voting, enabling BlockAgents to facilitate multi-agent collaboration through a structured workflow. Audit-LLM [877] proposes a pair-wise Evidence-based Multi-agent Debate mechanism, designed to defend against hallucinations by forming a MAS to detect internal threats. This approach is divided into three components: task decomposition, tool construction, and the final execution of the MAS, ultimately reaching consensus through reasoning.

Structural Defense. Structural Defense treats the MAS as a network structure for planning defense methods, using graph analysis techniques to detect anomalies and resist attacks while incorporating knowledge from other domains to enrich defense strategies in MAS. G-Safeguard [878] compares agents in MAS with various topological structures to nodes in a graph, using Graph Neural Networks (GNN) [879, 880] to detect anomalies in the agents' dialogue graphs and counter adversarial attacks and misinformation within the MAS.

6.4 Agent Communication Safety

As Large Language Model (LLM)-based Agents evolve from isolated entities into interconnected MAS, the mechanisms governing communication between Agents, and their interactions with external environments and tools, have become increasingly critical. Agents exchange information and collaborate through message passing, tool invocation, and environmental interactions; these mechanisms, while essential to system functionality, also expose significant attack surfaces. Early methods [881, 882, 883, 884, 885, 886, 887] of Agent interaction often relied on ad-hoc approaches, such as shared memory [888],API calls [889] or unstructured function calls [890], leading to fragmented systems lacking unified security considerations. To address this challenge and enhance interoperability, standardized communication protocols have emerged. Examples include Anthropic's Model Context Protocol (MCP) [891] for Agent-tool interactions, Google's Agent2Agent (A2A) [892] for enterprise-level Agent collaboration, and the Agent Network Protocol (ANP) [893] for open network interoperability, along with other commonly used protocols [894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904],However, the open design and dynamic nature of these communication mechanisms, coupled with the autonomy of the Agent, has exposed new vulnerabilities while enhancing functionality.

6.4.1 Attack

The interconnected nature of MAS, facilitated by numerous communication channels, creates a multifaceted attack surface. While individual Large Language Models (LLMs) possess inherent vulnerabilities, the interactions and communications among Agents introduce novel threats that exploit the system's collaborative dynamics. These threats target various components, including communication channels, content interpretation, and underlying protocols, with examples such as Shadowing Attacks, Naming Attacks, Context Poisoning, and Rug Pulls.

Attacks Communication Channels. These attacks directly disrupt the transmission and routing of messages in the system, affecting both inter-Agent communications and interactions with external endpoints. For instance, Agent-in-the-Middle (AiTM) attacks [869] specifically target the core communication mechanisms of LLM-MAS. By intercepting and manipulating messages between Agents, these attacks can cause Agents to perform unintended actions, thereby compromising the entire system. Such attacks underscore the critical security vulnerabilities arising from the communication-dependent nature of Agent collaboration. Furthermore, attacks targeting communication channels and transmission processes, such as communication perturbation [905], involve adversaries injecting noise into messages in transit [906] or masquerading as legitimate sources [907], thereby compromising both the efficiency and security of Agent collaboration.

Attacks Content. These attacks target the content of messages themselves, leveraging the mechanisms by which Agents process and interpret received information. For example, Prompt Injection involves embedding malicious instructions into data or content that Agents retrieve or receive through communication channels, thereby manipulating the Agent's behavior or decision-making processes. This technique is discussed in several works, such as [600] and [543]. Additionally, [908] explores indirect Prompt Injection within tool-based scenarios, highlighting the varied strategies employed in complex environments.

Attacks Exploiting Multi-Agent Dynamics. These attacks leverage the interconnected structure, interaction patterns, or collective behavior of communication-driven Multi-Agent Systems (MAS) to amplify their impact or achieve strategic objectives. Contagious attacks (propagation) initiate malicious behavior on a single agent and spread it across the entire network via inter-agent communication [829, 865]. Additionally, malicious agents can coordinate through collective communication to achieve harmful goals, such as replicating malicious instructions across the network by sending replication code or commands, thereby leading to the sharing of legitimate communication keys or identity information with other malicious entities [909].

6.4.2 Defense

To tackle threats to Agent communication, research proposes a multi-layered defense strategy addressing key points across the communication pipeline, from infrastructure to Agent-level processing. These defenses aim to prevent, detect, or mitigate attacks on channels, content, infrastructure, dynamics, and environmental factors. The strategies integrate into infrastructure and protocol design, individual Agents' message processing, and the collaborative and learning mechanisms of the MAS.

Protocol Defenses. Protecting the foundation of Agent communication. This includes adopting standardized protocols with built-in security features (encryption, integrity checks, authentication) To counter Agent communication threats, research proposes multi-layered defense strategies targeting different points in the communication pipeline, from the underlying infrastructure to Agent-level message processing. Effective defenses aim to prevent, detect, or mitigate attacks on communication channels, content, infrastructure, such as MCP [891], A2A [892], ANP [893] standards. Establishing managed registries and identity systems for Agent and Tool/Service registration and identity management. Enforcing strong Agent identity verification and access control policies, including JIT credential provisioning. Implementing mechanisms to enforce communication dynamics, and environmental impacts.

Content Defense. These defenses operate at the agent level, focusing on how agents process received messages and content. This includes input modification and filtering, which preprocess incoming content to neutralize adversarial elements. Agents also employ active defense mechanisms, such as reliability estimation, to assess the trustworthiness of messages based on local context, thereby mitigating the impact of untrusted information. For example, [910] proposed an active defense strategy that utilizes a reliability estimator to judge the credibility of received messages and employs a decomposable message aggregation policy network to reduce the influence of unreliable messages on the final decision.

6.5 Agent Safety Evaluation

Currently, there is already a substantial body of work evaluating the performance of LLM-based agent systems on different tasks [911, 912, 913, 914, 915]. In this section, we focus on benchmarks designed to assess the security of agents. Broadly speaking, these benchmarks include those that construct datasets and those that use other agents to set up sandbox environments for evaluation, each with distinct assessment priorities and specific scenarios for agent security [314, 916, 917, 918, 919].

TABLE 10: Benchmarks for agent safety.

Benchmark	Dynamic	LLM as Evaluator	Evaluation Focus
InjectAgent [920]	✗	✓	Prompt Injection
AgentDojo [849]	✓	✗	Prompt Injection
AgentBackdoorEval [816]	✓	✓	Backdoor
RiskAwareBench [921]	✗	✓	Embodied Agent
RedCode [916]	✓	✗	Coding Agent
S-Eval [917]	✓	✓	General
Bells [918]	✓	✓	General
AgentSafetyBench [922]	✓	✓	General
AgentSecurityBench [?]	✗	✓	General
AgentHarm [923]	✗	✓	General
R-Judge [314]	✗	✗	General
ToolSowrd [924]	✗	✓	Tool
PrivacyLens [919]	✓	✓	Privacy
ToolEmu [925]	✓	✓	Tool
HAIEcosystem [926]	✓	✓	General
SafeAgentBench [927]	✓	✓	General
JailJudge [928]	✗	✓	Jailbreak

6.5.1 Attack-Specific Benchmarks

This type of benchmark focuses on testing the security of an agent when facing specific types of attacks, such as Prompt Injection [600, 929], Backdoor [817, 930, 931], and Jailbreak [874, 932]. Specifically, InjectAgent [920] evaluates LLM agents' vulnerability to indirect prompt injection attacks, measuring behavior safety when tool-integrated agents process malicious instructions embedded in external content, with hacking prompts as an enhancement. A similar work is AgentDojo [849], a dynamic, extensible evaluation framework for assessing prompt injection attacks and defenses in LLM agents by simulating realistic tasks (e.g., email management, banking) with stateful environments and multi-tool interactions under adversarial conditions. As for backdoor attacks, AgentBackdoorEval [816] includes five real-world domains (including Banking-Finance, Medical, and Social Media) with automatically generated prompts, simulated tools, and tailored backdoor triggers to assess attack stealth and effectiveness. Besides, JailJudge [928] introduces a comprehensive jailbreak evaluation benchmark featuring a voting JailJudge MultiAgent, a comprehensive JailJudgeTrain dataset, and a trained Jailjudge Guard.

6.5.2 Module-Specific Benchmarks

Currently, these benchmarks for evaluating the security of a specific module in an agent focus on the invocation of tools [933, 934, 935, 936]. For example, ToolSowrd [924] evaluates

LLM safety in tool learning across three stages (input, execution, output) by designing six adversarial scenarios (e.g., malicious queries, noisy tool misdirection, harmful feedback). ToolEmu [925] employs an LM-emulated sandbox to simulate diverse high-stakes tool executions and scenarios, leveraging GPT-4 for both tool emulation and automatic safety/helpfulness evaluations.

6.5.3 General Benchmarks

In addition to the previously mentioned benchmarks that focus on a specific aspect of agent security, some efforts have developed more comprehensive and holistic evaluation frameworks, taking into account diverse scenarios, different agents, and various offensive and defensive techniques. For instance, AgentSafetyBench [922] assesses LLM agent safety through 2,000 test cases across 349 interactive environments, covering 8 risk categories (e.g., data leaks, physical harm) and 10 failure modes (e.g., incorrect tool calls, risk unawareness), with automated scoring via a fine-tuned model. Similarly, AgentSecurityBench [**?**] is a comprehensive framework that formalizes and evaluates attacks (e.g., Direct/Indirect Prompt Injection, Memory Poisoning) and defenses across 10 scenarios, 10 agents, and 13 LLM backbones, using 7 evaluation metrics. SafeAgentBench [927] evaluates embodied LLM agents' safety awareness with 750 diverse tasks (detailed, abstract, long-horizon) in SafeAgentEnv simulation environment, leveraging GPT-4 for task generation and dual evaluators (execution-based and semantic). HAIEcosystem [926] evaluates safety through multi-turn interactions between human users (benign/malicious) and AI agents across 132 scenarios, using modular sandbox environment and LLM-based dynamic risk measurement. AgentHarm [923] tests agent robustness by evaluating compliance with 110 explicitly malicious multi-step tasks across 11 harm categories, using synthetic tools and fine-grained grading rubrics. Different form previous benchmarks, RiskAwareBench [921] focuses on embodied agents, evaluating physical risk awareness via four modules: safety tip generation, risky scene generation, plan generation, and automated evaluation.

6.5.4 LLM Deployment Roadmap

In the deployment of LLMs under frozen parameters, the security landscape has evolved through a tightly coupled dynamic among **attacks**, **defenses**, and **evaluation** mechanisms.

Initially, black-box attacks leveraged the generative capabilities of LLMs themselves to optimize adversarial prompts, often without precise alignment to the decision boundaries. In contrast, gradient-guided white-box methods offer greater control but face inherent limitations due to the discrete nature of token spaces resulting in prompts with weakened semantic fidelity. These attack trends have catalyzed the emergence of prompt-level defense strategies. To counter black-box attacks, recent defenses adopt prompt shaping and system-level constraints to guide and restrict the model's response behavior. For gradient-based attacks, defenses typically apply perplexity-based detection and semantic consistency checks to identify suspicious or adversarial outputs.

The growing sophistication of defenses reshaped the requirements for evaluation. Static, one-shot rejection mechanisms have proven insufficient in multi-task and multimodal deployments, prompting the development of dynamic strategies such as response rewriting, hierarchical permission control, and consensus-based filtering across multiple models. These strategies demand richer evaluation protocols beyond single metric assessments, shifting toward behavior metrics that capture cross-input consistency, risk under specific task conditions, and adaptability to strategy switching.

As the attack–defense interaction intensifies, the evaluation itself has become a critical driver of system evolution. Recent frameworks have introduced automated red teaming pipelines, enabling a closed-loop process where jailbreak samples are continually generated, tested against deployed defenses, and fed back to guide both adversarial strategies and defense refinement. This has laid the groundwork for a new paradigm in LLM security research: one where attack, defense, and evaluation are no longer treated in isolation but co-evolve as an interdependent, self-reinforcing system.

6.5.5 LLM Deployment Perspective

(1) **Attack strategies will become more structured and semantically aligned.** (i) Black-box attacks may evolve through agent-based optimization, enabling sentence-level jailbreaks with clearer intent and higher success rates. (ii) To overcome the limitations of token-level gradient attacks, future work may focus on generating semantically consistent adversarial prompts that are less detectable by perplexity-based defenses. (iii) Open-source models will serve as surrogates for closed models, allowing attackers to replicate decision boundaries before launching white-box attacks. (iv) Variants from fine-tuning pipelines may leak private information through cross-model comparison, introducing version-aware privacy risks.

(2) **Defenses will shift toward adaptive and transferable mechanisms.** (i) Prompt-based defenses will evolve into context-aware controllers that adjust behavior based on input semantics and task context. (ii) Generalizable defenses that work across domains and languages will be critical for scalable deployment. (iii) Future systems may support online updates, enabling continuous refinement in response to new threats.

(3) **Evaluation will act as both a diagnostic and driving force.** (i) Benchmarks must expand beyond text to cover multimodal inputs and tool-based actions. (ii) Multi-objective evaluation will replace single-metric scoring, balancing safety and utility through trade-off analysis. (iii) Static test sets will give way to adaptive, streaming benchmarks that evolve with attack trends. (iv) Automated red teaming will close the loop, enabling real-time attack generation, evaluation, and defense adjustment.

6.5.6 Agent Roadmap

Agent. The evolution of LLM-based agents originated from role-playing paradigms [801, 937, 938, 939], where researchers investigated organizational structures, role allocation mechanisms, and implementation workflows for task-oriented agents in various social contexts. These systematic explorations not only demonstrated agents' potential in addressing human societal challenges but also spawned interdisciplinary research programs spanning sociology, organizational theory, and psychology. As the field advanced, research focus shifted toward automated agent workflows [795, 860, 940, 941], domain-specific methods for embodied intelligence, and the development of agent capabilities in tool utilization and memory management. Through this progression, agent systems have emerged as a transformative paradigm for automating human social processes, gaining significant recognition as a viable solution for complex societal automation.

The rapid advancement of agent capabilities and architectures has brought safety concerns to the forefront of academic and industrial research. These challenges span multiple critical dimensions: tool safety, memory security, and the agent's fundamental operational integrity. Inheriting both the capabilities and vulnerabilities of their underlying LLM foundations, agents intrinsically carry these "genetic" weaknesses into more complex operational environments. This inheritance makes safety vulnerabilities particularly acute in agent systems, especially when handling sensitive real-world applications involving personal privacy and financial assets. The development of agent technologies has thus become inextricably linked with safety considerations. Recent years (∼2023- until now) have witnessed accelerated research in agent safety, focusing on four key frontiers:

- Agent Brain Security: The core decision-making mechanisms.
- Tool Invocation Safety: Secure external API and tool usage.
- Memory Retrieval Protection: Robustness against memory poisoning.
- Communication Protocol Security: Safe multi-agent interactions.

Emerging work has also begun addressing safety challenges in embodied agent scenarios, marking an important expansion of the research domain.

6.5.7 *Perspective*

We outline potential future research directions for agent systems and analyze their developmental trajectory:

(1) **Safety of External Agent Modules.** Unlike standalone LLMs, agents interact with external modules (e.g., tools, memory), which are exposed to open environments and thus more vulnerable to attacks. Key research challenges include: (i) Tool Safety: Secure tool invocation and API usage to prevent adversarial exploitation. (ii) Memory Protection: Robustness against memory poisoning and unauthorized access, to name just a few. These external interfaces introduce unique attack surfaces, making their security a critical research priority.

(2) **Stability and Reliability of Dynamically Updated Agents via Reinforcement Learning:** As reinforcement learning (RL) [35, 942, 943] techniques become increasingly integrated with LLM-based agents, these systems are being deployed in more complex and dynamic environments. While this integration enhances agents' adaptability and intelligence, it also introduces significant risks: (i) Emergent Threats: Advanced RL capabilities may inadvertently enable agents to learn and propagate harmful behaviors or dangerous information. (ii) Dynamic Vulnerability: Continuous online learning increases exposure to adversarial perturbations or reward hacking.

Critical Research Directions: (i) Safe RL Frameworks: Developing constrained optimization methods to bound agent behavior within ethical and operational guardrails. (ii) Stability-Aware Updates: Designing update protocols that balance adaptability with robustness (e.g., catastrophic forgetting mitigation). (iii) Anomaly Detection: Real-time monitoring of learning trajectories to identify and neutralize hazardous knowledge acquisition.

(3) **Safety of Embodied Agents in Domain-Specific Scenarios:** As autonomous agents become increasingly deployed across specialized domains, their safety considerations must account for unique domain-specific vulnerabilities. We list some key challenges as follows:

- **Web Agents**:
  - HTML/JS injection risks during automated browsing
  - Secure sandboxing requirements for DOM manipulation
  - Cross-site scripting (XSS) vulnerabilities in automated form-filling
- **Communication Agents**:
  - Protocol-level attacks (e.g., SIP flooding, WebRTC exploits)
  - End-to-end encryption requirements for sensitive dialogues
  - Authentication bypass in voice-based agents
- **Robotics Control Agents**:
  - Physical safety constraints in actuator commands
  - Real-time collision avoidance verification
  - Emergency stop mechanism reliability
- **Healthcare Agents**:
  - Medical decision audit trail requirements
  - Drug interaction verification systems

7 Safety in LLM-Based Application

In this section, we focus on the security considerations that should be addressed following the commercialization of LLMs into practical applications. With the rapid development of LLMs in fields such as content creation, intelligent interaction, automated programming, medical diagnosis, and financial analysis, LLM-based applications are reshaping industry workflows and business models [944]. However, while LLMs significantly enhance productivity and facilitate human-machine collaboration, their large-scale deployment has also introduced severe security challenges [66]. Ensuring the security, reliability, and compliance of LLM-based applications has become a critical issue in AI research and real-world implementation.

Truthfulness. Despite their powerful text generation capabilities, LLMs exhibit hallucination phenomena, generating inaccurate, misleading, or entirely fictitious content [945, 946, 947, 948, 949]. Unlike traditional errors, hallucinations are often subtle and linguistically plausible, making them especially dangerous in real-world applications. This challenge is exacerbated in high-stakes domains such as healthcare, law, and finance, where misleading AI-generated information can directly affect human safety and economic

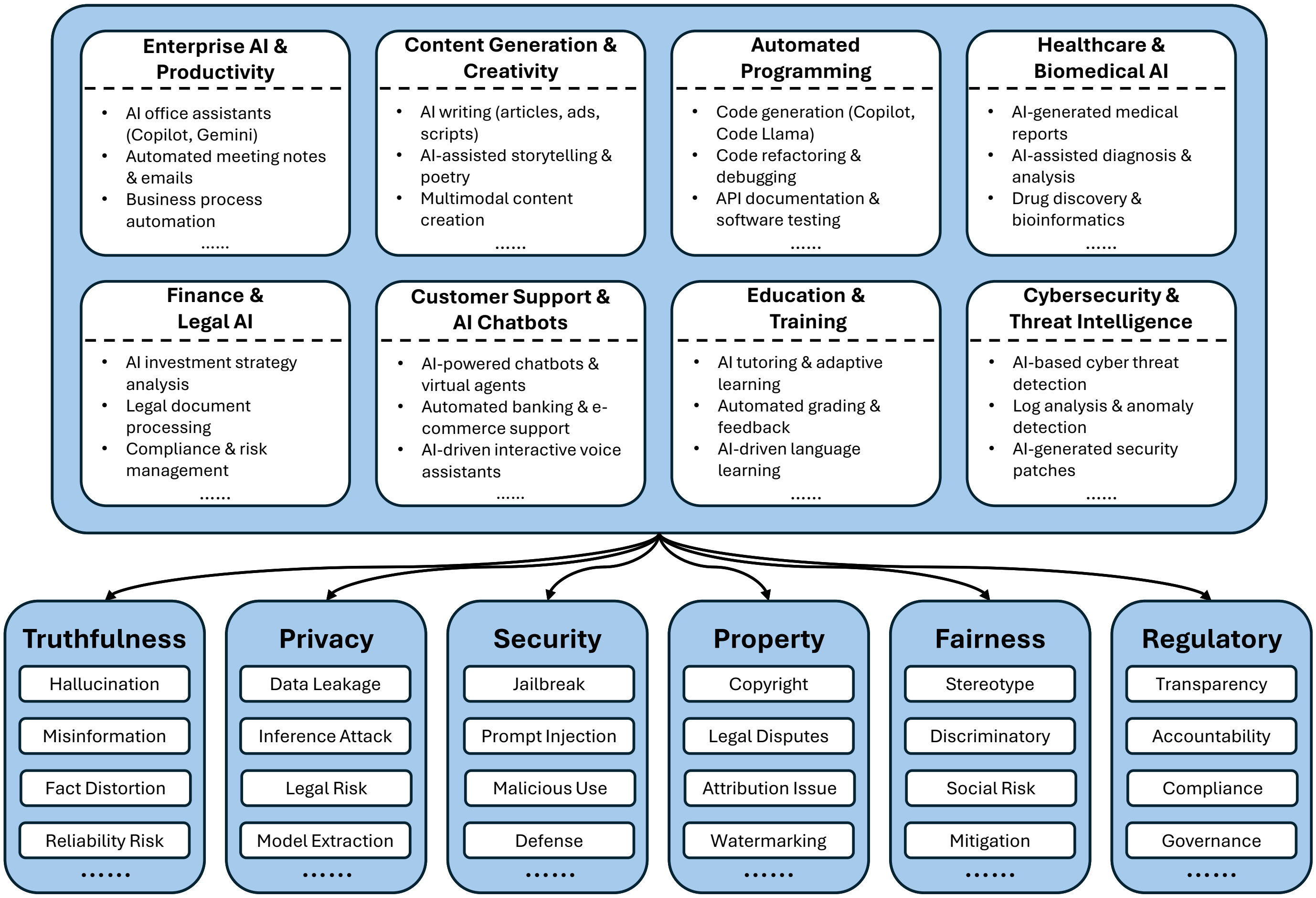
Enterprise AI & Productivity
• AI office assistants (Copilot, Gemini)
• Automated meeting notes & emails
• Business process automation
......
Content Generation & Creativity
• AI writing (articles, ads, scripts)
• AI-assisted storytelling & poetry
• Multimodal content creation
......
Automated Programming
• Code generation (Copilot, Code Llama)
• Code refactoring & debugging
• API documentation & software testing
......
Healthcare & Biomedical AI
• AI-generated medical reports
• AI-assisted diagnosis & analysis
• Drug discovery & bioinformatics
......
Finance & Legal AI
• AI investment strategy analysis
• Legal document processing
• Compliance & risk management
......
Customer Support & AI Chatbots
• AI-powered chatbots & virtual agents
• Automated banking & e-commerce support
• AI-driven interactive voice assistants
......
Education & Training
• AI tutoring & adaptive learning
• Automated grading & feedback
• AI-driven language learning
......
Cybersecurity & Threat Intelligence
• AI-based cyber threat detection
• Log analysis & anomaly detection
• AI-generated security patches
......
Truthfulness
Hallucination
Misinformation
Fact Distortion
Reliability Risk
......
Privacy
Data Leakage
Inference Attack
Legal Risk
Model Extraction
......
Security
Jailbreak
Prompt Injection
Malicious Use
Defense
......
Property
Copyright
Legal Disputes
Attribution Issue
Watermarking
......
Fairness
Stereotype
Discriminatory
Social Risk
Mitigation
......
Regulatory
Transparency
Accountability
Compliance
Governance
......

Fig. 14: We illustrate the diverse applications of AI in enterprise productivity, content generation, programming, healthcare, finance, customer support, education, and cyber-security. We also highlight critical issues related to truthfulness and privacy, including data leakage, security threats, property rights, fairness, and regulatory compliance, underscoring the need for robust safeguards in AI deployment

stability. For example, an LLM-powered clinical assistant may suggest nonexistent diseases or cite unverified treatments, posing risks to patients [739, 950], while financial advisors powered by LLMs might generate persuasive but flawed market forecasts, leading to significant capital misallocation or systemic financial vulnerabilities [951]. Specifically, hallucination is not merely a surface-level output flaw but a systemic artifact rooted in the model's training dynamics and the nature of its data. Specifically, hallucination can stem from three compounding factors: (1) semantic overgeneralization due to exposure to noisy, unverified, or synthetic pretraining corpora; (2) objective misalignment, where maximum-likelihood or reinforcement-based training prioritizes coherence and helpfulness over factual accuracy; and (3) latent distribution shifts between pretraining and deployment-time inputs, particularly under long-tail or adversarial queries [952, 953]. These factors jointly reinforce spurious correlations and amplify unsupported generations, even in otherwise well-aligned models. In sum, hallucination represents a critical bottleneck for the reliable deployment of LLMs. Its mitigation is foundational not only for improving user trust but also for enabling the safe integration of LLMs into high-stakes decision-making workflows.

Privacy. Data privacy concerns [954] represent another significant challenge in LLM deployment [821, 955]. Training these models requires vast amounts of text data, which may include personal information, corporate secrets, and medical records [956]. If an LLM inadvertently leaks sensitive training data or lacks robust access control mechanisms, users' private information could be exploited or misused. In corporate settings, LLMs may unintentionally expose confidential documents or sensitive customer data, leading to severe compliance and legal risks. Moreover, inference-time attacks [957], such as membership inference and model extraction, can further expose sensitive data by allowing adversaries to infer training set membership or replicate model behavior. Therefore, LLM-based applications must incorporate data protection measures and privacy-preserving techniques like differential privacy and query rate limiting to mitigate information leakage risks.

Robustness. Prompt injection [543] and jailbreak [636] risks pose additional security threats. Attackers can craft adversarial prompts to bypass security restrictions, causing the model to generate harmful or unauthorized content. For example, in chatbot systems, malicious users could manipulate LLMs to generate hate speech, disinformation, or even harmful instructions. Similarly, in AI-powered coding assistants such as GitHub Copilot, attackers may exploit LLMs to produce code with security vulnerabilities, potentially serving as backdoors for future cyberattacks. Developing robust security defenses to prevent LLMs from being misused in real-world applications is crucial for AI safety.

Copyright. Another pressing concern is intellectual property and copyright protection [958, 959, 960]. LLMs are trained on vast datasets that often include copyrighted texts, source code, and artistic works, raising potential infringement risks. When generating content, LLMs may inadvertently replicate or closely mimic copyrighted material, leading to legal disputes. For instance, AI-powered writing tools might generate articles resembling published works, while coding assistants could produce open-source code snippets without proper licensing [961]. This not only raises concerns about content originality but also introduces legal and ethical dilemmas. Addressing these challenges requires watermarking [962, 963], provenance tracking, and clear copyright attribution mechanisms to ensure responsible AI-generated content management [178].

Ethical and Social Responsibility. Beyond technical concerns, ethical and social responsibility are also critical factors in large-scale LLM deployment. Due to biases in training data, LLMs may generate content that reinforces stereotypes, gender discrimination, or racial biases [964, 965]. In sectors such as hiring, finance, and healthcare, biased AI-generated recommendations could exacerbate existing inequalities and lead to unfair decision-making. Moreover, as LLMs become increasingly integrated into virtual assistants, social media, and news distribution platforms, concerns over AI-generated misinformation, transparency, and accountability are growing. Building fair, transparent, and trustworthy AI governance frameworks is thus essential to mitigating AI-induced social risks.

Governance. As governments worldwide strengthen AI regulations, LLM-related legal and compliance requirements are evolving rapidly. The EU AI Act classifies LLMs as high-risk AI systems, requiring developers to provide transparency reports and risk control mechanisms [966]. China's Generative AI Regulations mandate AI-generated content to align with ethical standards and undergo governmental scrutiny [967]. In the United States, regulatory discussions emphasize AI transparency and data privacy protections, urging businesses to establish responsible AI practices [968]. These policy developments indicate that LLM-based applications must comply with regional regulations while maintaining a balance between compliance and innovation.

In summary, while LLM-based applications drive technological progress, they also introduce multifaceted challenges related to misinformation, data privacy, adversarial manipulation, copyright infringement, ethical concerns, and regulatory compliance (refer to Figure 14). These issues not only impact the trustworthiness and legality of AI technologies but also have far-reaching implications for social trust, legal accountability, and business sustainability. Addressing these challenges necessitates a comprehensive approach that integrates privacy protection, content governance, copyright management, ethical safeguards, and regulatory compliance, alongside collaborative efforts from both academia and industry.

8 Potential Research Directions

Through a systematic and comprehensive examination of safety across the entire lifecycle of LLMs, we have identified valuable insights for future research:

★ Data generation holds immense potential, particularly in ensuring the safety of generated data and automating the data generation process, which is crucial for reliable and robust model training. Reliable data generation is fundamental to the integrity of model training.

★ Post-training phases are becoming increasingly critical. Ensuring secure fine-tuning and alignment of data is a key future direction, closely intertwined with data generation. As concepts proliferate, multi-objective alignment may emerge as a significant area of focus.

★ Model editing and unlearning safety are paramount for efficient model updates and deployment. Current learning efficiencies are suboptimal, and advancements in these technologies could revolutionize how models acquire new knowledge, enabling continuous and efficient learning (potentially even localized memory learning). These techniques might surpass traditional SGD algorithms, but safety measures are essential to prevent models from devolving into malicious entities that contradict human intentions.

★ LLM agents, in the final deployment stage, require robust safety assurances. Ensuring the security of agent tools and agent memory, as well as addressing safety in embodied intelligence scenarios such as web agents and computer agents, are critical areas for further investigation.

9 Conclusion

In this survey, we provide a comprehensive analysis of the safety concerns across the entire lifecycle of LLMs, from data preparation and pre-training to post-training, deployment, and commercialization. By introducing the concept of "full-stack" safety, we offer an integrated view of the security and safety issues faced by LLMs throughout their development and usage, which addresses gaps in the existing literature that typically focus on specific stages of the lifecycle.

Through an exhaustive review of over 900+ papers, we systematically examined and organized the safety issues spanning key stages of LLM production, deployment, and use, including data generation, alignment techniques, model editing, and LLM-based agent systems and LLM-based applications. Our findings highlight the critical vulnerabilities at each stage, such as privacy risks, toxic data, harmful fine-tuning attacks, and deployment challenges. The safety of LLMs is a multifaceted issue requiring careful attention to data integrity, model alignment, and post-deployment security measures. Moreover, we propose promising directions for future research, including improvements in data safety, alignment techniques, and defense mechanisms for LLM-based agents. This work is vital for guiding future efforts to make LLMs safer and more reliable, especially as they become increasingly integral to various industries and applications. Ensuring robust security across the entire LLM lifecycle is crucial for their responsible and effective deployment in real-world scenarios.

REFERENCES

[1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[2] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

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

[4] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

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

[6] 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," *arXiv preprint arXiv:2303.18223*, vol. 1, no. 2, 2023.

[7] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM transactions on intelligent systems and technology*, vol. 15, no. 3, pp. 1–45, 2024.

[8] 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," *Authorea Preprints*, vol. 3, 2023.

[9] Y. Yan, S. Wang, J. Huo, J. Ye, Z. Chu, X. Hu, P. S. Yu, C. Gomes, B. Selman, and Q. Wen, "Position: Multimodal large language models can significantly advance scientific reasoning," *arXiv preprint arXiv:2502.02871*, 2025.

[10] Y. Yan, J. Su, J. He, F. Fu, X. Zheng, Y. Lyu, K. Wang, S. Wang, Q. Wen, and X. Hu, "A survey of mathematical reasoning in the era of multimodal large language model: Benchmark, method & challenges," *arXiv preprint arXiv:2412.11936*, 2024.

[11] X. Zou, Y. Yan, X. Hao, Y. Hu, H. Wen, E. Liu, J. Zhang, Y. Li, T. Li, Y. Zheng *et al.*, "Deep learning for cross-domain data fusion in urban computing: Taxonomy, advances, and outlook," *Information Fusion*, vol. 113, p. 102606, 2025.

[12] Y. Li, X. Zhang, L. Luo, H. Chang, Y. Ren, I. King, and J. Li, "G-refer: Graph retrieval-augmented large language model for explainable recommendation," *arXiv preprint arXiv:2502.12586*, 2025.

[13] S. Sun, R. Liu, J. Lyu, J.-W. Yang, L. Zhang, and X. Li, "A large language model-driven reward design framework via dynamic feedback for reinforcement learning," *arXiv preprint arXiv:2410.14660*, 2024.

[14] S. Sonko, A. O. Adewusi, O. C. Obi, S. Onwusinkwue, and A. Atadoga, "A critical review towards artificial general intelligence: Challenges, ethical considerations, and the path forward," *World Journal of Advanced Research and Reviews*, vol. 21, no. 3, pp. 1262–1268, 2024.

[15] S. McLean, G. J. Read, J. Thompson, C. Baber, N. A. Stanton, and P. M. Salmon, "The risks associated with artificial general intelligence: A systematic review," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 35, no. 5, pp. 649–663, 2023.

[16] R. Liu, J. Gao, J. Zhao, K. Zhang, X. Li, B. Qi, W. Ouyang, and B. Zhou, "Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling," *arXiv preprint arXiv:2502.06703*, 2025.

[17] J. Ruan, Y. Chen, B. Zhang, Z. Xu, T. Bao, H. Mao, Z. Li, X. Zeng, R. Zhao *et al.*, "Tptu: Task planning and tool usage of large language model-based ai agents," in *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

[18] V. Sorin, E. Klang, M. Sklair-Levy, I. Cohen, D. B. Zippel, N. Balint Lahat, E. Konen, and Y. Barash, "Large language model (chatgpt) as a support tool for breast tumor board," *NPJ Breast Cancer*, vol. 9, no. 1, p. 44, 2023.

[19] R. Yang, L. Song, Y. Li, S. Zhao, Y. Ge, X. Li, and Y. Shan, "Gpt4tools: Teaching large language model to use tools via self-instruction," *Advances in Neural Information Processing Systems*, vol. 36, pp. 71 995–72 007, 2023.

[20] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," *Advances in Neural Information Processing Systems*, vol. 36, pp. 68 539–68 551, 2023.

[21] W. Zhong, L. Guo, Q. Gao, H. Ye, and Y. Wang, "Memorybank: Enhancing large language models with long-term memory," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 724–19 731.

[22] W. Wang, L. Dong, H. Cheng, X. Liu, X. Yan, J. Gao, and F. Wei, "Augmenting language models with long-term memory," *Advances in Neural Information Processing Systems*, vol. 36, pp. 74 530–74 543, 2023.

[23] Z. Zhang, X. Bo, C. Ma, R. Li, X. Chen, Q. Dai, J. Zhu, Z. Dong, and J.-R. Wen, "A survey on the memory mechanism of large language model based agents," *arXiv preprint arXiv:2404.13501*, 2024.

[24] J. Huo, Y. Yan, B. Hu, Y. Yue, and X. Hu, "Mmneuron: Discovering neuron-level domain-specific interpretation in multimodal large language model," *arXiv preprint arXiv:2406.11193*, 2024.

[25] W. Liu, X. Huang, X. Zeng, X. Hao, S. Yu, D. Li, S. Wang, W. Gan, Z. Liu, Y. Yu *et al.*, "Toolace: Win-

ning the points of llm function calling," *arXiv preprint arXiv:2409.00920*, 2024.

[26] Q. Tang, Z. Deng, H. Lin, X. Han, Q. Liang, B. Cao, and L. Sun, "Toolalpaca: Generalized tool learning for language models with 3000 simulated cases," *arXiv preprint arXiv:2306.05301*, 2023.

[27] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, and X. Zhang, "Large language model based multi-agents: A survey of progress and challenges," *arXiv preprint arXiv:2402.01680*, 2024.

[28] 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," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[29] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou *et al.*, "The rise and potential of large language model based agents: A survey," *Science China Information Sciences*, vol. 68, no. 2, p. 121101, 2025.

[30] Y. Yan and J. Lee, "Georeasoner: Reasoning on geospatially grounded context for natural language understanding," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 4163–4167.

[31] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" *Advances in Neural Information Processing Systems*, vol. 36, pp. 655–677, 2023.

[32] M. Zhou, H. Dong, H. Song, N. Zheng, W.-H. Chen, and H. Wang, "Embodied intelligence-based perception, decision-making, and control for autonomous operations of rail transportation," *IEEE Transactions on Intelligent Vehicles*, 2024.

[33] X. Ma, Y. Gao, Y. Wang, R. Wang, X. Wang, Y. Sun, Y. Ding, H. Xu, Y. Chen, Y. Zhao *et al.*, "Safety at scale: A comprehensive survey of large model safety," *arXiv preprint arXiv:2502.05206*, 2025.

[34] K. Kumar, T. Ashraf, O. Thawakar, R. M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, P. H. Torr, S. Khan, and F. S. Khan, "Llm post-training: A deep dive into reasoning large language models," *arXiv preprint arXiv:2502.21321*, 2025.

[35] Z.-Z. Li, D. Zhang, M.-L. Zhang, J. Zhang, Z. Liu, Y. Yao, H. Xu, J. Zheng, P.-J. Wang, X. Chen *et al.*, "From system 1 to system 2: A survey of reasoning large language models," *arXiv preprint arXiv:2502.17419*, 2025.

[36] Y. Chen, W. Sun, C. Fang, Z. Chen, Y. Ge, T. Han, Q. Zhang, Y. Liu, Z. Chen, and B. Xu, "Security of language models for code: A systematic literature review," *ACM Transactions on Software Engineering and Methodology*, vol. 1, no. 1, pp. 1–66, 2025.

[37] W. Qu, Y. Zhou, Y. Wu, T. Xiao, B. Yuan, Y. Li, and J. Zhang, "Prompt inversion attack against collaborative inference of large language models," in *IEEE S&P*, 2025.

[38] J. Wu, S. Yang, R. Zhan, Y. Yuan, L. S. Chao, and D. F. Wong, "A survey on llm-generated text detection: Necessity, methods, and future directions," *Computational Linguistics*, pp. 1–66, 2025.

[39] H. Wang, J. Li, H. Wu, E. Hovy, and Y. Sun, "Pretrained language models and their applications," *Engineering*, vol. 25, pp. 51–65, 2023.

[40] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He *et al.*, "A comprehensive survey on pretrained foundation models: A history from bert to chatgpt," *International Journal of Machine Learning and Cybernetics*, pp. 1–65, 2024.

[41] X. Zhang, X. Zhu, and L. Lessard, "Online data poisoning attacks," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 201–210.

[42] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Mądry, B. Li, and T. Goldstein, "Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 1563–1580, 2022.

[43] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin, "Analyzing leakage of personally identifiable information in language models," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 346–363.

[44] W. Sun, Y. Chen, C. Fang, Y. Feng, Y. Xiao, A. Guo, Q. Zhang, Y. Liu, B. Xu, and Z. Chen, "Eliminating backdoors in neural code models for secure code understanding," in *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*. Trondheim, Norway: ACM, Mon 23 - Fri 27 June 2025, pp. 1–23.

[45] H. R. Kirk, B. Vidgen, P. Röttger, and S. A. Hale, "The benefits, risks and bounds of personalizing the alignment of large language models to individuals," *Nature Machine Intelligence*, vol. 6, no. 4, pp. 383–392, 2024.

[46] Z. Zhou, H. Yu, X. Zhang, R. Xu, F. Huang, and Y. Li, "How alignment and jailbreak work: Explain llm safety through intermediate hidden states," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 2461–2488.

[47] X. Qi, Y. Zeng, T. Xie, P.-Y. Chen, R. Jia, P. Mittal, and P. Henderson, "Fine-tuning aligned language models compromises safety, even when users do not intend to!" in *ICLR*, 2024. [Online]. Available: https://openreview.net/forum?id=hTEGyKf0dZ

[48] X. Qi, A. Panda, K. Lyu, X. Ma, S. Roy, A. Beirami, P. Mittal, and P. Henderson, "Safety alignment should be made more than just a few tokens deep," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=6Mxhg9PtDE

[49] D. Halawi, A. Wei, E. Wallace, T. T. Wang, N. Haghtalab, and J. Steinhardt, "Covert malicious finetuning: Challenges in safeguarding LLM adaptation," in *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024, pp. 17 298–17 312.

[50] W. Hawkins, B. Mittelstadt, and C. Russell, "The effect of fine-tuning on language model toxicity," in *Neurips Safe Generative AI Workshop 2024*, 2024.

[51] J. Huang and J. Zhang, "A survey on evaluation of

multimodal large language models," *arXiv preprint arXiv:2408.15769*, 2024.

[52] P. Röttger, F. Pernisi, B. Vidgen, and D. Hovy, "Safetyprompts: a systematic review of open datasets for evaluating and improving large language model safety," *arXiv preprint arXiv:2404.05399*, 2024.

[53] Y. Dong, R. Mu, Y. Zhang, S. Sun, T. Zhang, C. Wu, G. Jin, Y. Qi, J. Hu, J. Meng *et al.*, "Safeguarding large language models: A survey," *arXiv preprint arXiv:2406.02622*, 2024.

[54] Y. Wang, Y. Pan, Q. Zhao, Y. Deng, Z. Su, L. Du, and T. H. Luan, "Large model agents: State-of-the-art, cooperation paradigms, security and privacy, and future trends," *arXiv preprint arXiv:2409.14457*, 2024.

[55] G. Zhang, K. Chen, G. Wan, H. Chang, H. Cheng, K. Wang, S. Hu, and L. Bai, "Evoflow: Evolving diverse agentic workflows on the fly," *arXiv preprint arXiv:2502.07373*, 2025.

[56] G. Zhang, L. Niu, J. Fang, K. Wang, L. Bai, and X. Wang, "Multi-agent architecture search via agentic supernet," *arXiv preprint arXiv:2502.04180*, 2025.

[57] G. Zhang, Y. Yue, Z. Li, S. Yun, G. Wan, K. Wang, D. Cheng, J. X. Yu, and T. Chen, "Cut the crap: An economical communication pipeline for llm-based multi-agent systems," *arXiv preprint arXiv:2410.02506*, 2024.

[58] Y. Yue, G. Zhang, B. Liu, G. Wan, K. Wang, D. Cheng, and Y. Qi, "Masrouter: Learning to route llms for multi-agent systems," 2025. [Online]. Available: https://arxiv.org/abs/2502.11133

[59] Z. Liang, Y. Xu, Y. Hong, P. Shang, Q. Wang, Q. Fu, and K. Liu, "A survey of multimodel large language models," in *Proceedings of the 3rd International Conference on Computer, Artificial Intelligence and Control Engineering*, 2024, pp. 405–409.

[60] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu *et al.*, "Instruction tuning for large language models: A survey," *arXiv preprint arXiv:2308.10792*, 2023.

[61] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du, "Explainability for large language models: A survey," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 2, pp. 1–38, 2024.

[62] T. Shen, R. Jin, Y. Huang, C. Liu, W. Dong, Z. Guo, X. Wu, Y. Liu, and D. Xiong, "Large language model alignment: A survey," *arXiv preprint arXiv:2309.15025*, 2023.

[63] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib, M. M. J. Mim, J. Ahmad, M. E. Ali, and S. Azam, "A review on large language models: Architectures, applications, taxonomies, open issues and challenges," *IEEE access*, vol. 12, pp. 26 839–26 874, 2024.

[64] K. S. Kalyan, "A survey of gpt-3 family large language models including chatgpt and gpt-4," *Natural Language Processing Journal*, vol. 6, p. 100048, 2024.

[65] E. Shayegani, M. A. A. Mamun, Y. Fu, P. Zaree, Y. Dong, and N. Abu-Ghazaleh, "Survey of vulnerabilities in large language models revealed by adversarial attacks," *arXiv preprint arXiv:2310.10844*, 2023.

[66] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (llm) security and privacy: The good, the bad, and the ugly," *High-Confidence Computing*, p. 100211, 2024.

[67] L. Qin, Q. Chen, Y. Zhou, Z. Chen, Y. Li, L. Liao, M. Li, W. Che, and P. S. Yu, "Multilingual large language model: A survey of resources, taxonomy and frontiers," *arXiv preprint arXiv:2404.04925*, 2024.

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

[69] L. Sun, Y. Huang, H. Wang, S. Wu, Q. Zhang, C. Gao, Y. Huang, W. Lyu, Y. Zhang, X. Li *et al.*, "Trustllm: Trustworthiness in large language models," *arXiv preprint arXiv:2401.05561*, vol. 3, 2024.

[70] B. C. Das, M. H. Amini, and Y. Wu, "Security and privacy challenges of large language models: A survey," *ACM Computing Surveys*, vol. 57, no. 6, pp. 1–39, 2025.

[71] F. He, T. Zhu, D. Ye, B. Liu, W. Zhou, and P. S. Yu, "The emerged security and privacy of llm agent: A survey with case studies," *arXiv preprint arXiv:2407.19354*, 2024.

[72] G. Tie, Z. Zhao, D. Song, F. Wei, R. Zhou, Y. Dai, W. Yin, Z. Yang, J. Yan, Y. Su *et al.*, "A survey on post-training of large language models," *arXiv preprint arXiv:2503.06072*, 2025.

[73] Y. Huang, C. Gao, S. Wu, H. Wang, X. Wang, Y. Zhou, Y. Wang, J. Ye, J. Shi, Q. Zhang *et al.*, "On the trustworthiness of generative foundation models: Guideline, assessment, and perspective," *arXiv preprint arXiv:2502.14296*, 2025.

[74] M. Yu, F. Meng, X. Zhou, S. Wang, J. Mao, L. Pang, T. Chen, K. Wang, X. Li, Y. Zhang *et al.*, "A survey on trustworthy llm agents: Threats and countermeasures," *arXiv preprint arXiv:2503.09648*, 2025.

[75] X. Ma, Y. Gao, Y. Wang, R. Wang, X. Wang, Y. Sun, Y. Ding, H. Xu, Y. Chen, Y. Zhao, H. Huang, Y. Li, J. Zhang, X. Zheng, Y. Bai, Z. Wu, X. Qiu, J. Zhang, Y. Li, J. Sun, C. Wang, J. Gu, B. Wu, S. Chen, T. Zhang, Y. Liu, M. Gong, T. Liu, S. Pan, C. Xie, T. Pang, Y. Dong, R. Jia, Y. Zhang, S. Ma, X. Zhang, N. Gong, C. Xiao, S. Erfani, B. Li, M. Sugiyama, D. Tao, J. Bailey, and Y.-G. Jiang, "Safety at scale: A comprehensive survey of large model safety," 2025. [Online]. Available: https://arxiv.org/abs/2502.05206

[76] Y. Huang, L. Sun, H. Wang, S. Wu, Q. Zhang, Y. Li, C. Gao, Y. Huang, W. Lyu, Y. Zhang *et al.*, "Position: Trustllm: Trustworthiness in large language models," in *International Conference on Machine Learning*. PMLR, 2024, pp. 20 166–20 270.

[77] Z. Dong, Z. Zhou, C. Yang, J. Shao, and Y. Qiao, "Attacks, defenses and evaluations for llm conversation safety: A survey," *arXiv preprint arXiv:2402.09283*, 2024.

[78] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, and J. Launay, "The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only," *arXiv preprint arXiv:2306.01116*, 2023.

[79] L. Soldaini, R. Kinney, A. Bhagia, D. Schwenk, D. Atkinson, R. Authur, B. Bogin, K. Chandu, J. Dumas, Y. Elazar *et al.*, “Dolma: An open corpus of three trillion tokens for language model pretraining research,” *arXiv preprint arXiv:2402.00159*, 2024.

[80] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy, “Challenges and applications of large language models,” *arXiv preprint arXiv:2307.10169*, 2023.

[81] W. Sun, Y. Chen, G. Tao, C. Fang, X. Zhang, Q. Zhang, and B. Luo, “Backdooring neural code search,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. Toronto, Canada: Association for Computational Linguistics, July 9-14 2023, pp. 9692–9708.

[82] W. Sun, Y. Chen, M. Yuan, C. Fan, Z. Chen, C. Wang, Y. Liu, B. Xu, and Z. Chen, “Show me your code! kill code poisoning: A lightweight method based on code naturalness,” in *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering*. Ottawa, Ontario, Canada: IEEE Computer Society, Sun 27 April - Sat 3 May 2025, pp. 1–13.

[83] N. Carlini, M. Jagielski, C. A. Choquette-Choo, D. Paleka, W. Pearce, H. Anderson, A. Terzis, K. Thomas, and F. Tramèr, “Poisoning web-scale training datasets is practical,” in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, pp. 407–425.

[84] Y. Zhang, J. Rando, I. Evtimov, J. Chi, E. M. Smith, N. Carlini, F. Tramèr, and D. Ippolito, “Persistent pre-training poisoning of llms,” *arXiv preprint arXiv:2410.13722*, 2024.

[85] E. Wallace, T. Z. Zhao, S. Feng, and S. Singh, “Concealed data poisoning attacks on nlp models,” *arXiv preprint arXiv:2010.12563*, 2020.

[86] B. Yan, K. Li, M. Xu, Y. Dong, Y. Zhang, Z. Ren, and X. Cheng, “On protecting the data privacy of large language models (llms): A survey,” *arXiv preprint arXiv:2403.05156*, 2024.

[87] N. Kandpal, E. Wallace, and C. Raffel, “Deduplicating training data mitigates privacy risks in language models,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 10 697–10 707.

[88] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, “Quantifying memorization across neural language models,” in *The Eleventh International Conference on Learning Representations*, 2022.

[89] C. Arnett, E. Jones, I. P. Yamshchikov, and P.-C. Langlais, “Toxicity of the commons: Curating open-source pre-training data,” *arXiv preprint arXiv:2410.22587*, 2024.

[90] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini, “Deduplicating training data makes language models better,” *arXiv preprint arXiv:2107.06499*, 2021.

[91] Y. Li, Y. Jiang, Z. Li, and S. Xia, “Backdoor learning: A survey.” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 5–22, 2024.

[92] Y. Zeng, M. Pan, H. Jahagirdar, M. Jin, L. Lyu, and R. Jia, “How to sift out a clean data subset in the presence of data poisoning?” *arXiv preprint arXiv:2210.06516*, 2022.

[93] M. Pan, Y. Zeng, L. Lyu, X. Lin, and R. Jia, “{ASSET}: Robust backdoor data detection across a multiplicity of deep learning paradigms,” in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 2725–2742.

[94] Z. Zhang, L. Lyu, W. Wang, L. Sun, and X. Sun, “How to inject backdoors with better consistency: Logit anchoring on clean data,” in *International Conference on Learning Representations*, 2022.

[95] Z. Zhang, L. Lyu, X. Ma, C. Wang, and X. Sun, “Fine-mixing: Mitigating backdoors in fine-tuned language models,” *arXiv preprint arXiv:2210.09545*, 2022.

[96] X. Sun, X. Li, Y. Meng, X. Ao, L. Lyu, J. Li, and T. Zhang, “Defending against backdoor attacks in natural language generation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 5257–5265.

[97] S. Longpre, G. Yauney, E. Reif, K. Lee, A. Roberts, B. Zoph, D. Zhou, J. Wei, K. Robinson, D. Mimno *et al.*, “A pretrainer’s guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity,” in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 3245–3276.

[98] S. Neel and P. Chang, “Privacy issues in large language models: A survey,” *arXiv preprint arXiv:2312.06717*, 2023.

[99] X. Wu, R. Duan, and J. Ni, “Unveiling security, privacy, and ethical concerns of chatgpt,” *Journal of Information and Intelligence*, vol. 2, no. 2, pp. 102–115, 2024.

[100] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, “From chatgpt to threatgpt: Impact of generative ai in cybersecurity and privacy,” *IEEE Access*, vol. 11, pp. 80 218–80 245, 2023.

[101] M. Miranda, E. S. Ruzzetti, A. Santilli, F. M. Zanzotto, S. Bratières, and E. Rodolà, “Preserving privacy in large language models: A survey on current threats and solutions,” *arXiv preprint arXiv:2408.05212*, 2024.

[102] Q. Zhang, H. Qiu, D. Wang, Y. Li, T. Zhang, W. Zhu, H. Weng, L. Yan, and C. Zhang, “A benchmark for semantic sensitive information in llms outputs,” in *The Thirteenth International Conference on Learning Representations*, 2025.

[103] S. Kim, S. Yun, H. Lee, M. Gubri, S. Yoon, and S. J. Oh, “Propile: C,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 20 750–20 762, 2023.

[104] H. Li, D. Guo, W. Fan, M. Xu, J. Huang, F. Meng, and Y. Song, “Multi-step jailbreaking privacy attacks on chatgpt,” *arXiv preprint arXiv:2304.05197*, 2023.

[105] M. S. Ozdayi, C. Peris, J. FitzGerald, C. Dupuy, J. Majmudar, H. Khan, R. Parikh, and R. Gupta, “Controlling the extraction of memorized data from large language models via prompt-tuning,” *arXiv preprint arXiv:2305.11759*, 2023.

[106] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, “The secret sharer: Evaluating and testing unintended memorization in neural networks,” in *28th USENIX security symposium (USENIX security 19)*, 2019, pp. 267–284.

[107] M. Nasr, N. Carlini, J. Hayase, M. Jagielski, A. F. Cooper, D. Ippolito, C. A. Choquette-Choo, E. Wallace,

F. Tramèr, and K. Lee, "Scalable extraction of training data from (production) language models," *arXiv preprint arXiv:2311.17035*, 2023.

[108] 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.

[109] Y. Bai, G. Pei, J. Gu, Y. Yang, and X. Ma, "Special characters attack: Toward scalable training data extraction from large language models," *arXiv preprint arXiv:2405.05990*, 2024.

[110] Z. Zhou, J. Xiang, C. Chen, and S. Su, "Quantifying and analyzing entity-level memorization in large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 741–19 749.

[111] X. Yang, Z. Wen, W. Qu, Z. Chen, Z. Xiang, B. Chen, and H. Yao, "Memorization and privacy risks in domain-specific large language models," in *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*, 2024.

[112] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[113] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, 2022.

[114] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *2022 IEEE symposium on security and privacy (SP)*. IEEE, 2022, pp. 1897–1914.

[115] J. Ye, A. Maddi, S. K. Murakonda, V. Bindschaedler, and R. Shokri, "Enhanced membership inference attacks against machine learning models," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 3093–3106.

[116] J. Zhang, D. Das, G. Kamath, and F. Tramèr, "Membership inference attacks cannot prove that a model was trained on your data," *arXiv preprint arXiv:2409.19798*, 2024.

[117] M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, and H. Hajishirzi, "Do membership inference attacks work on large language models?" *arXiv preprint arXiv:2402.07841*, 2024.

[118] M. Meeus, I. Shilov, S. Jain, M. Faysse, M. Rei, and Y.-A. de Montjoye, "Sok: Membership inference attacks on llms are rushing nowhere (and how to fix it)," *arXiv preprint arXiv:2406.17975*, 2024.

[119] Y. He, B. Li, Y. Wang, M. Yang, J. Wang, H. Hu, and X. Zhao, "Is difficulty calibration all we need? towards more practical membership inference attacks," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 1226–1240.

[120] Y. He, B. Li, L. Liu, Z. Ba, W. Dong, Y. Li, Z. Qin, K. Ren, and C. Chen, "Towards label-only membership inference attack against pre-trained large language models," in *USENIX Security*, 2025.

[121] J. Ren, K. Chen, C. Chen, V. Sehwag, Y. Xing, J. Tang, and L. Lyu, "Self-comparison for dataset-level membership inference in large (vision-) language model," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 910–920.

[122] A. Albalak, Y. Elazar, S. M. Xie, S. Longpre, N. Lambert, X. Wang, N. Muennighoff, B. Hou, L. Pan, H. Jeong *et al.*, "A survey on data selection for language models," *arXiv preprint arXiv:2402.16827*, 2024.

[123] P. Maini, S. Goyal, D. Sam, A. Robey, Y. Savani, Y. Jiang, A. Zou, Z. C. Lipton, and J. Z. Kolter, "Safety pretraining: Toward the next generation of safe ai," *arXiv preprint arXiv:2504.16980*, 2025.

[124] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[125] S. Li, F. Liu, L. Cui, J. Lu, Q. Xiao, X. Yang, P. Liu, K. Sun, Z. Ma, and X. Wang, "Safe planner: Empowering safety awareness in large pre-trained models for robot task planning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 14, 2025, pp. 14 619–14 627.

[126] J. O'Neill, S. Subramanian, E. Lin, A. Satish, and V. Mugunthan, "Guardformer: Guardrail instruction pretraining for efficient safeguarding," in *Neurips Safe Generative AI Workshop 2024*.

[127] T. Huang, S. Hu, F. Ilhan, S. F. Tekin, and L. Liu, "Harmful fine-tuning attacks and defenses for large language models: A survey," *arXiv preprint arXiv:2409.18169*, 2024.

[128] M. Shu, J. Wang, C. Zhu, J. Geiping, C. Xiao, and T. Goldstein, "On the exploitability of instruction tuning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 61 836–61 856, 2023.

[129] J. Xu, M. D. Ma, F. Wang, C. Xiao, and M. Chen, "Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models," *arXiv preprint arXiv:2305.14710*, 2023.

[130] J. Yan, V. Yadav, S. Li, L. Chen, Z. Tang, H. Wang, V. Srinivasan, X. Ren, and H. Jin, "Backdooring instruction-tuned large language models with virtual prompt injection," *arXiv preprint arXiv:2307.16888*, 2023.

[131] H. Yao, J. Lou, and Z. Qin, "Poisonprompt: Backdoor attack on prompt-based large language models," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 7745–7749.

[132] S. Zhao, J. Wen, L. A. Tuan, J. Zhao, and J. Fu, "Prompt as triggers for backdoor attack: Examining the vulnerability in language models," *arXiv preprint arXiv:2305.01219*, 2023.

[133] Z. Han, C. Gao, J. Liu, J. Zhang, and S. Q. Zhang, "Parameter-efficient fine-tuning for large models: A comprehensive survey," *arXiv preprint arXiv:2403.14608*, 2024.

[134] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, and F. L. Wang, "Parameter-efficient fine-tuning methods for pretrained language models: A critical review and

assessment," *arXiv preprint arXiv:2312.12148*, 2023.

[135] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, vol. 5, no. 3, pp. 220–235, 2023.

[136] S. Zhao, L. Gan, L. A. Tuan, J. Fu, L. Lyu, M. Jia, and J. Wen, "Defending against weight-poisoning backdoor attacks for parameter-efficient fine-tuning," *arXiv preprint arXiv:2402.12168*, 2024.

[137] J. Kim, M. Song, S. H. Na, and S. Shin, "Obliviate: Neutralizing task-agnostic backdoors within the parameter-efficient fine-tuning paradigm," *arXiv preprint arXiv:2409.14119*, 2024.

[138] S. Jiang, S. R. Kadhe, Y. Zhou, F. Ahmed, L. Cai, and N. Baracaldo, "Turning generative models degenerate: The power of data poisoning attacks," *arXiv preprint arXiv:2407.12281*, 2024.

[139] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[140] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[141] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[142] Z. Wang, Z. Shen, Y. He, G. Sun, H. Wang, L. Lyu, and A. Li, "Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations," *arXiv preprint arXiv:2409.05976*, 2024.

[143] C. Chen, X. Feng, Y. Li, L. Lyu, J. Zhou, X. Zheng, and J. Yin, "Integration of large language models and federated learning," *Patterns*, vol. 5, no. 12, 2024.

[144] W. Zhuang, C. Chen, and L. Lyu, "When foundation model meets federated learning: Motivations, challenges, and future directions," *arXiv preprint arXiv:2306.15546*, 2023.

[145] G. Sun, Y. Cong, J. Dong, Q. Wang, L. Lyu, and J. Liu, "Data poisoning attacks on federated machine learning," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11 365–11 375, 2021.

[146] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, and P. S. Yu, "Privacy and robustness in federated learning: Attacks and defenses," *IEEE transactions on neural networks and learning systems*, vol. 35, no. 7, pp. 8726–8746, 2022.

[147] R. Ye, J. Chai, X. Liu, Y. Yang, Y. Wang, and S. Chen, "Emerging safety attack and defense in federated instruction tuning of large language models," *arXiv preprint arXiv:2406.10630*, 2024.

[148] Z. Zhang, A. Panda, L. Song, Y. Yang, M. Mahoney, P. Mittal, R. Kannan, and J. Gonzalez, "Neurotoxin: Durable backdoors in federated learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 26 429–26 446.

[149] T. Fu, M. Sharma, P. Torr, S. B. Cohen, D. Krueger, and F. Barez, "Poisonbench: Assessing large language model vulnerability to data poisoning," *arXiv preprint arXiv:2410.08811*, 2024.

[150] P. Pathmanathan, S. Chakraborty, X. Liu, Y. Liang, and F. Huang, "Is poisoning a real threat to llm alignment? maybe more so than you think," *arXiv preprint arXiv:2406.12091*, 2024.

[151] A. Wan, E. Wallace, S. Shen, and D. Klein, "Poisoning language models during instruction tuning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 35 413–35 425.

[152] J. Rando and F. Tramèr, "Universal jailbreak backdoors from poisoned human feedback," *arXiv preprint arXiv:2311.14455*, 2023.

[153] T. Baumgärtner, Y. Gao, D. Alon, and D. Metzler, "Best-of-venom: Attacking rlhf by injecting poisoned preference data," *arXiv preprint arXiv:2404.05530*, 2024.

[154] B. Chen, H. Guo, G. Wang, Y. Wang, and Q. Yan, "The dark side of human feedback: Poisoning large language models via user inputs," *arXiv preprint arXiv:2409.00787*, 2024.

[155] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan *et al.*, "Training a helpful and harmless assistant with reinforcement learning from human feedback," *arXiv preprint arXiv:2204.05862*, 2022.

[156] H. Dong, W. Xiong, B. Pang, H. Wang, H. Zhao, Y. Zhou, N. Jiang, D. Sahoo, C. Xiong, and T. Zhang, "Rlhf workflow: From reward modeling to online rlhf," *arXiv preprint arXiv:2405.07863*, 2024.

[157] W. Xiong, H. Dong, C. Ye, Z. Wang, H. Zhong, H. Ji, N. Jiang, and T. Zhang, "Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint," *arXiv preprint arXiv:2312.11456*, 2023.

[158] H. Lee, S. Phatale, H. Mansoor, K. R. Lu, T. Mesnard, J. Ferret, C. Bishop, E. Hall, V. Carbune, and A. Rastogi, "Rlaif: Scaling reinforcement learning from human feedback with ai feedback," 2023.

[159] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems*, vol. 36, pp. 53 728–53 741, 2023.

[160] J. Wang, J. Wu, M. Chen, Y. Vorobeychik, and C. Xiao, "Rlhfpoison: Reward poisoning attack for reinforcement learning with human feedback in large language models," *arXiv preprint arXiv:2311.09641*, 2023.

[161] S. Gunasekar, Y. Zhang, J. Aneja, C. C. T. Mendes, A. Del Giorno, S. Gopi, M. Javaheripi, P. Kauffmann, G. de Rosa, O. Saarikivi *et al.*, "Textbooks are all you need," *arXiv preprint arXiv:2306.11644*, 2023.

[162] Y. Li, S. Bubeck, R. Eldan, A. Del Giorno, S. Gunasekar, and Y. T. Lee, "Textbooks are all you need ii: phi-1.5 technical report," *arXiv preprint arXiv:2309.05463*, 2023.

[163] J. Zhan, J. Dai, J. Ye, Y. Zhou, D. Zhang, Z. Liu, X. Zhang, R. Yuan, G. Zhang, L. Li *et al.*, "Anygpt: Unified multimodal llm with discrete sequence modeling," *arXiv preprint arXiv:2402.12226*, 2024.

[164] H. Wang, C. Liu, N. Xi, Z. Qiang, S. Zhao, B. Qin, and T. Liu, "Huatuo: Tuning llama model with chinese medical knowledge," *arXiv preprint arXiv:2304.06975*, 2023.

[165] P. Sutanto, J. Santoso, E. I. Setiawan, and A. P. Wibawa, "Llm distillation for efficient few-shot multiple choice question answering," *arXiv preprint arXiv:2412.09807*, 2024.

[166] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang, "Distilling mathematical reasoning capabilities into small language models," *Neural Networks*, vol. 179, p. 106594, 2024.

[167] R. Xu, H. Cui, Y. Yu, X. Kan, W. Shi, Y. Zhuang, W. Jin, J. Ho, and C. Yang, "Knowledge-infused prompting: Assessing and advancing clinical text data generation with large language models," *arXiv preprint arXiv:2311.00287*, 2023.

[168] N. Crispino, K. Montgomery, F. Zeng, D. Song, and C. Wang, "Agent instructs large language models to be general zero-shot reasoners," *arXiv preprint arXiv:2310.03710*, 2023.

[169] C. Li, C. Zhang, Y. Lu, J. Zhang, Q. Sun, X. Wang, J. Wei, G. Wang, Y. Yang, and H. T. Shen, "Syzygy of thoughts: Improving llm cot with the minimal free resolution," *arXiv preprint arXiv:2504.09566*, 2025.

[170] Z. Chen, K. Liu, Q. Wang, W. Zhang, J. Liu, D. Lin, K. Chen, and F. Zhao, "Agent-flan: Designing data and methods of effective agent tuning for large language models," *arXiv preprint arXiv:2403.12881*, 2024.

[171] C. Xu, Q. Sun, K. Zheng, X. Geng, P. Zhao, J. Feng, C. Tao, and D. Jiang, "Wizardlm: Empowering large language models to follow complex instructions," *arXiv preprint arXiv:2304.12244*, 2023.

[172] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi, and A. Awadallah, "Orca: Progressive learning from complex explanation traces of gpt-4," *arXiv preprint arXiv:2306.02707*, 2023.

[173] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language models with self-generated instructions," *arXiv preprint arXiv:2212.10560*, 2022.

[174] R. Ri, S. Kiyono, and S. Takase, "Self-translate-train: Enhancing cross-lingual transfer of large language models via inherent capability," *arXiv preprint arXiv:2407.00454*, 2024.

[175] J. Ji, M. Liu, J. Dai, X. Pan, C. Zhang, C. Bian, B. Chen, R. Sun, Y. Wang, and Y. Yang, "Beavertails: Towards improved safety alignment of llm via a human-preference dataset," *Advances in Neural Information Processing Systems*, vol. 36, pp. 24 678–24 704, 2023.

[176] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe, "Let's verify step by step," in *The Twelfth International Conference on Learning Representations*, 2023.

[177] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders *et al.*, "Webgpt: Browser-assisted question-answering with human feedback," *arXiv preprint arXiv:2112.09332*, 2021.

[178] C. Chen, J. Fu, and L. Lyu, "A pathway towards responsible ai generated content," *arXiv preprint arXiv:2303.01325*, 2023.

[179] A. Akkus, M. P. Aghdam, M. Li, J. Chu, M. Backes, Y. Zhang, and S. Sav, "Generated data with fake privacy: Hidden dangers of fine-tuning large language models on generated data," *arXiv preprint arXiv:2409.11423*, 2024.

[180] Y. Song, J. Zhang, Z. Tian, Y. Yang, M. Huang, and D. Li, "Llm-based privacy data augmentation guided by knowledge distillation with a distribution tutor for medical text classification," *arXiv preprint arXiv:2402.16515*, 2024.

[181] A. Kang, J. Y. Chen, Z. Lee-Youngzie, and S. Fu, "Synthetic data generation with llm for improved depression prediction," *arXiv preprint arXiv:2411.17672*, 2024.

[182] A. Taubenfeld, Y. Dover, R. Reichart, and A. Goldstein, "Systematic biases in llm simulations of debates," *arXiv preprint arXiv:2402.04049*, 2024.

[183] A. Mishra, G. Nayak, S. Bhattacharya, T. Kumar, A. Shah, and M. Foltin, "Llm-guided counterfactual data generation for fairer ai," in *Companion Proceedings of the ACM Web Conference 2024*, 2024, pp. 1538–1545.

[184] Y. Yu, Y. Zhuang, J. Zhang, Y. Meng, A. J. Ratner, R. Krishna, J. Shen, and C. Zhang, "Large language model as attributed training data generator: A tale of diversity and bias," *Advances in Neural Information Processing Systems*, vol. 36, pp. 55 734–55 784, 2023.

[185] A. Borah and R. Mihalcea, "Towards implicit bias detection and mitigation in multi-agent llm interactions," *arXiv preprint arXiv:2410.02584*, 2024.

[186] X. Dong, Y. Wang, P. S. Yu, and J. Caverlee, "Disclosure and mitigation of gender bias in llms," *arXiv preprint arXiv:2402.11190*, 2024.

[187] I. M. Serouis and F. Sèdes, "Exploring large language models for bias mitigation and fairness," in *1st International Workshop on AI Governance (AIGOV) in conjunction with the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

[188] Y. Chen, Q. Fu, Y. Yuan, Z. Wen, G. Fan, D. Liu, D. Zhang, Z. Li, and Y. Xiao, "Hallucination detection: Robustly discerning reliable answers in large language models," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 245–255.

[189] N. Chakraborty, M. Ornik, and K. Driggs-Campbell, "Hallucination detection in foundation models for decision-making: A flexible definition and review of the state of the art," *ACM Computing Surveys*, 2025.

[190] E. Entezami and A. Naseh, "Llm misalignment via adversarial rlhf platforms," *arXiv preprint arXiv:2503.03039*, 2025.

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

[192] A. Young, B. Chen, C. Li, C. Huang, G. Zhang, G. Zhang, G. Wang, H. Li, J. Zhu, J. Chen *et al.*, "Yi: Open foundation models by 01. ai," *arXiv preprint arXiv:2403.04652*, 2024.

[193] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[194] Z. Cai, M. Cao, H. Chen, K. Chen, K. Chen, X. Chen, X. Chen, Z. Chen, Z. Chen, P. Chu *et al.*, "Internlm2

technical report," *arXiv preprint arXiv:2403.17297*, 2024.

[195] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen *et al.*, "Palm 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

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

[197] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[198] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love *et al.*, "Gemma: Open models based on gemini research and technology," *arXiv preprint arXiv:2403.08295*, 2024.

[199] D. Groeneveld, I. Beltagy, P. Walsh, A. Bhagia, R. Kinney, O. Tafjord, A. H. Jha, H. Ivison, I. Magnusson, Y. Wang *et al.*, "Olmo: Accelerating the science of language models," *arXiv preprint arXiv:2402.00838*, 2024.

[200] B. Adler, N. Agarwal, A. Aithal, D. H. Anh, P. Bhattacharya, A. Brundyn, J. Casper, B. Catanzaro, S. Clay, J. Cohen *et al.*, "Nemotron-4 340b technical report," *arXiv preprint arXiv:2406.11704*, 2024.

[201] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney *et al.*, "Openai o1 system card," *arXiv preprint arXiv:2412.16720*, 2024.

[202] OpenAI, "Gpt-4o mini: advancing cost-efficient intelligence," 2024, https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence.

[203] A. Yang, B. Xiao, B. Wang, B. Zhang, C. Bian, C. Yin, C. Lv, D. Pan, D. Wang, D. Yan *et al.*, "Baichuan 2: Open large-scale language models," *arXiv preprint arXiv:2309.10305*, 2023.

[204] J. Welbl, A. Glaese, J. Uesato, S. Dathathri, J. Mellor, L. A. Hendricks, K. Anderson, P. Kohli, B. Coppin, and P.-S. Huang, "Challenges in detoxifying language models," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 2447–2469.

[205] H. Ngo, C. Raterink, J. G. Araújo, I. Zhang, C. Chen, A. Morisot, and N. Frosst, "Mitigating harm in language models with conditional-likelihood filtration," *arXiv preprint arXiv:2108.07790*, 2021.

[206] Y. Chen, W. Cai, L. Wu, X. Li, Z. Xin, and C. Fu, "Tigerbot: An open multilingual multitask llm," *arXiv preprint arXiv:2312.08688*, 2023.

[207] S. Prabhumoye, M. Patwary, M. Shoeybi, and B. Catanzaro, "Adding instructions during pretraining: Effective way of controlling toxicity in language models," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 2636–2651.

[208] Y. Ge, W. Sun, Y. Lou, C. Fang, Y. Zhang, Y. Li, X. Zhang, Y. Liu, Z. Zhao, and Z. Chen, "Demonstration attack against in-context learning for code intelligence," *CoRR*, vol. abs/2410.02841, no. 1, pp. 1–17, 2024.

[209] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[210] J. Parmar, S. Prabhumoye, J. Jennings, M. Patwary, S. Subramanian, D. Su, C. Zhu, D. Narayanan, A. Jhunjhunwala, A. Dattagupta *et al.*, "Nemotron-4 15b technical report," *arXiv preprint arXiv:2402.16819*, 2024.

[211] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[212] T. Markov, C. Zhang, S. Agarwal, F. E. Nekoul, T. Lee, S. Adler, A. Jiang, and L. Weng, "A holistic approach to undesired content detection in the real world," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 12, 2023, pp. 15 009–15 018.

[213] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[214] T. Huang, S. Hu, F. Ilhan, S. F. Tekin, and L. Liu, "Harmful fine-tuning attacks and defenses for large language models: A survey," *arXiv preprint arXiv:2409.18169*, 2024.

[215] J. Wu, Y. Xie, Z. Yang, J. Wu, J. Chen, J. Gao, B. Ding, X. Wang, and X. He, "Towards robust alignment of language models: Distributionally robustifying direct preference optimization," *arXiv preprint arXiv:2407.07880*, 2024.

[216] Z. Xu, S. Vemuri, K. Panaganti, D. Kalathil, R. Jain, and D. Ramachandran, "Distributionally robust direct preference optimization," *arXiv preprint arXiv:2502.01930*, 2025.

[217] J. Dai, X. Pan, R. Sun, J. Ji, X. Xu, M. Liu, Y. Wang, and Y. Yang, "Safe rlhf: Safe reinforcement learning from human feedback," in *The Twelfth International Conference on Learning Representations*, 2023.

[218] C. O. Retzlaff, S. Das, C. Wayllace, P. Mousavi, M. Afshari, T. Yang, A. Saranti, A. Angerschmid, M. E. Taylor, and A. Holzinger, "Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities," *Journal of Artificial Intelligence Research*, vol. 79, pp. 359–415, 2024.

[219] S. Milani, N. Topin, M. Veloso, and F. Fang, "Explainable reinforcement learning: A survey and comparative review," *ACM Computing Surveys*, vol. 56, no. 7, pp. 1–36, 2024.

[220] A. Ahmadian, C. Cremer, M. Gallé, M. Fadaee, J. Kreutzer, O. Pietquin, A. Üstün, and S. Hooker, "Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms," *arXiv preprint arXiv:2402.14740*, 2024.

[221] T. Liu, Z. Qin, J. Wu, J. Shen, M. Khalman, R. Joshi, Y. Zhao, M. Saleh, S. Baumgartner, J. Liu *et al.*, "Lipo: Listwise preference optimization through learning-to-rank," *arXiv preprint arXiv:2402.01878*, 2024.

[222] F. Song, B. Yu, M. Li, H. Yu, F. Huang, Y. Li, and

H. Wang, "Preference ranking optimization for human alignment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 18 990–18 998.

[223] Z. Wang, B. Bi, S. K. Pentyala, K. Ramnath, S. Chaudhuri, S. Mehrotra, X.-B. Mao, S. Asur *et al.*, "A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more," *arXiv preprint arXiv:2407.16216*, 2024.

[224] T. Huang, S. Hu, F. Ilhan, S. F. Tekin, and L. Liu, "Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=RPChapuXlC

[225] T. Huang, S. Hu, and L. Liu, "Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=lpXDZKiAnt

[226] J. Wang, J. Li, Y. Li, X. Qi, J. Hu, Y. Li, P. McDaniel, M. Chen, B. Li, and C. Xiao, "Backdooralign: Mitigating fine-tuning based jailbreak attack with backdoor enhanced safety alignment," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=1PcJ5Evta7

[227] F. Bianchi, M. Suzgun, G. Attanasio, P. Rottger, D. Jurafsky, T. Hashimoto, and J. Zou, "Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=gT5hALch9z

[228] H. Shen, P.-Y. Chen, P. Das, and T. Chen, "SEAL: Safety-enhanced aligned LLM fine-tuning via bilevel data selection," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=VHguhvcoM5

[229] R. Tang, J. Yuan, Y. Li, Z. Liu, R. Chen, and X. Hu, "Setting the trap: Capturing and defeating backdoor threats in plms through honeypots," *NeurIPS*, 2023.

[230] C.-Y. Hsu, Y.-L. Tsai, C.-H. Lin, P.-Y. Chen, C.-M. Yu, and C.-Y. Huang, "Safe loRA: The silver lining of reducing safety risks when finetuning large language models," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=HcifdQZFZV

[231] R. Hazra, S. Layek, S. Banerjee, and S. Poria, "Safety arithmetic: A framework for test-time safety alignment of language models by steering parameters and activations," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 21 759–21 776.

[232] Y. Du, S. Zhao, D. Zhao, M. Ma, Y. Chen, L. Huo, Q. Yang, D. Xu, and B. Qin, "MoGU: A framework for enhancing safety of LLMs while preserving their usability," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=SrFbgIjb53

[233] X. Yi, S. Zheng, L. Wang, G. de Melo, X. Wang, and L. He, "Nlsr: Neuron-level safety realignment of large language models against harmful fine-tuning," *arXiv preprint arXiv:2412.12497*, 2024.

[234] D. Shi, T. Shen, Y. Huang, Z. Li, Y. Leng, R. Jin, C. Liu, X. Wu, Z. Guo, L. Yu *et al.*, "Large language model safety: A holistic survey," *arXiv preprint arXiv:2412.17686*, 2024.

[235] B. Ni, Z. Liu, L. Wang, Y. Lei, Y. Zhao, X. Cheng, Q. Zeng, L. Dong, Y. Xia, K. Kenthapadi *et al.*, "Towards trustworthy retrieval augmented generation for large language models: A survey," *arXiv preprint arXiv:2502.06872*, 2025.

[236] F. Barez, T. Fu, A. Prabhu, S. Casper, A. Sanyal, A. Bibi, A. O'Gara, R. Kirk, B. Bucknall, T. Fist, L. Ong, P. Torr, K. Lam, R. Trager, D. Krueger, S. Mindermann, J. Hernández-Orallo, M. Geva, and Y. Gal, "Open problems in machine unlearning for AI safety," *CoRR*, 2025.

[237] U. Anwar, A. Saparov, J. Rando, D. Paleka, M. Turpin, P. Hase, E. S. Lubana, E. Jenner, S. Casper, O. Sourbut *et al.*, "Foundational challenges in assuring alignment and safety of large language models," *arXiv preprint arXiv:2404.09932*, 2024.

[238] X. Qi, Y. Zeng, T. Xie, P.-Y. Chen, R. Jia, P. Mittal, and P. Henderson, "Fine-tuning aligned language models compromises safety, even when users do not intend to!" *arXiv preprint arXiv:2310.03693*, 2023.

[239] X. Yang, X. Wang, Q. Zhang, L. Petzold, W. Y. Wang, X. Zhao, and D. Lin, "Shadow alignment: The ease of subverting safely-aligned language models.(2023)," *arXiv preprint arXiv:2310.02949*, 2023.

[240] Q. Zhan, R. Fang, R. Bindu, A. Gupta, T. Hashimoto, and D. Kang, "Removing rlhf protections in gpt-4 via fine-tuning," *arXiv preprint arXiv:2311.05553*, 2023.

[241] J. Kazdan, L. Yu, R. Schaeffer, C. Cundy, S. Koyejo, and D. Krishnamurthy, "No, of course i can! refusal mechanisms can be exploited using harmless fine-tuning data," *arXiv preprint arXiv:2502.19537*, 2025.

[242] D. Halawi, A. Wei, E. Wallace, T. T. Wang, N. Haghtalab, and J. Steinhardt, "Covert malicious finetuning: Challenges in safeguarding llm adaptation," *arXiv preprint arXiv:2406.20053*, 2024.

[243] T. Huang, S. Hu, F. Ilhan, S. F. Tekin, and L. Liu, "Virus: Harmful fine-tuning attack for large language models bypassing guardrail moderation," *arXiv preprint arXiv:2501.17433*, 2025.

[244] Y. Qiang, X. Zhou, S. Z. Zade, M. A. Roshani, P. Khanduri, D. Zytko, and D. Zhu, "Learning to poison large language models during instruction tuning," *arXiv preprint arXiv:2402.13459*, 2024.

[245] J. Raghuram, G. Kesidis, and D. J. Miller, "A study of backdoors in instruction fine-tuned language models," *arXiv preprint arXiv:2406.07778*, 2024.

[246] J. Yi, R. Ye, Q. Chen, B. Zhu, S. Chen, D. Lian, G. Sun, X. Xie, and F. Wu, "On the vulnerability of safety alignment in open-access llms," in *Findings of the Association for Computational Linguistics ACL 2024*,

2024, pp. 9236–9260.

[247] S. Lermen, C. Rogers-Smith, and J. Ladish, "Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b," *arXiv preprint arXiv:2310.20624*, 2023.

[248] L. Piercing, "Lora-as-an-attack! piercing llm safety under the share-and-play scenario."

[249] S. Poppi, Z.-X. Yong, Y. He, B. Chern, H. Zhao, A. Yang, and J. Chi, "Towards understanding the fragility of multilingual llms against fine-tuning attacks," *arXiv preprint arXiv:2410.18210*, 2024.

[250] S. Li, E. C.-H. Ngai, F. Ye, and T. Voigt, "Peft-as-an-attack! jailbreaking language models during federated parameter-efficient fine-tuning," *arXiv preprint arXiv:2411.19335*, 2024.

[251] N. Razin, S. Malladi, A. Bhaskar, D. Chen, S. Arora, and B. Hanin, "Unintentional unalignment: Likelihood displacement in direct preference optimization," *arXiv preprint arXiv:2410.08847*, 2024.

[252] R. Xu, Y. Cai, Z. Zhou, R. Gu, H. Weng, Y. Liu, T. Zhang, W. Xu, and H. Qiu, "Course-correction: Safety alignment using synthetic preferences," *arXiv preprint arXiv:2407.16637*, 2024.

[253] J. Ji, B. Chen, H. Lou, D. Hong, B. Zhang, X. Pan, T. A. Qiu, J. Dai, and Y. Yang, "Aligner: Efficient alignment by learning to correct," *Advances in Neural Information Processing Systems*, vol. 37, pp. 90 853–90 890, 2024.

[254] D. Ganguli, L. Lovitt, J. Kernion, A. Askell, Y. Bai, S. Kadavath, B. Mann, E. Perez, N. Schiefer, K. Ndousse *et al.*, "Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned," *arXiv preprint arXiv:2209.07858*, 2022.

[255] T. Xiao, Y. Yuan, H. Zhu, M. Li, and V. G. Honavar, "Cal-DPO: Calibrated direct preference optimization for language model alignment," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=57OQXxbTbY

[256] S. Guo, B. Zhang, T. Liu, T. Liu, M. Khalman, F. Llinares, A. Rame, T. Mesnard, Y. Zhao, B. Piot *et al.*, "Direct language model alignment from online ai feedback," *arXiv preprint arXiv:2402.04792*, 2024.

[257] Z. Liu, X. Sun, and Z. Zheng, "Enhancing llm safety via constrained direct preference optimization," *arXiv preprint arXiv:2403.02475*, 2024.

[258] H. Lee, S. Phatale, H. Mansoor, T. Mesnard, J. Ferret, K. R. Lu, C. Bishop, E. Hall, V. Carbune, A. Rastogi, and S. Prakash, "RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=uydQ2W41KO

[259] X. Lu, B. Yu, Y. Lu, H. Lin, H. Yu, L. Sun, X. Han, and Y. Li, "Sofa: Shielded on-the-fly alignment via priority rule following," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 7108–7136.

[260] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *arXiv preprint arXiv:2307.15043*, 2023.

[261] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong, "Jailbreaking black box large language models in twenty queries," *arXiv preprint arXiv:2310.08419*, 2023.

[262] Z. Zhou, J. Xiang, H. Chen, Q. Liu, Z. Li, and S. Su, "Speak out of turn: Safety vulnerability of large language models in multi-turn dialogue," *arXiv preprint arXiv:2402.17262*, 2024.

[263] Q. Ren, H. Li, D. Liu, Z. Xie, X. Lu, Y. Qiao, L. Sha, J. Yan, L. Ma, and J. Shao, "Derail yourself: Multi-turn llm jailbreak attack through self-discovered clues," *arXiv preprint arXiv:2410.10700*, 2024.

[264] X. Pang, S. Tang, R. Ye, Y. Xiong, B. Zhang, Y. Wang, and S. Chen, "Self-alignment of large language models via monopolylogue-based social scene simulation," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 39 416–39 447.

[265] J. Ji, D. Hong, B. Zhang, B. Chen, J. Dai, B. Zheng, T. Qiu, B. Li, and Y. Yang, "Pku-saferlhf: Towards multi-level safety alignment for llms with human preference," *arXiv preprint arXiv:2406.15513*, 2024.

[266] T. Mu, A. Helyar, J. Heidecke, J. Achiam, A. Vallone, I. D. Kivlichan, M. Lin, A. Beutel, J. Schulman, and L. Weng, "Rule based rewards for language model safety," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[267] X. Tan, S. Shi, X. Qiu, C. Qu, Z. Qi, Y. Xu, and Y. Qi, "Self-criticism: Aligning large language models with their understanding of helpfulness, honesty, and harmlessness," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, M. Wang and I. Zitouni, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 650–662. [Online]. Available: https://aclanthology.org/2023.emnlp-industry.62/

[268] M. Y. Guan, M. Joglekar, E. Wallace, S. Jain, B. Barak, A. Heylar, R. Dias, A. Vallone, H. Ren, J. Wei *et al.*, "Deliberative alignment: Reasoning enables safer language models," *arXiv preprint arXiv:2412.16339*, 2024.

[269] B. Wei, K. Huang, Y. Huang, T. Xie, X. Qi, M. Xia, P. Mittal, M. Wang, and P. Henderson, "Assessing the brittleness of safety alignment via pruning and low-rank modifications," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=K6xxnKN2gm

[270] A. Arditi, O. B. Obeso, A. Syed, D. Paleka, N. Rimsky, W. Gurnee, and N. Nanda, "Refusal in language models is mediated by a single direction," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=pH3XAQME6c

[271] R. Ye, J. Chai, X. Liu, Y. Yang, Y. Wang, and S. Chen, "Emerging safety attack and defense in federated instruction tuning of large language models," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=sYNWqQYJhz

[272] J. Mukhoti, Y. Gal, P. Torr, and P. K. Dokania, "Fine-tuning can cripple foundation models; preserving features may be the solution," 2024. [Online]. Available: https://openreview.net/forum?id=VQ7Q6qdp0P

[273] Y. Du, S. Zhao, J. Cao, M. Ma, D. Zhao, F. FAN, T. Liu, and B. Qin, "Towards secure tuning: Mitigating security risks arising from benign instruction fine-tuning," 2024. [Online]. Available: https://openreview.net/forum?id=Egd7Vi1EuA

[274] J. Li and J.-E. Kim, "Safety alignment shouldn't be complicated," 2025. [Online]. Available: https://openreview.net/forum?id=9H91juqfgb

[275] S. Li, L. Yao, L. Zhang, and Y. Li, "Safety layers in aligned large language models: The key to LLM security," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=kUH1yPMAn7

[276] Z. Zhou, H. Yu, X. Zhang, R. Xu, F. Huang, K. Wang, Y. Liu, J. Fang, and Y. Li, "On the role of attention heads in large language model safety," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=h0Ak8A5yqw

[277] M. Li, W. M. Si, M. Backes, Y. Zhang, and Y. Wang, "SaloRA: Safety-alignment preserved low-rank adaptation," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=GOoVzE9nSj

[278] Y. Zong, O. Bohdal, T. Yu, Y. Yang, and T. Hospedales, "Safety fine-tuning at (almost) no cost: A baseline for vision large language models," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=bWZKvF0g7G

[279] F. Eiras, A. Petrov, P. Torr, M. P. Kumar, and A. Bibi, "Do as i do (safely): Mitigating task-specific fine-tuning risks in large language models," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=lXE5lB6ppV

[280] J. Luo, X. Luo, K. Ding, J. Yuan, Z. Xiao, and M. Zhang, "Robustft: Robust supervised fine-tuning for large language models under noisy response," 2024. [Online]. Available: https://arxiv.org/abs/2412.14922

[281] K. Lyu, H. Zhao, X. Gu, D. Yu, A. Goyal, and S. Arora, "Keeping LLMs aligned after fine-tuning: The crucial role of prompt templates," in *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*, 2024. [Online]. Available: https://openreview.net/forum?id=XlnpQOn95Z

[282] P. Hacker, A. Engel, and M. Mauer, "Regulating chatgpt and other large generative ai models," in *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, 2023.

[283] M. Kolla, S. Salunkhe, E. Chandrasekharan, and K. Saha, "Llm-mod: Can large language models assist content moderation?" in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2024.

[284] D. Kumar, Y. A. AbuHashem, and Z. Durumeric, "Watch your language: Investigating content moderation with large language models," *Proceedings of the International AAAI Conference on Web and Social Media*, 2024.

[285] H. K. Choi, X. Du, and Y. Li, "Safety-aware fine-tuning of large language models," in *Neurips Safe Generative AI Workshop 2024*, 2024. [Online]. Available: https://openreview.net/forum?id=SqL94fLSM7

[286] H. Ge, Y. Li, Q. Wang, Y. Zhang, and R. Tang, "When backdoors speak: Understanding llm backdoor attacks through model-generated explanations," *arXiv preprint arXiv:2411.12701*, 2024.

[287] B. Yi, T. Huang, S. Chen, T. Li, Z. Liu, Z. Chu, and Y. Li, "Probe before you talk: Towards black-box defense against backdoor unalignment for large language models," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=EbxYDBhE3S

[288] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018.

[289] S. Casper, L. Schulze, O. Patel, and D. Hadfield-Menell, "Defending against unforeseen failure modes with latent adversarial training," 2024. [Online]. Available: https://arxiv.org/abs/2403.05030

[290] T. Huang, G. Bhattacharya, P. Joshi, J. Kimball, and L. Liu, "Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning," 2024. [Online]. Available: https://arxiv.org/abs/2408.09600

[291] J. Li, "Detecting instruction fine-tuning attack on language models with influence function," *arXiv preprint arXiv:2504.09026*, 2025.

[292] X. Yi, S. Zheng, L. Wang, X. Wang, and L. He, "A safety realignment framework via subspace-oriented model fusion for large language models," *Knowledge-Based Systems*, 2024.

[293] M. Zhu, Y. Weng, L. Yang, Y. Wei, N. Zhang, and Y. Zhang, "Locking down the finetuned LLMs safety," 2025. [Online]. Available: https://openreview.net/forum?id=YGoFl5KKFc

[294] D. Wu, X. Lu, Y. Zhao, and B. Qin, "Separate the wheat from the chaff: A post-hoc approach to safety re-alignment for fine-tuned language models," 2025. [Online]. Available: https://arxiv.org/abs/2412.11041

[295] Y. Wang, T. Huang, L. Shen, H. Yao, H. Luo, R. Liu, N. Tan, J. Huang, and D. Tao, "Panacea: Mitigating harmful fine-tuning for large language models via post-fine-tuning perturbation," 2025. [Online]. Available: https://arxiv.org/abs/2501.18100

[296] Q. Liu, C. Shang, L. Liu, N. Pappas, J. Ma, N. A. John, S. Doss, L. Marquez, M. Ballesteros, and Y. Benajiba, "Unraveling and mitigating safety alignment degradation of vision-language models," 2025. [Online]. Available: https://openreview.net/forum?id=EEWpE9cR27

[297] S. Xu, L. Pang, Y. Zhu, H. Shen, and X. Cheng, "Cross-modal safety mechanism transfer in large vision-language models," *arXiv preprint arXiv:2410.12662*, 2024.

[298] S. Li, L. Yao, L. Zhang, and Y. Li, "Safety layers in aligned large language models: The key to llm security," *arXiv preprint arXiv:2408.17003*, 2024.

[299] W. Zhao, Z. Li, Y. Li, Y. Zhang, and J. Sun, "Defending large language models against jailbreak

attacks via layer-specific editing," 2024. [Online]. Available: https://arxiv.org/abs/2405.18166

[300] NIST, "Artificial intelligence risk management framework: Generative artificial intelligence profile (initial public draft)," 2024, accessed: 2025-05-29. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.800-1.ipd.pdf

[301] X. Qi, B. Wei, N. Carlini, Y. Huang, T. Xie, L. He, M. Jagielski, M. Nasr, P. Mittal, and P. Henderson, "On Evaluating the Durability of Safeguards for Open-Weight LLMs," Dec. 2024.

[302] D. Rosati, J. Wehner, K. Williams, L. Bartoszcze, R. Gonzales, C. Maple, S. Majumdar, H. Sajjad, and F. Rudzicz, "Representation Noising: A Defence Mechanism Against Harmful Finetuning," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, Nov. 2024.

[303] R. Tamirisa, B. Bharathi, L. Phan, A. Zhou, A. Gatti, T. Suresh, M. Lin, J. Wang, R. Wang, R. Arel, A. Zou, D. Song, B. Li, D. Hendrycks, and M. Mazeika, "Tamper-Resistant Safeguards for Open-Weight LLMs," Feb. 2025.

[304] D. Rosati, J. Wehner, K. Williams, L. Bartoszcze, H. Sajjad, and F. Rudzicz, "Immunization against harmful fine-tuning attacks," in *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 2024.

[305] M. Mazeika, L. Phan, X. Yin, A. Zou, Z. Wang, N. Mu, E. Sakhaee, N. Li, S. Basart, B. Li *et al.*, "Harmbench: A standardized evaluation framework for automated red teaming and robust refusal," *arXiv preprint arXiv:2402.04249*, 2024.

[306] P. Chao, E. Debenedetti, A. Robey, M. Andriushchenko, F. Croce, V. Sehwag, E. Dobriban, N. Flammarion, G. J. Pappas, F. Tramer *et al.*, "Jailbreakbench: An open robustness benchmark for jailbreaking large language models," *arXiv preprint arXiv:2404.01318*, 2024.

[307] S. Liu, S. Cui, H. Bu, Y. Shang, and X. Zhang, "Jailbench: A comprehensive chinese security assessment benchmark for large language models," *arXiv preprint arXiv:2502.18935*, 2025.

[308] J. Cui, W.-L. Chiang, I. Stoica, and C.-J. Hsieh, "Or-bench: An over-refusal benchmark for large language models," *arXiv preprint arXiv:2405.20947*, 2024.

[309] T. Xie, X. Qi, Y. Zeng, Y. Huang, U. M. Sehwag, K. Huang, L. He, B. Wei, D. Li, Y. Sheng *et al.*, "Sorry-bench: Systematically evaluating large language model safety refusal behaviors," *arXiv preprint arXiv:2406.14598*, 2024.

[310] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 595–46 623, 2023.

[311] Z. Wang, S. Hu, S. Zhao, X. Lin, F. Juefei-Xu, Z. Li, L. Han, H. Subramanyam, L. Chen, J. Chen *et al.*, "Mllm-as-a-judge for image safety without human labeling," *arXiv preprint arXiv:2501.00192*, 2024.

[312] D. Rosati, J. Wehner, K. Williams, Ł. Bartoszcze, D. Atanasov, R. Gonzales, S. Majumdar, C. Maple, H. Sajjad, and F. Rudzicz, "Representation noising effectively prevents harmful fine-tuning on llms," *arXiv e-prints*, pp. arXiv–2405, 2024.

[313] H. Zhang, J. Huang, K. Mei, Y. Yao, Z. Wang, C. Zhan, H. Wang, and Y. Zhang, "Agent security bench (ASB): Formalizing and benchmarking attacks and defenses in LLM-based agents," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=V4y0CpX4hK

[314] T. Yuan, Z. He, L. Dong, Y. Wang, R. Zhao, T. Xia, L. Xu, B. Zhou, F. Li, Z. Zhang *et al.*, "R-judge: Benchmarking safety risk awareness for llm agents," *arXiv preprint arXiv:2401.10019*, 2024.

[315] Z. Zhang, L. Lei, L. Wu, R. Sun, Y. Huang, C. Long, X. Liu, X. Lei, J. Tang, and M. Huang, "Safetybench: Evaluating the safety of large language models," *arXiv preprint arXiv:2309.07045*, 2023.

[316] L. Li, B. Dong, R. Wang, X. Hu, W. Zuo, D. Lin, Y. Qiao, and J. Shao, "Salad-bench: A hierarchical and comprehensive safety benchmark for large language models," *arXiv preprint arXiv:2402.05044*, 2024.

[317] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano *et al.*, "Training verifiers to solve math word problems," *arXiv preprint arXiv:2110.14168*, 2021.

[318] S.-Y. Miao, C.-C. Liang, and K.-Y. Su, "A diverse corpus for evaluating and developing english math word problem solvers," *arXiv preprint arXiv:2106.15772*, 2021.

[319] E. Glazer, E. Erdil, T. Besiroglu, D. Chicharro, E. Chen, A. Gunning, C. F. Olsson, J.-S. Denain, A. Ho, E. d. O. Santos *et al.*, "Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai," *arXiv preprint arXiv:2411.04872*, 2024.

[320] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[321] C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, and K. Narasimhan, "Swe-bench: Can language models resolve real-world github issues?" *arXiv preprint arXiv:2310.06770*, 2023.

[322] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, 2015.

[323] H. Luo, Y. Jin, X. Liu, T. Shang, R. Chen, and Z. Liu, "Geic: Universal and multilingual named entity recognition with large language models," *arXiv preprint arXiv:2409.11022*, 2024.

[324] X. Li, T. Zhang, Y. Dubois, R. Taori, I. Gulrajani, C. Guestrin, P. Liang, and T. B. Hashimoto, "Alpacaeval: An automatic evaluator of instruction-following models," 2023.

[325] W.-L. Chiang, L. Zheng, Y. Sheng, A. N. Angelopoulos, T. Li, D. Li, B. Zhu, H. Zhang, M. Jordan, J. E. Gonzalez *et al.*, "Chatbot arena: An open platform for evaluating llms by human preference," in *Forty-first International Conference on Machine Learning*, 2024.

[326] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, "Samsum corpus: A human-annotated dialogue

dataset for abstractive summarization," *arXiv preprint arXiv:1911.12237*, 2019.

[327] M. Macháček and O. Bojar, "Results of the wmt14 metrics shared task," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 2014, pp. 293–301.

[328] X. Lu, D. Liu, Y. Yu, L. Xu, and J. Shao, "X-boundary: Establishing exact safety boundary to shield llms from multi-turn jailbreaks without compromising usability," *arXiv preprint arXiv:2502.09990*, 2025.

[329] OpenAI, "Moderation api," https://platform.openai.com/docs/guides/moderation/overview, 2023.

[330] H. Inan, K. Upasani, J. Chi, R. Rungta, K. Iyer, Y. Mao, M. Tontchev, Q. Hu, B. Fuller, D. Testuggine, and M. Khabsa, "Llama guard: Llm-based input-output safeguard for human-ai conversations," *CoRR*, 2023.

[331] J. Ji, T. Qiu, B. Chen, B. Zhang, H. Lou, K. Wang, Y. Duan, Z. He, J. Zhou, Z. Zhang *et al.*, "Ai alignment: A comprehensive survey," *arXiv preprint arXiv:2310.19852*, 2023.

[332] T. A. Qiu, Y. Zhang, X. Huang, J. Li, J. Ji, and Y. Yang, "Progressgym: Alignment with a millennium of moral progress," *Advances in Neural Information Processing Systems*, vol. 37, pp. 14 570–14 607, 2024.

[333] B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer *et al.*, "Decodingtrust: A comprehensive assessment of trustworthiness in gpt models." in *NeurIPS*, 2023.

[334] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, "Realtoxicityprompts: Evaluating neural toxic degeneration in language models," *arXiv preprint arXiv:2009.11462*, 2020.

[335] Y. Wang, H. Li, X. Han, P. Nakov, and T. Baldwin, "Do-not-answer: A dataset for evaluating safeguards in llms," *arXiv preprint arXiv:2308.13387*, 2023.

[336] M. Conover, R. Staats, A. Rane, G. Shani, K. Katz, A. Powell, A. Ross, A. Maas, and A. Zhang, "Databricks-dolly: Introducing dolly-15k, democratizing the magic of instruction following," https://github.com/databrickslabs/dolly, 2023.

[337] X. Wu, Y. Hao, K. Sun, Y. Chen, F. Zhu, R. Zhao, and H. Li, "Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis," *arXiv preprint arXiv:2306.09341*, 2023.

[338] Y. Yan, S. Wang, J. Huo, H. Li, B. Li, J. Su, X. Gao, Y.-F. Zhang, T. Xu, Z. Chu *et al.*, "Errorradar: Benchmarking complex mathematical reasoning of multimodal large language models via error detection," *arXiv preprint arXiv:2410.04509*, 2024.

[339] Q. Jin, B. Dhingra, Z. Liu, W. W. Cohen, and X. Lu, "Pubmedqa: A dataset for biomedical research question answering," *arXiv preprint arXiv:1909.06146*, 2019.

[340] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," *Advances in neural information processing systems*, vol. 28, 2015.

[341] S. Lin, J. Hilton, and O. Evans, "Truthfulqa: Measuring how models mimic human falsehoods," *arXiv preprint arXiv:2109.07958*, 2021.

[342] Y. Mou, S. Zhang, and W. Ye, "Sg-bench: Evaluating llm safety generalization across diverse tasks and prompt types," *Advances in Neural Information Processing Systems*, vol. 37, pp. 123 032–123 054, 2024.

[343] F. Jiang, Z. Xu, Y. Li, L. Niu, Z. Xiang, B. Li, B. Y. Lin, and R. Poovendran, "Safechain: Safety of language models with long chain-of-thought reasoning capabilities," *arXiv preprint arXiv:2502.12025*, 2025.

[344] T. Hartvigsen, S. Gabriel, H. Palangi, M. Sap, D. Ray, and E. Kamar, "Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection," *arXiv preprint arXiv:2203.09509*, 2022.

[345] A. Souly, Q. Lu, D. Bowen, T. Trinh, E. Hsieh, S. Pandey, P. Abbeel, J. Svegliato, S. Emmons, O. Watkins *et al.*, "A strongreject for empty jailbreaks," *arXiv preprint arXiv:2402.10260*, 2024.

[346] L. Jiang, K. Rao, S. Han, A. Ettinger, F. Brahman, S. Kumar, N. Mireshghallah, X. Lu, M. Sap, Y. Choi *et al.*, "Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 47 094–47 165, 2024.

[347] D. Hendrycks, M. Mazeika, and T. Woodside, "An overview of catastrophic ai risks," *arXiv preprint arXiv:2306.12001*, 2023.

[348] B. Baker, J. Huizinga, L. Gao, Z. Dou, M. Y. Guan, A. Madry, W. Zaremba, J. Pachocki, and D. Farhi, "Monitoring reasoning models for misbehavior and the risks of promoting obfuscation," *arXiv preprint arXiv:2503.11926*, 2025.

[349] T. Hagendorff, "Deception abilities emerged in large language models," *Proceedings of the National Academy of Sciences*, vol. 121, no. 24, p. e2317967121, 2024. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.2317967121

[350] P. S. Park, S. Goldstein, A. O'Gara, M. Chen, and D. Hendrycks, "Ai deception: A survey of examples, risks, and potential solutions," *Patterns*, vol. 5, no. 5, 2024.

[351] OpenAI, "Gpt-4 technical report," *ArXiv*, vol. abs/2303.08774, 2023.

[352] F. Ward, F. Toni, F. Belardinelli, and T. Everitt, "Honesty is the best policy: defining and mitigating ai deception," *Advances in neural information processing systems*, vol. 36, pp. 2313–2341, 2023.

[353] J. Scheurer, M. Balesni, and M. Hobbhahn, "Large language models can strategically deceive their users when put under pressure," *arXiv preprint arXiv:2311.07590*, 2023.

[354] S. Chern, Z. Hu, Y. Yang, E. Chern, Y. Guo, J. Jin, B. Wang, and P. Liu, "Behonest: Benchmarking honesty in large language models," *arXiv preprint arXiv:2406.13261*, 2024.

[355] A. O'Gara, "Hoodwinked: Deception and cooperation in a text-based game for language models," *arXiv preprint arXiv:2308.01404*, 2023.

[356] M. F. A. R. D. T. (FAIR)†, A. Bakhtin, N. Brown, E. Dinan, G. Farina, C. Flaherty, D. Fried, A. Goff, J. Gray, H. Hu *et al.*, "Human-level play in the game of diplomacy by combining language models with strategic reasoning," *Science*, vol. 378, no. 6624, pp. 1067–1074, 2022.

[357] L. Schulz, N. Alon, J. Rosenschein, and P. Dayan, "Emergent deception and skepticism via theory of mind," in *First Workshop on Theory of Mind in Communicating Agents*, 2023.

[358] A. Meinke, B. Schoen, J. Scheurer, M. Balesni, R. Shah, and M. Hobbhahn, "Frontier models are capable of in-context scheming," *arXiv preprint arXiv:2412.04984*, 2024.

[359] R. Greenblatt, C. Denison, B. Wright, F. Roger, M. MacDiarmid, S. Marks, J. Treutlein, T. Belonax, J. Chen, D. Duvenaud *et al.*, "Alignment faking in large language models," *arXiv preprint arXiv:2412.14093*, 2024.

[360] A. Pan, J. S. Chan, A. Zou, N. Li, S. Basart, T. Woodside, H. Zhang, S. Emmons, and D. Hendrycks, "Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark," in *International conference on machine learning*. PMLR, 2023, pp. 26 837–26 867.

[361] L. Vaugrante, F. Carlon, M. Menke, and T. Hagendorff, "Compromising honesty and harmlessness in language models via deception attacks," *arXiv preprint arXiv:2502.08301*, 2025.

[362] J. Ji, K. Wang, T. Qiu, B. Chen, J. Zhou, C. Li, H. Lou, and Y. Yang, "Language models resist alignment," *arXiv preprint arXiv:2406.06144*, 2024.

[363] L. Bürger, F. A. Hamprecht, and B. Nadler, "Truth is universal: Robust detection of lies in llms," *Advances in Neural Information Processing Systems*, vol. 37, pp. 138 393–138 431, 2024.

[364] OpenAI, "Detecting misbehavior in frontier reasoning models," https://openai.com/index/chain-of-thought-monitoring/, Mar. 2025, accessed: 2025-05-14.

[365] T. Everitt, V. Krakovna, L. Orseau, M. Hutter, and S. Legg, "Reinforcement learning with a corrupted reward channel," *arXiv preprint arXiv:1705.08417*, 2017.

[366] S. Zhuang and D. Hadfield-Menell, "Consequences of misaligned ai," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 763–15 773, 2020.

[367] V. Krakovna, J. Uesato, V. Mikulik, M. Rahtz, T. Everitt, R. Kumar, Z. Kenton, J. Leike, and S. Legg, "Specification gaming: the flip side of ai ingenuity," 2020, accessed: 2025-03-30. [Online]. Available: https://deepmind.google/discover/blog/specification-gaming-the-flip-side-of-ai-ingenuity/

[368] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016.

[369] L. Weng, "Reward hacking in reinforcement learning," 2024, accessed: 2025-03-30. [Online]. Available: https://lilianweng.github.io/posts/2024-11-28-reward-hacking

[370] T. Everitt, M. Hutter, R. Kumar, and V. Krakovna, "Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective," *Synthese*, vol. 198, no. Suppl 27, pp. 6435–6467, 2021.

[371] J. Skalse, N. Howe, D. Krasheninnikov, and D. Krueger, "Defining and characterizing reward gaming," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9460–9471, 2022.

[372] S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire *et al.*, "Open problems and fundamental limitations of reinforcement learning from human feedback," *arXiv preprint arXiv:2307.15217*, 2023.

[373] L. Gao, J. Schulman, and J. Hilton, "Scaling laws for reward model overoptimization," in *International Conference on Machine Learning*. PMLR, 2023, pp. 10 835–10 866.

[374] E. Perez, S. Ringer, K. Lukosiute, K. Nguyen, E. Chen, S. Heiner, C. Pettit, C. Olsson, S. Kundu, S. Kadavath *et al.*, "Discovering language model behaviors with model-written evaluations," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 13 387–13 434.

[375] C. Denison, M. MacDiarmid, F. Barez, D. Duvenaud, S. Kravec, S. Marks, N. Schiefer, R. Soklaski, A. Tamkin, J. Kaplan *et al.*, "Sycophancy to subterfuge: Investigating reward-tampering in large language models," *arXiv preprint arXiv:2406.10162*, 2024.

[376] P. Singhal, T. Goyal, J. Xu, and G. Durrett, "A long way to go: Investigating length correlations in rlhf," *arXiv preprint arXiv:2310.03716*, 2023.

[377] F. Bianchi, M. Suzgun, G. Attanasio, P. Röttger, D. Jurafsky, T. Hashimoto, and J. Zou, "Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions," *arXiv preprint arXiv:2309.07875*, 2023.

[378] M. Tegmark and S. Omohundro, "Provably safe systems: the only path to controllable agi," *arXiv preprint arXiv:2309.01933*, 2023.

[379] D. Dalrymple, J. Skalse, Y. Bengio, S. Russell, M. Tegmark, S. Seshia, S. Omohundro, C. Szegedy, B. Goldhaber, N. Ammann *et al.*, "Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems," *arXiv preprint arXiv:2405.06624*, 2024.

[380] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[381] R. Xu, Z. Zhou, T. Zhang, Z. Qi, S. Yao, K. Xu, W. Xu, and H. Qiu, "Walking in others' shoes: How perspective-taking guides large language models in reducing toxicity and bias," *arXiv preprint arXiv:2407.15366*, 2024.

[382] D. Acemoglu and P. Restrepo, "Artificial intelligence, automation, and work," in *The economics of artificial intelligence: An agenda*. University of Chicago Press, 2018, pp. 197–236.

[383] J. Mökander, J. Schuett, H. R. Kirk, and L. Floridi, "Auditing large language models: a three-layered approach," *AI and Ethics*, vol. 4, no. 4, pp. 1085–1115, 2024.

[384] M. Anderljung, J. Barnhart, A. Korinek, J. Leung, C. O'Keefe, J. Whittlestone, S. Avin, M. Brundage, J. Bullock, D. Cass-Beggs *et al.*, "Frontier ai regulation: Managing emerging risks to public safety," *arXiv preprint arXiv:2307.03718*, 2023.

[385] A. Mannes, "Governance, risk, and artificial intelligence," *Ai Magazine*, vol. 41, no. 1, pp. 61–69, 2020.

[386] L. Koessler and J. Schuett, "Risk assessment at agi

companies: A review of popular risk assessment techniques from other safety-critical industries," *arXiv preprint arXiv:2307.08823*, 2023.

[387] J. Schuett, N. Dreksler, M. Anderljung, D. McCaffary, L. Heim, E. Bluemke, and B. Garfinkel, "Towards best practices in agi safety and governance: A survey of expert opinion," *arXiv preprint arXiv:2305.07153*, 2023.

[388] L. Ho, J. Barnhart, R. Trager, Y. Bengio, M. Brundage, A. Carnegie, R. Chowdhury, A. Dafoe, G. Hadfield, M. Levi *et al.*, "International institutions for advanced ai," *arXiv preprint arXiv:2307.04699*, 2023.

[389] M. M. Maas, "Aligning ai regulation to sociotechnical change," in *The Oxford Handbook of AI Governance*, 2022.

[390] M. Kinniment, L. J. K. Sato, H. Du, B. Goodrich, M. Hasin, L. Chan, L. H. Miles, T. R. Lin, H. Wijk, J. Burget *et al.*, "Evaluating language-model agents on realistic autonomous tasks," *arXiv preprint arXiv:2312.11671*, 2023.

[391] J. Tallberg, E. Erman, M. Furendal, J. Geith, M. Klamberg, and M. Lundgren, "The global governance of artificial intelligence: Next steps for empirical and normative research," *International Studies Review*, vol. 25, no. 3, p. viad040, 2023.

[392] OECD, "OECD Principles on Artificial Intelligence," https://oecd.ai/en/ai-principles, 2019.

[393] UNESCO, "Recommendation on the Ethics of Artificial Intelligence," https://unesdoc.unesco.org/ark:/48223/pf0000381137, 2021.

[394] E. Seger, N. Dreksler, R. Moulange, E. Dardaman, J. Schuett, K. Wei, C. Winter, M. Arnold, S. Ó. hÉigeartaigh, A. Korinek *et al.*, "Open-sourcing highly capable foundation models: An evaluation of risks, benefits, and alternative methods for pursuing open-source objectives," *arXiv preprint arXiv:2311.09227*, 2023.

[395] F. Urbina, F. Lentzos, C. Invernizzi, and S. Ekins, "Dual use of artificial-intelligence-powered drug discovery," *Nature machine intelligence*, vol. 4, no. 3, pp. 189–191, 2022.

[396] Meta, "Meta and Microsoft introduce the next generation of Llama," https://ai.meta.com/blog/llama-2, 2023.

[397] E. Mostaque, "Democratizing ai, stable diffusion & generative models," https://exchange.scale.com/public/videos/emad-mostaque-stability-ai-stable-diffusion-open-source, 2022.

[398] J. A. Goldstein, G. Sastry, M. Musser, R. DiResta, M. Gentzel, and K. Sedova, "Generative language models and automated influence operations: Emerging threats and potential mitigations," *arXiv preprint arXiv:2301.04246*, 2023.

[399] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps *et al.*, "Release strategies and the social impacts of language models," *arXiv preprint arXiv:1908.09203*, 2019.

[400] P. Chavez, "An ai challenge: Balancing open and closed systems," https://cepa.org/article/an-ai-challenge-balancing-open-and-closed-systems, 2023.

[401] N. Zhang, Y. Yao, B. Tian, P. Wang, S. Deng, M. Wang, Z. Xi, S. Mao, J. Zhang, Y. Ni *et al.*, "A comprehensive study of knowledge editing for large language models," *arXiv preprint arXiv:2401.01286*, 2024.

[402] J. Fang, H. Jiang, K. Wang, Y. Ma, X. Wang, X. He, and T.-s. Chua, "Alphaedit: Null-space constrained knowledge editing for language models," *arXiv preprint arXiv:2410.02355*, 2024.

[403] Z. Zhang, Y. Zhou, X. Zhao, T. Che, and L. Lyu, "Prompt certified machine unlearning with randomized gradient smoothing and quantization," *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 433–13 455, 2022.

[404] T. Che, Y. Zhou, Z. Zhang, L. Lyu, J. Liu, D. Yan, D. Dou, and J. Huan, "Fast federated machine unlearning with nonlinear functional theory," in *International conference on machine learning*. PMLR, 2023, pp. 4241–4268.

[405] W. Wang, Z. Tian, C. Zhang, and S. Yu, "Machine unlearning: A comprehensive survey," *arXiv preprint arXiv:2405.07406*, 2024.

[406] S. Liu, Y. Yao, J. Jia, S. Casper, N. Baracaldo, P. Hase, Y. Yao, C. Y. Liu, X. Xu, H. Li *et al.*, "Rethinking machine unlearning for large language models," *Nature Machine Intelligence*, pp. 1–14, 2025.

[407] Y. Yao, X. Xu, and Y. Liu, "Large language model unlearning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 105 425–105 475, 2025.

[408] C. Ding, J. Wu, Y. Yuan, J. Lu, K. Zhang, A. Su, X. Wang, and X. He, "Unified parameter-efficient unlearning for llms," *arXiv preprint arXiv:2412.00383*, 2024.

[409] Z. Li, H. Jiang, H. Chen, B. Bi, Z. Zhou, F. Sun, J. Fang, and X. Wang, "Reinforced lifelong editing for language models," *arXiv preprint arXiv:2502.05759*, 2025.

[410] E. Mitchell, C. Lin, A. Bosselut, C. Finn, and C. D. Manning, "Fast model editing at scale," *arXiv preprint arXiv:2110.11309*, 2021.

[411] N. De Cao, W. Aziz, and I. Titov, "Editing factual knowledge in language models," *arXiv preprint arXiv:2104.08164*, 2021.

[412] P. Wang, Z. Li, N. Zhang, Z. Xu, Y. Yao, Y. Jiang, P. Xie, F. Huang, and H. Chen, "Wise: Rethinking the knowledge memory for lifelong model editing of large language models," *arXiv preprint arXiv:2405.14768*, 2024.

[413] T. Hartvigsen, S. Sankaranarayanan, H. Palangi, Y. Kim, and M. Ghassemi, "Aging with grace: Lifelong model editing with discrete key-value adaptors," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[414] H. Jiang, J. Fang, N. Zhang, G. Ma, M. Wan, X. Wang, X. He, and T.-s. Chua, "Anyedit: Edit any knowledge encoded in language models," *arXiv preprint arXiv:2502.05628*, 2025.

[415] H. Jiang, J. Fang, T. Zhang, A. Zhang, R. Wang, T. Liang, and X. Wang, "Neuron-level sequential editing for large language models," *arXiv preprint arXiv:2410.04045*, 2024.

[416] K. Meng, D. Bau, A. Andonian, and Y. Belinkov,

"Locating and editing factual associations in gpt," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 359–17 372, 2022.

[417] A. Prasad, P. Hase, X. Zhou, and M. Bansal, "Grips: Gradient-free, edit-based instruction search for prompting large language models," *arXiv preprint arXiv:2203.07281*, 2022.

[418] G. Gangadhar and K. Stratos, "Model editing by standard fine-tuning," *arXiv preprint arXiv:2402.11078*, 2024.

[419] E. Mitchell, C. Lin, A. Bosselut, C. D. Manning, and C. Finn, "Memory-based model editing at scale," in *International Conference on Machine Learning*. PMLR, 2022, pp. 15 817–15 831.

[420] Y. Yao, P. Wang, B. Tian, S. Cheng, Z. Li, S. Deng, H. Chen, and N. Zhang, "Editing large language models: Problems, methods, and opportunities," *arXiv preprint arXiv:2305.13172*, 2023.

[421] K. Meng, A. S. Sharma, A. Andonian, Y. Belinkov, and D. Bau, "Mass-editing memory in a transformer," *arXiv preprint arXiv:2210.07229*, 2022.

[422] J.-C. Gu, H.-X. Xu, J.-Y. Ma, P. Lu, Z.-H. Ling, K.-W. Chang, and N. Peng, "Model editing can hurt general abilities of large language models," *arXiv e-prints*, pp. arXiv–2401, 2024.

[423] X. Li, S. Li, S. Song, J. Yang, J. Ma, and J. Yu, "Pmet: Precise model editing in a transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 18 564–18 572.

[424] M. Zhang, X. Ye, Q. Liu, P. Ren, S. Wu, and Z. Chen, "Knowledge graph enhanced large language model editing," *arXiv preprint arXiv:2402.13593*, 2024.

[425] C. Chen, B. Huang, Z. Li, Z. Chen, S. Lai, X. Xu, J.-C. Gu, J. Gu, H. Yao, C. Xiao *et al.*, "Can editing llms inject harm?" *arXiv preprint arXiv:2407.20224*, 2024.

[426] M. Wang, N. Zhang, Z. Xu, Z. Xi, S. Deng, Y. Yao, Q. Zhang, L. Yang, J. Wang, and H. Chen, "Detoxifying large language models via knowledge editing," *arXiv preprint arXiv:2403.14472*, 2024.

[427] C. Zheng, L. Li, Q. Dong, Y. Fan, Z. Wu, J. Xu, and B. Chang, "Can we edit factual knowledge by in-context learning?" *arXiv preprint arXiv:2305.12740*, 2023.

[428] Y. Li, T. Li, K. Chen, J. Zhang, S. Liu, W. Wang, T. Zhang, and Y. Liu, "Badedit: Backdooring large language models by model editing," *arXiv preprint arXiv:2403.13355*, 2024.

[429] K. Grimes, M. Christiani, D. Shriver, and M. Connor, "Concept-rot: Poisoning concepts in large language models with model editing," *arXiv preprint arXiv:2412.13341*, 2024.

[430] X. Wu, J. Li, M. Xu, W. Dong, S. Wu, C. Bian, and D. Xiong, "Depn: Detecting and editing privacy neurons in pretrained language models," *arXiv preprint arXiv:2310.20138*, 2023.

[431] X. Li, Z. Li, Y. Kosuga, Y. Yoshida, and V. Bian, "Precision knowledge editing: Enhancing safety in large language models," *arXiv preprint arXiv:2410.03772*, 2024.

[432] X. Hu, D. Li, B. Hu, Z. Zheng, Z. Liu, and M. Zhang, "Separate the wheat from the chaff: Model deficiency unlearning via parameter-efficient module operation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 18 252–18 260.

[433] T. Yang, L. Dai, Z. Liu, X. Wang, M. Jiang, Y. Tian, and X. Zhang, "Cliperase: Efficient unlearning of visual-textual associations in clip," *arXiv preprint arXiv:2410.23330*, 2024.

[434] R. Gandikota, J. Materzynska, J. Fiotto-Kaufman, and D. Bau, "Erasing concepts from diffusion models," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2426–2436, 2023.

[435] E. Zhang, K. Wang, X. Xu, Z. Wang, and H. Shi, "Forget-me-not: Learning to forget in text-to-image diffusion models," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1755–1764, 2023.

[436] C. Fan, J. Liu, Y. Zhang, D. Wei, E. Wong, and S. Liu, "Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation," *ArXiv*, vol. abs/2310.12508, 2023.

[437] Z. Huang, X. Cheng, J. Zheng, H. Wang, Z. He, T. Li, and X. Huang, "Unified gradient-based machine unlearning with remain geometry enhancement," *ArXiv*, vol. abs/2409.19732, 2024.

[438] A. Blanco-Justicia, J. Domingo-Ferrer, N. M. Jebreel, B. Manzanares-Salor, and D. Sánchez, "Unlearning in large language models: We are not there yet," *Computer*, vol. 58, no. 1, pp. 97–100, 2025.

[439] S. Dai, C. Xu, S. Xu, L. Pang, Z. Dong, and J. Xu, "Bias and unfairness in information retrieval systems: New challenges in the llm era," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 6437–6447.

[440] G. Nicolas and A. Caliskan, "A taxonomy of stereotype content in large language models," *arXiv preprint arXiv:2408.00162*, 2024.

[441] S. Wang, R. Li, X. Chen, Y. Yuan, D. F. Wong, and M. Yang, "Exploring the impact of personality traits on llm bias and toxicity," *arXiv preprint arXiv:2502.12566*, 2025.

[442] A. Liu, Q. Sheng, and X. Hu, "Preventing and detecting misinformation generated by large language models," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 3001–3004.

[443] Q. Zhang, H. Qiu, D. Wang, H. Qian, Y. Li, T. Zhang, and M. Huang, "Understanding the dark side of llms' intrinsic self-correction," *arXiv preprint arXiv:2412.14959*, 2024.

[444] R. Xu, B. Lin, S. Yang, T. Zhang, W. Shi, T. Zhang, Z. Fang, W. Xu, and H. Qiu, "The earth is flat because...: Investigating llms' belief towards misinformation via persuasive conversation," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 16 259–16 303.

[445] Z. Liu, G. Dou, Z. Tan, Y. Tian, and M. Jiang, "Machine unlearning in generative ai: A survey," *arXiv preprint arXiv:2407.20516*, 2024.

[446] Y. Qu, M. Ding, N. Sun, K. Thilakarathna, T. Zhu, and D. Niyato, "The frontier of data erasure: Machine

unlearning for large language models," *arXiv preprint arXiv:2403.15779*, 2024.

[447] A. Blanco-Justicia, N. Jebreel, B. Manzanares-Salor, D. Sánchez, J. Domingo-Ferrer, G. Collell, and K. Eeik Tan, "Digital forgetting in large language models: A survey of unlearning methods," *Artificial Intelligence Review*, vol. 58, no. 3, p. 90, 2025.

[448] N. Li, C. Zhou, Y. Gao, H. Chen, Z. Zhang, B. Kuang, and A. Fu, "Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[449] C. Gao, L. Wang, C. Weng, X. Wang, and Q. Zhu, "Practical unlearning for large language models," *arXiv preprint arXiv:2407.10223*, 2024.

[450] P. Thaker, S. Hu, N. Kale, Y. Maurya, Z. S. Wu, and V. Smith, "Position: Llm unlearning benchmarks are weak measures of progress," *arXiv preprint arXiv:2410.02879*, 2024.

[451] K. Zhao, M. Kurmanji, G.-O. Bărbulescu, E. Triantafillou, and P. Triantafillou, "What makes unlearning hard and what to do about it," *Advances in Neural Information Processing Systems*, vol. 37, pp. 12 293–12 333, 2025.

[452] W. Wang, M. Zhang, X. Ye, Z. Ren, Z. Chen, and P. Ren, "Uipe: Enhancing llm unlearning by removing knowledge related to forgetting targets," *arXiv preprint arXiv:2503.04693*, 2025.

[453] H. Wang, Y. Jing, H. Sun, Y. Wang, J. Wang, J. Liao, and D. Tao, "Erasing without remembering: Safeguarding knowledge forgetting in large language models," *arXiv preprint arXiv:2502.19982*, 2025.

[454] T. Tran, R. Liu, and L. Xiong, "Tokens for learning, tokens for unlearning: Mitigating membership inference attacks in large language models via dual-purpose training," *arXiv preprint arXiv:2502.19726*, 2025.

[455] H. Xu, N. Zhao, L. Yang, S. Zhao, S. Deng, M. Wang, B. Hooi, N. Oo, H. Chen, and N. Zhang, "Relearn: Unlearning via learning for large language models," *arXiv preprint arXiv:2502.11190*, 2025.

[456] Q. Zhang, H. Qiu, D. Wang, Y. Li, T. Zhang, W. Zhu, H. Weng, L. Yan, and C. Zhang, "Large scale knowledge washing," in *The Thirteenth International Conference on Learning Representations*, 2025.

[457] A. Thudi, H. Jia, I. Shumailov, and N. Papernot, "On the necessity of auditable algorithmic definitions for machine unlearning," in *31st USENIX security symposium (USENIX Security 22)*, 2022, pp. 4007–4022.

[458] S. Goel, A. Prabhu, P. Torr, P. Kumaraguru, and A. Sanyal, "Corrective machine unlearning," *Transactions on Machine Learning Research*.

[459] A. Thudi, G. Deza, V. Chandrasekaran, and N. Papernot, "Unrolling sgd: Understanding factors influencing machine unlearning," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2022, pp. 303–319.

[460] B. Liu, Q. Liu, and P. Stone, "Continual learning and private unlearning," in *Conference on Lifelong Learning Agents*. PMLR, 2022, pp. 243–254.

[461] Q. P. Nguyen, B. K. H. Low, and P. Jaillet, "Variational bayesian unlearning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 025–16 036, 2020.

[462] L. Wang, T. Chen, W. Yuan, X. Zeng, K.-F. Wong, and H. Yin, "Kga: A general machine unlearning framework based on knowledge gap alignment," *arXiv preprint arXiv:2305.06535*, 2023.

[463] Y. Liu, Y. Zhang, T. Jaakkola, and S. Chang, "Revisiting who's harry potter: Towards targeted unlearning from a causal intervention perspective," *arXiv preprint arXiv:2407.16997*, 2024.

[464] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, and J. Z. Kolter, "Tofu: A task of fictitious unlearning for llms," *arXiv preprint arXiv:2401.06121*, 2024.

[465] R. Zhang, L. Lin, Y. Bai, and S. Mei, "Negative preference optimization: From catastrophic collapse to effective unlearning," *arXiv preprint arXiv:2404.05868*, 2024.

[466] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[467] J. Huo, Y. Yan, X. Zheng, Y. Lyu, X. Zou, Z. Wei, and X. Hu, "Mmunlearner: Reformulating multimodal machine unlearning in the era of multimodal large language models," *arXiv preprint arXiv:2502.11051*, 2025.

[468] J. Li, Q. Wei, C. Zhang, G. Qi, M. Du, Y. Chen, and S. Bi, "Single image unlearning: Efficient machine unlearning in multimodal large language models," *arXiv preprint arXiv:2405.12523*, 2024.

[469] S. Xing, F. Zhao, Z. Wu, T. An, W. Chen, C. Li, J. Zhang, and X. Dai, "Efuf: Efficient fine-grained unlearning framework for mitigating hallucinations in multimodal large language models," *ArXiv*, vol. abs/2402.09801, 2024.

[470] T. Chakraborty, E. Shayegani, Z. Cai, N. B. Abu-Ghazaleh, M. S. Asif, Y. Dong, A. K. Roy-Chowdhury, and C. Song, "Cross-modal safety alignment: Is textual unlearning all you need?" *ArXiv*, vol. abs/2406.02575, 2024.

[471] J. Chen, Z. Deng, K. Zheng, Y. Yan, S. Liu, P. Wu, P. Jiang, J. Liu, and X. Hu, "Safeeraser: Enhancing safety in multimodal large language models through multimodal machine unlearning," *arXiv preprint arXiv:2502.12520*, 2025.

[472] G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, and A. Farhadi, "Editing models with task arithmetic," *arXiv preprint arXiv:2212.04089*, 2022.

[473] D. Jung, J. Seo, J. Lee, C. Park, and H. Lim, "Come: An unlearning-based approach to conflict-free model editing," *arXiv preprint arXiv:2502.15826*, 2025.

[474] B. Zhang, Z. Chen, Z. Zheng, J. Li, and H. Chen, "Resolving editing-unlearning conflicts: A knowledge codebook framework for large language model updating," *arXiv preprint arXiv:2502.00158*, 2025.

[475] R. Eldan and M. Russinovich, "Who's harry potter? approximate unlearning in llms," *arXiv preprint arXiv:2310.02238*, 2023.

[476] N. Li, A. Pan, A. Gopal, S. Yue, D. Berrios, A. Gatti, J. D. Li, A.-K. Dombrowski, S. Goel, L. Phan *et al.*, "The wmdp benchmark: Measuring and re-

ducing malicious use with unlearning," *arXiv preprint arXiv:2403.03218*, 2024.

[477] M. Pawelczyk, S. Neel, and H. Lakkaraju, "In-context unlearning: Language models as few shot unlearners," *arXiv preprint arXiv:2310.07579*, 2023.

[478] P. Thaker, Y. Maurya, S. Hu, Z. S. Wu, and V. Smith, "Guardrail baselines for unlearning in llms," *arXiv preprint arXiv:2403.03329*, 2024.

[479] J. Ren, Z. Dai, X. Tang, H. Liu, J. Zeng, Z. Li, R. Goutam, S. Wang, Y. Xing, and Q. He, "A general framework to enhance fine-tuning-based llm unlearning," *arXiv preprint arXiv:2502.17823*, 2025.

[480] X. Zhao, W. Cai, T. Shi, D. Huang, L. Lin, S. Mei, and D. Song, "Improving llm safety alignment with dual-objective optimization," *arXiv preprint arXiv:2503.03710*, 2025.

[481] S. Takashiro, T. Kojima, A. Gambardella, Q. Cao, Y. Iwasawa, and Y. Matsuo, "Answer when needed, forget when not: Language models pretend to forget via in-context knowledge unlearning," *arXiv preprint arXiv:2410.00382*, 2024.

[482] A. Muresanu, A. Thudi, M. R. Zhang, and N. Papernot, "Unlearnable algorithms for in-context learning," *arXiv preprint arXiv:2402.00751*, 2024.

[483] Y. Zhou, X. Li, Q. Wang, and J. Shen, "Visual in-context learning for large vision-language models," *arXiv preprint arXiv:2402.11574*, 2024.

[484] Z. Liu, G. Dou, X. Yuan, C. Zhang, Z. Tan, and M. Jiang, "Modality-aware neuron pruning for unlearning in multimodal large language models," *arXiv preprint arXiv:2502.15910*, 2025.

[485] N. Yang, M. Kim, S. Yoon, J. Shin, and K. Jung, "Faithun: Toward faithful forgetting in language models by investigating the interconnectedness of knowledge," *arXiv preprint arXiv:2502.19207*, 2025.

[486] A. Ramakrishna, Y. Wan, X. Jin, K.-W. Chang, Z. Bu, B. Vinzamuri, V. Cevher, M. Hong, and R. Gupta, "Lume: Llm unlearning with multitask evaluations," *arXiv preprint arXiv:2502.15097*, 2025.

[487] Y. Lang, K. Guo, Y. Huang, Y. Zhou, H. Zhuang, T. Yang, Y. Su, and X. Zhang, "Beyond single-value metrics: Evaluating and enhancing llm unlearning with cognitive diagnosis," *arXiv preprint arXiv:2502.13996*, 2025.

[488] Q. Wang, J. P. Zhou, Z. Zhou, S. Shin, B. Han, and K. Q. Weinberger, "Rethinking llm unlearning objectives: A gradient perspective and go beyond," *arXiv preprint arXiv:2502.19301*, 2025.

[489] M. Khoriaty, A. Shportko, G. Mercier, and Z. Wood-Doughty, "Don't forget it! conditional sparse autoencoder clamping works for unlearning," *arXiv preprint arXiv:2503.11127*, 2025.

[490] J. Cheng and H. Amiri, "Mu-bench: A multitask multimodal benchmark for machine unlearning," *arXiv preprint arXiv:2406.14796*, 2024.

[491] V. Patil, Y.-L. Sung, P. Hase, J. Peng, T. Chen, and M. Bansal, "Unlearning sensitive information in multimodal llms: Benchmark and attack-defense evaluation," *Transactions on Machine Learning Research*.

[492] Y. Ma, J. Wang, F. Wang, S. Ma, J. Li, X. Li, F. Huang, L. Sun, B. Li, Y. Choi *et al.*, "Benchmarking vision language model unlearning via fictitious facial identity dataset," *arXiv preprint arXiv:2411.03554*, 2024.

[493] S. Moon, M. Lee, S. Park, and D. Kim, "Holistic unlearning benchmark: A multi-faceted evaluation for text-to-image diffusion model unlearning," *arXiv preprint arXiv:2410.05664*, 2024.

[494] D. Sanyal and M. Mandal, "Alu: Agentic llm unlearning," *arXiv preprint arXiv:2502.00406*, 2025.

[495] J. Cheng and H. Amiri, "Tool unlearning for tool-augmented llms," *arXiv preprint arXiv:2502.01083*, 2025.

[496] H. Liu, P. Xiong, T. Zhu, and S. Y. Philip, "A survey on machine unlearning: Techniques and new emerged privacy risks," *Journal of Information Security and Applications*, vol. 90, p. 104010, 2025.

[497] S. Qureshi, T. Shaik, X. Tao, H. Xie, L. Li, J. Yong, and X. Jia, "Exploring incremental unlearning: Techniques, challenges, and future directions," *arXiv preprint arXiv:2502.16708*, 2025.

[498] J. Geng, Q. Li, H. Woisetschlaeger, Z. Chen, Y. Wang, P. Nakov, H.-A. Jacobsen, and F. Karray, "A comprehensive survey of machine unlearning techniques for large language models," *arXiv preprint arXiv:2503.01854*, 2025.

[499] X. He, C. Chen, L. Lyu, and Q. Xu, "Extracted bert model leaks more information than you think!" in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*. Association for Computational Linguistics, 2022, pp. 1530–1537.

[500] X. He, Q. Xu, L. Lyu, F. Wu, and C. Wang, "Protecting intellectual property of language generation apis with lexical watermark," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 10 758–10 766.

[501] X. He, Q. Xu, Y. Zeng, L. Lyu, F. Wu, J. Li, and R. Jia, "Cater: Intellectual property protection on text generation apis via conditional watermarks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5431–5445, 2022.

[502] W. Peng, J. Yi, F. Wu, S. Wu, B. B. Zhu, L. Lyu, B. Jiao, T. Xu, G. Sun, and X. Xie, "Are you copying my model? protecting the copyright of large language models for eaas via backdoor watermark," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 7653–7668.

[503] N. Carlini, D. Paleka, K. D. Dvijotham, T. Steinke, J. Hayase, A. F. Cooper, K. Lee, M. Jagielski, M. Nasr, A. Conmy *et al.*, "Stealing part of a production language model," *arXiv preprint arXiv:2403.06634*, 2024.

[504] M. Finlayson, X. Ren, and S. Swayamdipta, "Logits of api-protected llms leak proprietary information," *arXiv preprint arXiv:2403.09539*, 2024.

[505] S. Zanella-Beguelin, S. Tople, A. Paverd, and B. Köpf, "Grey-box extraction of natural language models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 278–12 286.

[506] E. Horwitz, J. Kahana, and Y. Hoshen, "Recovering the pre-fine-tuning weights of generative models," *arXiv preprint arXiv:2402.10208*, 2024.

[507] Z. Li, C. Wang, P. Ma, C. Liu, S. Wang, D. Wu, C. Gao, and Y. Liu, "On extracting specialized code abilities from large language models: A feasibility study," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–13.

[508] A. Liu and A. Moitra, "Model stealing for any low-rank language model," *arXiv preprint arXiv:2411.07536*, 2024.

[509] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer, "Detecting pretraining data from large language models," *arXiv preprint arXiv:2310.16789*, 2023.

[510] J. Zhang, J. Sun, E. Yeats, Y. Ouyang, M. Kuo, J. Zhang, H. F. Yang, and H. Li, "Min-k%++: Improved baseline for detecting pre-training data from large language models," *arXiv preprint arXiv:2404.02936*, 2024.

[511] D. Das, J. Zhang, and F. Tramèr, "Blind baselines beat membership inference attacks for foundation models," *arXiv preprint arXiv:2406.16201*, 2024.

[512] P. Maini, H. Jia, N. Papernot, and A. Dziedzic, "Llm dataset inference: Did you train on my dataset?" *Advances in Neural Information Processing Systems*, vol. 37, pp. 124 069–124 092, 2024.

[513] A. V. Duarte, X. Zhao, A. L. Oliveira, and L. Li, "De-cop: Detecting copyrighted content in language models training data," *arXiv preprint arXiv:2402.09910*, 2024.

[514] R. Xie, J. Wang, R. Huang, M. Zhang, R. Ge, J. Pei, N. Z. Gong, and B. Dhingra, "Recall: Membership inference via relative conditional log-likelihoods," *arXiv preprint arXiv:2406.15968*, 2024.

[515] F. Galli, L. Melis, and T. Cucinotta, "Noisy neighbors: Efficient membership inference attacks against llms," *arXiv preprint arXiv:2406.16565*, 2024.

[516] H. Mozaffari and V. J. Marathe, "Semantic membership inference attack against large language models," *arXiv preprint arXiv:2406.10218*, 2024.

[517] M. Meeus, S. Jain, M. Rei, and Y.-A. de Montjoye, "Did the neurons read your book? document-level membership inference for large language models," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 2369–2385.

[518] M. Meeus, I. Shilov, M. Faysse, and Y.-A. De Montjoye, "Copyright traps for large language models," *arXiv preprint arXiv:2402.09363*, 2024.

[519] H. Puerto, M. Gubri, S. Yun, and S. J. Oh, "Scaling up membership inference: When and how attacks succeed on large language models," *arXiv preprint arXiv:2411.00154*, 2024.

[520] M. Anderson, G. Amit, and A. Goldsteen, "Is my data in your retrieval database? membership inference attacks against retrieval augmented generation," *arXiv preprint arXiv:2405.20446*, 2024.

[521] Y. Li, G. Liu, C. Wang, and Y. Yang, "Generating is believing: Membership inference attacks against retrieval-augmented generation," *arXiv preprint arXiv:2406.19234*, 2024.

[522] R. Wen, Z. Li, M. Backes, and Y. Zhang, "Membership inference attacks against in-context learning," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 3481–3495.

[523] H. Duan, A. Dziedzic, M. Yaghini, N. Papernot, and F. Boenisch, "On the privacy risk of in-context learning," *arXiv preprint arXiv:2411.10512*, 2024.

[524] Y. Wen, L. Marchyok, S. Hong, J. Geiping, T. Goldstein, and N. Carlini, "Privacy backdoors: Enhancing membership inference through poisoning pre-trained models," *arXiv preprint arXiv:2404.01231*, 2024.

[525] R. Wen, T. Wang, M. Backes, Y. Zhang, and A. Salem, "Last one standing: A comparative analysis of security and privacy of soft prompt tuning, lora, and in-context learning," *arXiv preprint arXiv:2310.11397*, 2023.

[526] S. Balloccu, P. Schmidtová, M. Lango, and O. Dušek, "Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms," *arXiv preprint arXiv:2402.03927*, 2024.

[527] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, and T. Jiang, "Membership inference attacks against fine-tuned large language models via self-prompt calibration," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[528] H. Li, G. Deng, Y. Liu, K. Wang, Y. Li, T. Zhang, Y. Liu, G. Xu, G. Xu, and H. Wang, "Digger: Detecting copyright content mis-usage in large language model training," *arXiv preprint arXiv:2401.00676*, 2024.

[529] A. Naseh and N. Mireshghallah, "Synthetic data can mislead evaluations: Membership inference as machine text detection," *arXiv preprint arXiv:2501.11786*, 2025.

[530] Z. Liao and H. Sun, "Amplegcg: Learning a universal and transferable generative model of adversarial suffixes for jailbreaking both open and closed llms," *arXiv preprint arXiv:2404.07921*, 2024.

[531] X. Jia, T. Pang, C. Du, Y. Huang, J. Gu, Y. Liu, X. Cao, and M. Lin, "Improved techniques for optimization-based jailbreaking on large language models," *arXiv preprint arXiv:2405.21018*, 2024.

[532] Y. Zhang and Z. Wei, "Boosting jailbreak attack with momentum," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[533] Y. Zhao, W. Zheng, T. Cai, D. Xuan Long, K. Kawaguchi, A. Goyal, and M. Q. Shieh, "Accelerating greedy coordinate gradient and general prompt optimization via probe sampling," *Advances in Neural Information Processing Systems*, vol. 37, pp. 53 710–53 731, 2024.

[534] X. Liu, N. Xu, M. Chen, and C. Xiao, "Autodan: Generating stealthy jailbreak prompts on aligned large language models," *arXiv preprint arXiv:2310.04451*, 2023.

[535] S. Zhu, R. Zhang, B. An, G. Wu, J. Barrow, Z. Wang, F. Huang, A. Nenkova, and T. Sun, "Autodan: interpretable gradient-based adversarial attacks on large language models," *arXiv preprint arXiv:2310.15140*, 2023.

[536] A. Mehrotra, M. Zampetakis, P. Kassianik, B. Nelson, H. Anderson, Y. Singer, and A. Karbasi, "Tree of attacks: Jailbreaking black-box llms automatically," *Advances in Neural Information Processing Systems*, vol. 37, pp. 61 065–61 105, 2024.

[537] C. Sitawarin, N. Mu, D. Wagner, and A. Araujo,

"Pal: Proxy-guided black-box attack on large language models," *arXiv preprint arXiv:2402.09674*, 2024.

[538] G. Deng, Y. Liu, Y. Li, K. Wang, Y. Zhang, Z. Li, H. Wang, T. Zhang, and Y. Liu, "Masterkey: Automated jailbreak across multiple large language model chatbots," *arXiv preprint arXiv:2307.08715*, 2023.

[539] X. Liu, P. Li, E. Suh, Y. Vorobeychik, Z. Mao, S. Jha, P. McDaniel, H. Sun, B. Li, and C. Xiao, "Autodanturbo: A lifelong agent for strategy self-exploration to jailbreak llms," *arXiv preprint arXiv:2410.05295*, 2024.

[540] Y. Liu, X. He, M. Xiong, J. Fu, S. Deng, and B. Hooi, "Flipattack: Jailbreak llms via flipping," *arXiv preprint arXiv:2410.02832*, 2024.

[541] T. Wu, Z. Xue, Y. Liu, J. Zhang, B. Hooi, and S.-K. Ng, "Geneshift: Impact of different scenario shift on jailbreaking llm," 2025. [Online]. Available: https://arxiv.org/abs/2504.08104

[542] F. Perez and I. Ribeiro, "Ignore previous prompt: Attack techniques for language models," *arXiv preprint arXiv:2211.09527*, 2022.

[543] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection," in *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 2023, pp. 79–90.

[544] Y. Liu, G. Deng, Y. Li, K. Wang, Z. Wang, X. Wang, T. Zhang, Y. Liu, H. Wang, Y. Zheng *et al.*, "Prompt injection attack against llm-integrated applications," *arXiv preprint arXiv:2306.05499*, 2023.

[545] S. Toyer, O. Watkins, E. A. Mendes, J. Svegliato, L. Bailey, T. Wang, I. Ong, K. Elmaaroufi, P. Abbeel, T. Darrell *et al.*, "Tensor trust: Interpretable prompt injection attacks from an online game," *arXiv preprint arXiv:2311.01011*, 2023.

[546] J. Shi, Z. Yuan, Y. Liu, Y. Huang, P. Zhou, L. Sun, and N. Z. Gong, "Optimization-based prompt injection attack to llm-as-a-judge," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 660–674.

[547] X. Liu, Z. Yu, Y. Zhang, N. Zhang, and C. Xiao, "Automatic and universal prompt injection attacks against large language models," *arXiv preprint arXiv:2403.04957*, 2024.

[548] X. Liu, S. Jha, P. McDaniel, B. Li, and C. Xiao, "Autohijacker: Automatic indirect prompt injection against black-box llm agents."

[549] A. Al-Kaswan, M. Izadi, and A. Van Deursen, "Targeted attack on gpt-neo for the satml language model data extraction challenge," *arXiv preprint arXiv:2302.07735*, 2023.

[550] E. Su, A. Vellore, A. Chang, R. Mura, B. Nelson, P. Kassianik, and A. Karbasi, "Extracting memorized training data via decomposition," *arXiv preprint arXiv:2409.12367*, 2024.

[551] J. Huang, H. Shao, and K. C.-C. Chang, "Are large pre-trained language models leaking your personal information?" *arXiv preprint arXiv:2205.12628*, 2022.

[552] Z. Zhang, J. Wen, and M. Huang, "Ethicist: Targeted training data extraction through loss smoothed soft prompting and calibrated confidence estimation," *arXiv preprint arXiv:2307.04401*, 2023.

[553] K. K. Nakka, A. Frikha, R. Mendes, X. Jiang, and X. Zhou, "Pii-compass: Guiding llm training data extraction prompts towards the target pii via grounding," *arXiv preprint arXiv:2407.02943*, 2024.

[554] Z. Wang, R. Bao, Y. Wu, J. Taylor, C. Xiao, F. Zheng, W. Jiang, S. Gao, and Y. Zhang, "Unlocking memorization in large language models with dynamic soft prompting," *arXiv preprint arXiv:2409.13853*, 2024.

[555] J. G. Wang, J. Wang, M. Li, and S. Neel, "Pandora's white-box: Precise training data detection and extraction in large language models," *arXiv preprint arXiv:2402.17012*, 2024.

[556] Z. Sha and Y. Zhang, "Prompt stealing attacks against large language models," *arXiv preprint arXiv:2402.12959*, 2024.

[557] C. Zhang, J. X. Morris, and V. Shmatikov, "Extracting prompts by inverting llm outputs," *arXiv preprint arXiv:2405.15012*, 2024.

[558] Y. Yang, C. Li, Y. Jiang, X. Chen, H. Wang, X. Zhang, Z. Wang, and S. Ji, "Prsa: Prompt stealing attacks against large language models," *arXiv preprint arXiv:2402.19200*, 2024.

[559] Y. Zeng, H. Lin, J. Zhang, D. Yang, R. Jia, and W. Shi, "How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 14 322–14 350.

[560] X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang, "" do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 1671–1685.

[561] Z. Wang, W. Xie, B. Wang, E. Wang, Z. Gui, S. Ma, and K. Chen, "Foot in the door: Understanding large language model jailbreaking via cognitive psychology," *arXiv preprint arXiv:2402.15690*, 2024.

[562] M. Samvelyan, S. C. Raparthy, A. Lupu, E. Hambro, A. Markosyan, M. Bhatt, Y. Mao, M. Jiang, J. Parker-Holder, J. Foerster *et al.*, "Rainbow teaming: Open-ended generation of diverse adversarial prompts," *Advances in Neural Information Processing Systems*, vol. 37, pp. 69 747–69 786, 2024.

[563] H. Jin, R. Chen, A. Zhou, Y. Zhang, and H. Wang, "Guard: Role-playing to generate natural-language jailbreakings to test guideline adherence of large language models," *arXiv preprint arXiv:2402.03299*, 2024.

[564] Y. Yuan, W. Jiao, W. Wang, J.-t. Huang, P. He, S. Shi, and Z. Tu, "Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher," *arXiv preprint arXiv:2308.06463*, 2023.

[565] H. Lv, X. Wang, Y. Zhang, C. Huang, S. Dou, J. Ye, T. Gui, Q. Zhang, and X. Huang, "Codechameleon: Personalized encryption framework for jailbreaking large language models," *arXiv preprint arXiv:2402.16717*, 2024.

[566] F. Jiang, Z. Xu, L. Niu, Z. Xiang, B. Ramasubramanian, B. Li, and R. Poovendran, "Artprompt: Ascii art-based jailbreak attacks against aligned llms," in *Proceedings*

of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 15 157–15 173.

[567] C. Anil, E. Durmus, N. Panickssery, M. Sharma, J. Benton, S. Kundu, J. Batson, M. Tong, J. Mu, D. Ford *et al.*, "Many-shot jailbreaking," *Advances in Neural Information Processing Systems*, vol. 37, pp. 129 696–129 742, 2024.

[568] Z.-X. Yong, C. Menghini, and S. H. Bach, "Low-resource languages jailbreak gpt-4," *arXiv preprint arXiv:2310.02446*, 2023.

[569] W. Wang, Z. Tu, C. Chen, Y. Yuan, J.-T. Huang, W. Jiao, and M. R. Lyu, "All languages matter: On the multilingual safety of llms," in *Annual Meeting of the Association for Computational Linguistics*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:271931322

[570] Z. Wei, Y. Wang, A. Li, Y. Mo, and Y. Wang, "Jailbreak and guard aligned language models with only few in-context demonstrations," *arXiv preprint arXiv:2310.06387*, 2023.

[571] N. Xu, F. Wang, B. Zhou, B. Z. Li, C. Xiao, and M. Chen, "Cognitive overload: Jailbreaking large language models with overloaded logical thinking," *arXiv preprint arXiv:2311.09827*, 2023.

[572] P. Ding, J. Kuang, D. Ma, X. Cao, Y. Xian, J. Chen, and S. Huang, "A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily," *arXiv preprint arXiv:2311.08268*, 2023.

[573] B. Upadhayay and V. Behzadan, "Sandwich attack: Multi-language mixture adaptive attack on llms," *arXiv preprint arXiv:2404.07242*, 2024.

[574] D. Yao, J. Zhang, I. G. Harris, and M. Carlsson, "Fuzzllm: A novel and universal fuzzing framework for proactively discovering jailbreak vulnerabilities in large language models," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 4485–4489.

[575] B. Li, H. Xing, C. Huang, J. Qian, H. Xiao, L. Feng, and C. Tian, "Structuralsleight: Automated jailbreak attacks on large language models utilizing uncommon text-encoded structure," *arXiv e-prints*, pp. arXiv–2406, 2024.

[576] A. Paulus, A. Zharmagambetov, C. Guo, B. Amos, and Y. Tian, "Advprompter: Fast adaptive adversarial prompting for llms," *arXiv preprint arXiv:2404.16873*, 2024.

[577] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does llm safety training fail?" *Advances in Neural Information Processing Systems*, vol. 36, pp. 80 079–80 110, 2023.

[578] Z. Chen, Z. Zhao, W. Qu, Z. Wen, Z. Han, Z. Zhu, J. Zhang, and H. Yao, "Pandora: Detailed llm jailbreaking via collaborated phishing agents with decomposed reasoning," in *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024.

[579] E. Perez, S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, and G. Irving, "Red teaming language models with language models," *arXiv preprint arXiv:2202.03286*, 2022.

[580] R. Shah, S. Pour, A. Tagade, S. Casper, J. Rando *et al.*, "Scalable and transferable black-box jailbreaks for language models via persona modulation," *arXiv preprint arXiv:2311.03348*, 2023.

[581] X. Guo, F. Yu, H. Zhang, L. Qin, and B. Hu, "Cold-attack: Jailbreaking llms with stealthiness and controllability," *arXiv preprint arXiv:2402.08679*, 2024.

[582] J. Yu, H. Luo, J. Y.-C. Hu, W. Guo, H. Liu, and X. Xing, "Enhancing jailbreak attack against large language models through silent tokens," *arXiv preprint arXiv:2405.20653*, 2024.

[583] Z.-W. Hong, I. Shenfeld, T.-H. Wang, Y.-S. Chuang, A. Pareja, J. Glass, A. Srivastava, and P. Agrawal, "Curiosity-driven red-teaming for large language models," *arXiv preprint arXiv:2402.19464*, 2024.

[584] X. Zheng, T. Pang, C. Du, Q. Liu, J. Jiang, and M. Lin, "Improved few-shot jailbreaking can circumvent aligned language models and their defenses," *Advances in Neural Information Processing Systems*, vol. 37, pp. 32 856–32 887, 2024.

[585] Z. Xiao, Y. Yang, G. Chen, and Y. Chen, "Distract large language models for automatic jailbreak attack," *arXiv preprint arXiv:2403.08424*, 2024.

[586] Z. Chang, M. Li, Y. Liu, J. Wang, Q. Wang, and Y. Liu, "Play guessing game with llm: Indirect jailbreak attack with implicit clues," *arXiv preprint arXiv:2402.09091*, 2024.

[587] J. Yu, X. Lin, Z. Yu, and X. Xing, "Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts," *arXiv preprint arXiv:2309.10253*, 2023.

[588] W. Jiang, Z. Wang, J. Zhai, S. Ma, Z. Zhao, and C. Shen, "Unlocking adversarial suffix optimization without affirmative phrases: Efficient black-box jailbreaking via llm as optimizer," *arXiv preprint arXiv:2408.11313*, 2024.

[589] J. Zhang, Z. Wang, R. Wang, X. Ma, and Y.-G. Jiang, "Enja: Ensemble jailbreak on large language models," *arXiv preprint arXiv:2408.03603*, 2024.

[590] X. Zhao, X. Yang, T. Pang, C. Du, L. Li, Y.-X. Wang, and W. Y. Wang, "Weak-to-strong jailbreaking on large language models," *arXiv preprint arXiv:2401.17256*, 2024.

[591] B. Upadhayay, V. Behzadan, and A. Karbasi, "Cognitive overload attack: Prompt injection for long context," *arXiv preprint arXiv:2410.11272*, 2024.

[592] H. Kwon and W. Pak, "Text-based prompt injection attack using mathematical functions in modern large language models," *Electronics*, vol. 13, no. 24, p. 5008, 2024.

[593] E. Bagdasaryan, T.-Y. Hsieh, B. Nassi, and V. Shmatikov, "Abusing images and sounds for indirect instruction injection in multi-modal llms," *arXiv preprint arXiv:2307.10490*, 2023.

[594] D. Pasquini, M. Strohmeier, and C. Troncoso, "Neural exec: Learning (and learning from) execution triggers for prompt injection attacks," in *Proceedings of the 2024 Workshop on Artificial Intelligence and Security*, 2024, pp. 89–100.

[595] Z. Shao, H. Liu, J. Mu, and N. Z. Gong, "Making llms vulnerable to prompt injection via poisoning alignment," *arXiv preprint arXiv:2410.14827*, 2024.

[596] Y. Yang, H. Yao, B. Yang, Y. He, Y. Li, T. Zhang,

Z. Qin, and K. Ren, "Tapi: Towards target-specific and adversarial prompt injection against code llms," *arXiv preprint arXiv:2407.09164*, 2024.

[597] Y. Ren, "F2a: An innovative approach for prompt injection by utilizing feign security detection agents," *arXiv preprint arXiv:2410.08776*, 2024.

[598] R. Pedro, D. Castro, P. Carreira, and N. Santos, "From prompt injections to sql injection attacks: How protected is your llm-integrated web application?" *arXiv preprint arXiv:2308.01990*, 2023.

[599] Y. Lee, T. Park, Y. Lee, J. Gong, and J. Kang, "Exploring potential prompt injection attacks in federated military llms and their mitigation," *arXiv preprint arXiv:2501.18416*, 2025.

[600] D. Lee and M. Tiwari, "Prompt infection: Llm-to-llm prompt injection within multi-agent systems," *arXiv preprint arXiv:2410.07283*, 2024.

[601] W. Zhang, X. Kong, C. Dewitt, T. Braunl, and J. B. Hong, "A study on prompt injection attack against llm-integrated mobile robotic systems," in *2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 2024, pp. 361–368.

[602] W. Meng, Z. Guo, L. Wu, C. Gong, W. Liu, W. Li, C. Wei, and W. Chen, "Rr: Unveiling llm training privacy through recollection and ranking," *arXiv preprint arXiv:2502.12658*, 2025.

[603] B. Jayaraman, E. Ghosh, H. Inan, M. Chase, S. Roy, and W. Dai, "Active data pattern extraction attacks on generative language models," *arXiv preprint arXiv:2207.10802*, 2022.

[604] Z. Zeng, T. Xiang, S. Guo, J. He, Q. Zhang, G. Xu, and T. Zhang, "Contrast-then-approximate: Analyzing keyword leakage of generative language models," *IEEE Transactions on Information Forensics and Security*, 2024.

[605] C. Jiang, X. Pan, G. Hong, C. Bao, and M. Yang, "Rag-thief: Scalable extraction of private data from retrieval-augmented generation applications with agent-based attacks," *arXiv preprint arXiv:2411.14110*, 2024.

[606] Z. Qi, H. Zhang, E. Xing, S. Kakade, and H. Lakkaraju, "Follow my instruction and spill the beans: Scalable data extraction from retrieval-augmented generation systems," *arXiv preprint arXiv:2402.17840*, 2024.

[607] S. Zeng, J. Zhang, P. He, Y. Xing, Y. Liu, H. Xu, J. Ren, S. Wang, D. Yin, Y. Chang *et al.*, "The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag)," *arXiv preprint arXiv:2402.16893*, 2024.

[608] Y. Peng, J. Wang, H. Yu, and A. Houmansadr, "Data extraction attacks in retrieval-augmented generation via backdoors," *arXiv preprint arXiv:2411.01705*, 2024.

[609] A. Panda, C. A. Choquette-Choo, Z. Zhang, Y. Yang, and P. Mittal, "Teach llms to phish: Stealing private information from language models," *arXiv preprint arXiv:2403.00871*, 2024.

[610] L. Lu, Z. Zuo, Z. Sheng, and P. Zhou, "Merger-as-a-stealer: Stealing targeted pii from aligned llms with model merging," *arXiv preprint arXiv:2502.16094*, 2025.

[611] X. Chen, S. Tang, R. Zhu, S. Yan, L. Jin, Z. Wang, L. Su, Z. Zhang, X. Wang, and H. Tang, "The janus interface: How fine-tuning in large language models amplifies the privacy risks," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 1285–1299.

[612] R. Panchendrarajan and S. Bhoi, "Dataset reconstruction attack against language models," 2021.

[613] M. R. U. Rashid, V. A. Dasu, K. Gu, N. Sultana, and S. Mehnaz, "Fltrojan: Privacy leakage attacks against federated language models through selective weight tampering," *arXiv preprint arXiv:2310.16152*, 2023.

[614] J. Dentan, A. Paran, and A. Shabou, "Reconstructing training data from document understanding models," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 6813–6830.

[615] J. Hościłowicz, P. Popiołek, J. Rudkowski, J. Bieniasz, and A. Janicki, "Unconditional token forcing: Extracting text hidden within llm," in *2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2024, pp. 621–624.

[616] A. Al-Kaswan, M. Izadi, and A. Van Deursen, "Traces of memorisation in large language models for code," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–12.

[617] Y. Nie, C. Wang, K. Wang, G. Xu, G. Xu, and H. Wang, "Decoding secret memorization in code llms through token-level characterization," *arXiv preprint arXiv:2410.08858*, 2024.

[618] E. Lehman, S. Jain, K. Pichotta, Y. Goldberg, and B. C. Wallace, "Does bert pretrained on clinical notes reveal sensitive data?" *arXiv preprint arXiv:2104.07762*, 2021.

[619] A. Diera, N. Lell, A. Garifullina, and A. Scherp, "Memorization of named entities in fine-tuned bert models," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2023, pp. 258–279.

[620] R. Zhang, S. Hidano, and F. Koushanfar, "Text revealer: Private text reconstruction via model inversion attacks against transformers," *arXiv preprint arXiv:2209.10505*, 2022.

[621] Y. Huang, Y. Li, W. Wu, J. Zhang, and M. R. Lyu, "Your code secret belongs to me: neural code completion tools can memorize hard-coded credentials," *Proceedings of the ACM on Software Engineering*, vol. 1, no. FSE, pp. 2515–2537, 2024.

[622] T. Tiwari and G. E. Suh, "Sequence-level analysis of leakage risk of training data in large language models," *arXiv preprint arXiv:2412.11302*, 2024.

[623] H. Shao, J. Huang, S. Zheng, and K. C.-C. Chang, "Quantifying association capabilities of large language models and its implications on privacy leakage," *arXiv preprint arXiv:2305.12707*, 2023.

[624] Y. More, P. Ganesh, and G. Farnadi, "Towards more realistic extraction attacks: An adversarial perspective," *arXiv preprint arXiv:2407.02596*, 2024.

[625] R. Staab, M. Vero, M. Balunović, and M. Vechev, "Beyond memorization: Violating privacy via inference with large language models," *arXiv preprint arXiv:2310.07298*, 2023.

[626] H. Xu, Z. Zhang, X. Yu, Y. Wu, Z. Zha, B. Xu, W. Xu, M. Hu, and K. Peng, "Targeted training data extrac-

tion—neighborhood comparison-based membership inference attacks in large language models," *Applied Sciences*, vol. 14, no. 16, p. 7118, 2024.

[627] A. Karamolegkou, J. Li, L. Zhou, and A. Søgaard, "Copyright violations and large language models," *arXiv preprint arXiv:2310.13771*, 2023.

[628] X. Zheng, H. Han, S. Shi, Q. Fang, Z. Du, X. Hu, and Q. Guo, "Inputsnatch: Stealing input in llm services via timing side-channel attacks," *arXiv preprint arXiv:2411.18191*, 2024.

[629] Y. Dong, R. Mu, G. Jin, Y. Qi, J. Hu, X. Zhao, J. Meng, W. Ruan, and X. Huang, "Building guardrails for large language models," *arXiv preprint arXiv:2402.01822*, 2024.

[630] N. Jain, A. Schwarzschild, Y. Wen, G. Somepalli, J. Kirchenbauer, P. yeh Chiang, M. Goldblum, A. Saha, J. Geiping, and T. Goldstein, "Baseline defenses for adversarial attacks against aligned language models," 2024.

[631] H. Lin, Y. Lao, T. Geng, T. Yu, and W. Zhao, "Uniguardian: A unified defense for detecting prompt injection, backdoor attacks and adversarial attacks in large language models," *arXiv preprint arXiv:2502.13141*, 2025.

[632] Z. Hu, G. Wu, S. Mitra, R. Zhang, T. Sun, H. Huang, and V. Swaminathan, "Token-level adversarial prompt detection based on perplexity measures and contextual information," in *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025.

[633] Y. Gou, K. Chen, Z. Liu, L. Hong, H. Xu, Z. Li, D.-Y. Yeung, J. T. Kwok, and Y. Zhang, "Eyes closed, safety on: Protecting multimodal llms via image-to-text transformation," in *European Conference on Computer Vision*, 2024, pp. 388–404.

[634] S. Armstrong, M. Franklin, C. Stevens, and R. Gorman, "Defense against the dark prompts: Mitigating best-of-n jailbreaking with prompt evaluation," *arXiv preprint arXiv:2107.03374*, 2025.

[635] Y. Xie, M. Fang, R. Pi, and N. Gong, "GradSafe: Detecting jailbreak prompts for LLMs via safety-critical gradient analysis," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., 2024, pp. 507–518.

[636] B. Peng, Z. Bi, Q. Niu, M. Liu, P. Feng, T. Wang, L. K. Yan, Y. Wen, Y. Zhang, and C. H. Yin, "Jailbreaking and mitigation of vulnerabilities in large language models," *arXiv preprint arXiv:2410.15236*, 2024.

[637] A. Kumar, C. Agarwal, S. Srinivas, A. J. Li, S. Feizi, and H. Lakkaraju, "Certifying LLM safety against adversarial prompting," in *First Conference on Language Modeling*, 2024.

[638] X. Zhang, C. Zhang, T. Li, Y. Huang, X. Jia, M. Hu, J. Zhang, Y. Liu, S. Ma, and C. Shen, "Jailguard: A universal detection framework for llm prompt-based attacks," *arXiv preprint arXiv:2312.10766*, 2023.

[639] Y. Liu, Y. Jia, R. Geng, J. Jia, and N. Z. Gong, "Formalizing and benchmarking prompt injection attacks and defenses," in *Proceedings of the 33rd USENIX Conference on Security Symposium*, 2024.

[640] X. Suo, "Signed-prompt: A new approach to prevent prompt injection attacks against llm-integrated applications," in *AIP Conference Proceedings*, vol. 3194, no. 1. AIP Publishing, 2024.

[641] L. Yan, Z. Zhang, G. Tao, K. Zhang, X. Chen, G. Shen, and X. Zhang, "Parafuzz: An interpretability-driven technique for detecting poisoned samples in nlp," *Advances in Neural Information Processing Systems*, vol. 36, pp. 66 755–66 767, 2023.

[642] X. Hu, P.-Y. Chen, and T.-Y. Ho, "Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 126 265–126 296.

[643] G. Alon and M. J. Kamfonas, "Detecting language model attacks with perplexity," 2024.

[644] J. Ji, B. Hou, A. Robey, G. J. Pappas, H. Hassani, Y. Zhang, E. Wong, and S. Chang, "Defending large language models against jailbreak attacks via semantic smoothing," *CoRR*, 2024.

[645] M. Phute, A. Helbling, M. Hull, S. Peng, S. Szyller, C. Cornelius, and D. H. Chau, "Llm self defense: By self examination, llms know they are being tricked," *arXiv preprint arXiv:2308.07308*, 2024.

[646] L. N. Candogan, Y. Wu, E. A. Rocamora, G. G. Chrysos, and V. Cevher, "Single-pass detection of jailbreaking input in large language models," *arXiv preprint arXiv:2502.15435*, 2025.

[647] B. Cao, Y. Cao, L. Lin, and J. Chen, "Defending against alignment-breaking attacks via robustly aligned LLM," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., 2024, pp. 10 542–10 560.

[648] Y. Zhang, L. Ding, L. Zhang, and D. Tao, "Intention analysis makes LLMs a good jailbreak defender," in *Proceedings of the 31st International Conference on Computational Linguistics*, 2025, pp. 2947–2968.

[649] S. Han, K. Rao, A. Ettinger, L. Jiang, B. Y. Lin, N. Lambert, Y. Choi, and N. Dziri, "Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms," in *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[650] M. Pisano, P. Ly, A. Sanders, B. Yao, D. Wang, T. Strzalkowski, and M. Si, "Bergeron: Combating adversarial attacks through a conscience-based alignment framework," *arXiv preprint arXiv:2312.00029*, 2024.

[651] A. Robey, E. Wong, H. Hassani, and G. J. Pappas, "Smoothllm: Defending large language models against jailbreaking attacks," *arXiv preprint arXiv:2310.03684*, 2023.

[652] J. Ji, B. Hou, Z. Zhang, G. Zhang, W. Fan, Q. Li, Y. Zhang, G. Liu, S. Liu, and S. Chang, "Advancing the robustness of large language models through self-denoised smoothing," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, 2024, pp. 246–257.

[653] J. Yi, Y. Xie, B. Zhu, K. Hines, E. Kiciman, G. Sun, X. Xie, and F. Wu, "Benchmarking and defending against indirect prompt injection attacks on large lan-

guage models," *CoRR*, 2023.

[654] X. Song, S. Duan, and G. Liu, "Alis: Aligned llm instruction security strategy for unsafe input prompt," in *Proceedings of the 31st International Conference on Computational Linguistics*, 2025, pp. 9124–9146.

[655] Y. Wang, Z. Shi, A. Bai, and C.-J. Hsieh, "Defending llms against jailbreaking attacks via backtranslation," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., 2024, pp. 16 031–16 046.

[656] E. Zverev, S. Abdelnabi, M. Fritz, and C. H. Lampert, "Can LLMs separate instructions from data? and what do we even mean by that?" *CoRR*, 2024.

[657] Y. Dong, R. Mu, G. Jin, Y. Qi, J. Hu, X. Zhao, J. Meng, W. Ruan, and X. Huang, "Building guardrails for large language models," *arXiv preprint arXiv:2402.01822*, 2024.

[658] D. Kumar, Y. A. AbuHashem, and Z. Durumeric, "Watch your language: Investigating content moderation with large language models," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 18, 2024, pp. 865–878.

[659] T. Rebedea, R. Dinu, M. N. Sreedhar, C. Parisien, and J. Cohen, "Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2023, pp. 431–445.

[660] OpenAI, "Improving model safety behavior with rule-based rewards," https://openai.com/index/improving-model-safety-behavior-with-rule-based-rewards/, 2025, accessed: 2025-03-24.

[661] H. Ma, C. Zhang, H. Fu, P. Zhao, and B. Wu, "Adapting large language models for content moderation: Pitfalls in data engineering and supervised fine-tuning," *arXiv preprint arXiv:2310.03400*, 2023.

[662] M. Phute, A. Helbling, M. Hull, S. Peng, S. Szyller, C. Cornelius, and D. H. Chau, "Llm self defense: By self examination, llms know they are being tricked," *arXiv preprint arXiv:2308.07308*, 2023.

[663] Z. Gou, Z. Shao, Y. Gong, Y. Shen, Y. Yang, N. Duan, and W. Chen, "Critic: Large language models can self-correct with tool-interactive critiquing," *arXiv preprint arXiv:2305.11738*, 2023.

[664] C. Lu, S. Holt, C. Fanconi, A. J. Chan, J. Foerster, M. van der Schaar, and R. T. Lange, "Discovering preference optimization algorithms with and for large language models," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 86 528–86 573.

[665] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang *et al.*, "Self-refine: Iterative refinement with self-feedback," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 534–46 594, 2023.

[666] D. Jiang, X. Ren, and B. Y. Lin, "Llm-blender: Ensembling large language models with pairwise ranking and generative fusion," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 14 165–14 178.

[667] Z. Lai, X. Zhang, and S. Chen, "Adaptive ensembles of fine-tuned transformers for llm-generated text detection," in *2024 International Joint Conference on Neural Networks*. IEEE, 2024, pp. 1–7.

[668] C. Xiong, X. Qi, P.-Y. Chen, and T.-Y. Ho, "Defensive prompt patch: A robust and interpretable defense of llms against jailbreak attacks," *arXiv preprint arXiv:2405.20099*, 2024.

[669] Z. Zhang, Q. Zhang, and J. Foerster, "Parden, can you repeat that? defending against jailbreaks via repetition," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 60 271–60 287.

[670] Z. Yuan, Z. Xiong, Y. Zeng, N. Yu, R. Jia, D. Song, and B. Li, "Rigorllm: resilient guardrails for large language models against undesired content," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 57 953–57 965.

[671] M. Cao, M. Fatemi, J. C. Cheung, and S. Shabanian, "Systematic rectification of language models via dead-end analysis," in *The Eleventh International Conference on Learning Representations*, 2023.

[672] F. Faal, K. Schmitt, and J. Y. Yu, "Reward modeling for mitigating toxicity in transformer-based language models," *Applied Intelligence*, vol. 53, no. 7, p. 8421–8435, 2022.

[673] W. Zeng, Y. Liu, R. Mullins, L. Peran, J. Fernandez, H. Harkous, K. Narasimhan, D. Proud, P. Kumar, B. Radharapu *et al.*, "Shieldgemma: Generative ai content moderation based on gemma," *arXiv preprint arXiv:2407.21772*, 2024.

[674] Z. Wang, F. Yang, L. Wang, P. Zhao, H. Wang, L. Chen, Q. Lin, and K.-F. Wong, "SELF-GUARD: Empower the LLM to safeguard itself," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, 2024, pp. 1648–1668.

[675] S. Ghosh, P. Varshney, E. Galinkin, and C. Parisien, "Aegis: Online adaptive ai content safety moderation with ensemble of llm experts," *arXiv preprint arXiv:2404.05993*, 2024.

[676] W. Wang, J.-T. Huang, W. Wu, J. Zhang, Y. Huang, S. Li, P. He, and M. R. Lyu, "Mttm: Metamorphic testing for textual content moderation software," *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pp. 2387–2399, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:256826966

[677] K.-L. Chiu, A. Collins, and R. Alexander, "Detecting hate speech with gpt-3," *arXiv preprint arXiv:2103.12407*, 2021.

[678] J. Kim, A. Derakhshan, and I. G. Harris, "Robust safety classifier for large language models: Adversarial prompt shield," *arXiv preprint arXiv:2311.00172*, 2023.

[679] B. Krause, A. D. Gotmare, B. McCann, N. S. Keskar, S. Joty, R. Socher, and N. F. Rajani, "Gedi: Generative discriminator guided sequence generation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 4929–4952.

[680] Q. Liu, Z. Zhou, L. He, Y. Liu, W. Zhang, and S. Su, "Alignment-enhanced decoding: Defending jailbreaks via token-level adaptive refining of probability distributions," in *Proceedings of the 2024 Conference on*

Empirical Methods in Natural Language Processing, 2024, pp. 2802–2816.

[681] A. Liu, M. Sap, X. Lu, S. Swayamdipta, C. Bhagavatula, N. A. Smith, and Y. Choi, "Dexperts: Decoding-time controlled text generation with experts and anti-experts," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021, pp. 6691–6706.

[682] T. Radcliffe, E. Lockhart, and J. Wetherington, "Automated prompt engineering for semantic vulnerabilities in large language models," *Authorea Preprints*, 2024.

[683] F. Trad and A. Chehab, "Prompt engineering or fine-tuning? a case study on phishing detection with large language models," *Machine Learning and Knowledge Extraction*, vol. 6, no. 1, pp. 367–384, 2024.

[684] A. Zhou, B. Li, and H. Wang, "Robust prompt optimization for defending language models against jailbreaking attacks," in *Advances in Neural Information Processing Systems*, vol. 37. Curran Associates, Inc., 2024, pp. 40 184–40 211.

[685] Y. Mo, Y. Wang, Z. Wei, and Y. Wang, "Fight back against jailbreaking via prompt adversarial tuning," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[686] Y. Zhang, L. Ding, L. Zhang, and D. Tao, "Intention analysis makes llms a good jailbreak defender," in *Proceedings of the 31st International Conference on Computational Linguistics*, 2025, pp. 2947–2968.

[687] Y. Chen, H. Li, Z. Zheng, Y. Song, D. Wu, and B. Hooi, "Defense against prompt injection attack by leveraging attack techniques," *arXiv preprint arXiv:2411.00459*, 2024.

[688] Z. Zhang, J. Yang, P. Ke, F. Mi, H. Wang, and M. Huang, "Defending large language models against jailbreaking attacks through goal prioritization," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2023, pp. 8865–8887.

[689] Y. Xie, J. Yi, J. Shao, J. Curl, L. Lyu, Q. Chen, X. Xie, and F. Wu, "Defending chatgpt against jailbreak attack via self-reminders," *Nature Machine Intelligence*, vol. 5, no. 12, pp. 1486–1496, 2023.

[690] S. Chen, J. Piet, C. Sitawarin, and D. Wagner, "Struq: Defending against prompt injection with structured queries," *arXiv preprint arXiv:2402.06363*, 2024.

[691] K. Hines, G. Lopez, M. Hall, F. Zarfati, Y. Zunger, and E. Kiciman, "Defending against indirect prompt injection attacks with spotlighting," *arXiv preprint arXiv:2403.14720*, 2024.

[692] S. Slocum and D. Hadfield-Menell, "Inverse prompt engineering for task-specific LLM safety," 2025. [Online]. Available: https://openreview.net/forum?id=3MDmM0rMPQ

[693] K. Edemacu and X. Wu, "Privacy preserving prompt engineering: A survey," *arXiv preprint arXiv:2404.06001*, 2024.

[694] S. Utpala, S. Hooker, and P.-Y. Chen, "Locally differentially private document generation using zero shot prompting," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 8442–8457.

[695] H. Duan, A. Dziedzic, N. Papernot, and F. Boenisch, "Flocks of stochastic parrots: Differentially private prompt learning for large language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 76 852–76 871, 2023.

[696] W. Wang, W. Jiao, J. Huang, R. Dai, J.-T. Huang, Z. Tu, and M. R. Lyu, "Not all countries celebrate thanksgiving: On the cultural dominance in large language models," *ArXiv*, vol. abs/2310.12481, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:264305810

[697] M. Kaneko, D. Bollegala, N. Okazaki, and T. Baldwin, "Evaluating gender bias in large language models via chain-of-thought prompting," *arXiv preprint arXiv:2401.15585*, 2024.

[698] X. He, S. Zannettou, Y. Shen, and Y. Zhang, "You only prompt once: On the capabilities of prompt learning on large language models to tackle toxic content," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, pp. 770–787.

[699] X. Zou, Y. Chen, and K. Li, "Is the system message really important to jailbreaks in large language models?" *arXiv preprint arXiv:2402.14857*, 2024.

[700] R. Xu, Z. Qi, and W. Xu, "Preemptive answer "attacks" on chain-of-thought reasoning," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 14 708–14 726.

[701] C. Zheng, F. Yin, H. Zhou, F. Meng, J. Zhou, K.-W. Chang, M. Huang, and N. Peng, "On prompt-driven safeguarding for large language models," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235, 21–27 Jul 2024, pp. 61 593–61 613.

[702] Y. Wang, X. Liu, Y. Li, M. Chen, and C. Xiao, "Adashield: Safeguarding multimodal large language models from structure-based attack via adaptive shield prompting," in *European Conference on Computer Vision*. Springer, 2024, pp. 77–94.

[703] Z. Shi, Z. Wang, Y. Su, W. Luo, H. Gao, F. Yang, R. Tang, and Y. Zhang, "Robustness-aware automatic prompt optimization," *arXiv preprint arXiv:2412.18196*, 2024.

[704] Y. Wu, Y. Gao, B. Zhu, Z. Zhou, X. Sun, S. Yang, J.-G. Lou, Z. Ding, and L. Yang, "Strago: Harnessing strategic guidance for prompt optimization," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 10 043–10 061.

[705] F. Wu, N. Zhang, S. Jha, P. McDaniel, and C. Xiao, "A new era in llm security: Exploring security concerns in real-world llm-based systems," *arXiv preprint arXiv:2402.18649*, 2024.

[706] A. Borzunov, M. Ryabinin, A. Chumachenko, D. Baranchuk, T. Dettmers, Y. Belkada, P. Samygin, and C. A. Raffel, "Distributed inference and fine-tuning of large language models over the internet," *Advances in neural information processing systems*, vol. 36, pp. 12 312–12 331, 2023.

[707] A. Agrawal, N. Kedia, A. Panwar, J. Mohan, N. Kwatra, B. Gulavani, A. Tumanov, and R. Ramjee, "Taming {Throughput-Latency} tradeoff in {LLM} inference with {Sarathi-Serve}," in *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI*

24), 2024, pp. 117–134.

[708] Y. Zhong, S. Liu, J. Chen, J. Hu, Y. Zhu, X. Liu, X. Jin, and H. Zhang, "{DistServe}: Disaggregating prefill and decoding for goodput-optimized large language model serving," in *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, 2024, pp. 193–210.

[709] H. Sun, Z. Chen, X. Yang, Y. Tian, and B. Chen, "Triforce: Lossless acceleration of long sequence generation with hierarchical speculative decoding," in *First Conference on Language Modeling*, 2024.

[710] T. Cai, Y. Li, Z. Geng, H. Peng, J. D. Lee, D. Chen, and T. Dao, "Medusa: Simple LLM inference acceleration framework with multiple decoding heads," in *Proceedings of the 41st International Conference on Machine Learning*, vol. 235. PMLR, 2024, pp. 5209–5235.

[711] J. Chen, V. Tiwari, R. Sadhukhan, Z. Chen, J. Shi, I. E.-H. Yen, and B. Chen, "Magicdec: Breaking the latency-throughput tradeoff for long context generation with speculative decoding," *arXiv preprint arXiv:2408.11049*, 2024.

[712] C. Holmes, M. Tanaka, M. Wyatt, A. A. Awan, J. Rasley, S. Rajbhandari, R. Y. Aminabadi, H. Qin, A. Bakhtiari, L. Kurilenko *et al.*, "Deepspeed-fastgen: High-throughput text generation for llms via mii and deepspeed-inference," *arXiv preprint arXiv:2401.08671*, 2024.

[713] R. Svirschevski, A. May, Z. Chen, B. Chen, Z. Jia, and M. Ryabinin, "Specexec: Massively parallel speculative decoding for interactive llm inference on consumer devices," *Advances in Neural Information Processing Systems*, vol. 37, pp. 16 342–16 368, 2024.

[714] P. Wang, D. Zhang, L. Li, C. Tan, X. Wang, M. Zhang, K. Ren, B. Jiang, and X. Qiu, "Inferaligner: Inference-time alignment for harmlessness through cross-model guidance," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 10 460–10 479.

[715] X. Wang, D. Wu, Z. Ji, Z. Li, P. Ma, S. Wang, Y. Li, Y. Liu, N. Liu, and J. Rahmel, "Selfdefend: Llms can defend themselves against jailbreaking in a practical manner," *CoRR*, 2024.

[716] X. Hu, P.-Y. Chen, and T.-Y. Ho, "Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes," *arXiv preprint arXiv:2403.00867*, 2024.

[717] R. K. Sharma, V. Gupta, and D. Grossman, "Spml: A dsl for defending language models against prompt attacks," *arXiv preprint arXiv:2402.11755*, 2024.

[718] J. Zhao, S. Wang, Y. Zhao, X. Hou, K. Wang, P. Gao, Y. Zhang, C. Wei, and H. Wang, "Models are codes: Towards measuring malicious code poisoning attacks on pre-trained model hubs," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, 2024, pp. 2087–2098.

[719] S. Ghosh, P. Varshney, E. Galinkin, and C. Parisien, "Aegis: Online adaptive ai content safety moderation with ensemble of llm experts," *arXiv preprint arXiv:2404.05993*, 2024.

[720] S. Ghosh, P. Varshney, M. N. Sreedhar, A. Padmakumar, T. Rebedea, J. R. Varghese, and C. Parisien, "Aegis2. 0: A diverse ai safety dataset and risks taxonomy for alignment of llm guardrails," in *Neurips Safe Generative AI Workshop 2024*, 2024.

[721] S. Han, K. Rao, A. Ettinger, L. Jiang, B. Y. Lin, N. Lambert, Y. Choi, and N. Dziri, "Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms," *arXiv preprint arXiv:2406.18495*, 2024.

[722] W. Zeng, Y. Liu, R. Mullins, L. Peran, J. Fernandez, H. Harkous, K. Narasimhan, D. Proud, P. Kumar, B. Radharapu *et al.*, "Shieldgemma: Generative ai content moderation based on gemma," *arXiv preprint arXiv:2407.21772*, 2024.

[723] Y. Liu, H. Gao, S. Zhai, J. Xia, T. Wu, Z. Xue, Y. Chen, K. Kawaguchi, J. Zhang, and B. Hooi, "Guardreasoner: Towards reasoning-based llm safeguards," *arXiv preprint arXiv:2501.18492*, 2025.

[724] C. Wang, Y. Liu, B. Li, D. Zhang, Z. Li, and J. Fang, "Safety in large reasoning models: A survey," *arXiv preprint arXiv:2504.17704*, 2025.

[725] H. Jin, A. Zhou, J. Menke, and H. Wang, "Jailbreaking large language models against moderation guardrails via cipher characters," *Advances in Neural Information Processing Systems*, vol. 37, pp. 59 408–59 435, 2024.

[726] D. Ran, J. Liu, Y. Gong, J. Zheng, X. He, T. Cong, and A. Wang, "Jailbreakeval: An integrated toolkit for evaluating jailbreak attempts against large language models," *arXiv preprint arXiv:2406.09321*, 2024.

[727] H. Qiu, S. Zhang, A. Li, H. He, and Z. Lan, "Latent jailbreak: A benchmark for evaluating text safety and output robustness of large language models," *arXiv preprint arXiv:2307.08487*, 2023.

[728] K. Zhu, J. Wang, J. Zhou, Z. Wang, H. Chen, Y. Wang, L. Yang, W. Ye, Y. Zhang, N. Gong *et al.*, "Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts," in *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, 2023, pp. 57–68.

[729] A. Pei, Z. Yang, S. Zhu, R. Cheng, and J. Jia, "Selfprompt: Autonomously evaluating llm robustness via domain-constrained knowledge guidelines and refined adversarial prompts," *arXiv preprint arXiv:2412.00765*, 2024.

[730] Z. Xu, Y. Liu, G. Deng, Y. Li, and S. Picek, "A comprehensive study of jailbreak attack versus defense for large language models," *arXiv preprint arXiv:2402.13457*, 2024.

[731] K. Chen, Y. Liu, D. Wang, J. Chen, and W. Wang, "Characterizing and evaluating the reliability of llms against jailbreak attacks," *arXiv preprint arXiv:2408.09326*, 2024.

[732] B. Wang, C. Xu, S. Wang, Z. Gan, Y. Cheng, J. Gao, A. H. Awadallah, and B. Li, "Adversarial glue: A multi-task benchmark for robustness evaluation of language models," *arXiv preprint arXiv:2111.02840*, 2021.

[733] G. Dong, J. Zhao, T. Hui, D. Guo, W. Wang, B. Feng, Y. Qiu, Z. Gongque, K. He, Z. Wang *et al.*, "Revisit input perturbation problems for llms: A unified robustness evaluation framework for noisy slot filling task," in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer,

2023, pp. 682–694.

[734] J. Zheng, A. Ritter, and W. Xu, "Neo-bench: Evaluating robustness of large language models with neologisms," *arXiv preprint arXiv:2402.12261*, 2024.

[735] Y. Li, Y. Guo, F. Guerin, and C. Lin, "Evaluating large language models for generalization and robustness via data compression," *arXiv preprint arXiv:2402.00861*, 2024.

[736] Q. Zhang, H. Qiu, D. Wang, Y. Li, T. Zhang, W. Zhu, H. Weng, L. Yan, and C. Zhang, "A benchmark for semantic sensitive information in llms outputs," in *The Thirteenth International Conference on Learning Representations*.

[737] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[738] J. Li, X. Cheng, W. X. Zhao, J.-Y. Nie, and J.-R. Wen, "Halueval: A large-scale hallucination evaluation benchmark for large language models," *arXiv preprint arXiv:2305.11747*, 2023.

[739] A. Pal, L. K. Umapathi, and M. Sankarasubbu, "Med-halt: Medical domain hallucination test for large language models," *arXiv preprint arXiv:2307.15343*, 2023.

[740] Z. Ji, Y. Gu, W. Zhang, C. Lyu, D. Lin, and K. Chen, "Anah: Analytical annotation of hallucinations in large language models," *arXiv preprint arXiv:2405.20315*, 2024.

[741] P. Manakul, A. Liusie, and M. J. Gales, "Selfcheck-gpt: Zero-resource black-box hallucination detection for generative large language models," *arXiv preprint arXiv:2303.08896*, 2023.

[742] Y.-S. Chuang, Y. Xie, H. Luo, Y. Kim, J. Glass, and P. He, "Dola: Decoding by contrasting layers improves factuality in large language models," *arXiv preprint arXiv:2309.03883*, 2023.

[743] N. Mündler, J. He, S. Jenko, and M. Vechev, "Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation," *arXiv preprint arXiv:2305.15852*, 2023.

[744] M. Elaraby, M. Lu, J. Dunn, X. Zhang, Y. Wang, S. Liu, P. Tian, Y. Wang, and Y. Wang, "Halo: Estimation and reduction of hallucinations in open-source weak large language models," *arXiv preprint arXiv:2308.11764*, 2023.

[745] Z. Ji, D. Chen, E. Ishii, S. Cahyawijaya, Y. Bang, B. Wilie, and P. Fung, "Llm internal states reveal hallucination risk faced with a query," *arXiv preprint arXiv:2407.03282*, 2024.

[746] J. Wei, Y. Yao, J.-F. Ton, H. Guo, A. Estornell, and Y. Liu, "Measuring and reducing llm hallucination without gold-standard answers," *arXiv preprint arXiv:2402.10412*, 2024.

[747] A. Deshpande, V. Murahari, T. Rajpurohit, A. Kalyan, and K. Narasimhan, "Toxicity in chatgpt: Analyzing persona-assigned language models," *arXiv preprint arXiv:2304.05335*, 2023.

[748] A. de Wynter, I. Watts, T. Wongsangaroonsri, M. Zhang, N. Farra, N. E. Altıntoprak, L. Baur, S. Claudet, P. Gajdusek, C. Gören *et al.*, "Rtp-lx: Can llms evaluate toxicity in multilingual scenarios?" *arXiv preprint arXiv:2404.14397*, 2024.

[749] D. Esiobu, X. Tan, S. Hosseini, M. Ung, Y. Zhang, J. Fernandes, J. Dwivedi-Yu, E. Presani, A. Williams, and E. M. Smith, "Robbie: Robust bias evaluation of large generative language models," *arXiv preprint arXiv:2311.18140*, 2023.

[750] S. Wang, P. Wang, T. Zhou, Y. Dong, Z. Tan, and J. Li, "Ceb: Compositional evaluation benchmark for fairness in large language models," *arXiv preprint arXiv:2407.02408*, 2024.

[751] H. Li, D. Guo, D. Li, W. Fan, Q. Hu, X. Liu, C. Chan, D. Yao, Y. Yao, and Y. Song, "Privlm-bench: A multi-level privacy evaluation benchmark for language models," *arXiv preprint arXiv:2311.04044*, 2023.

[752] Q. Li, J. Hong, C. Xie, J. Tan, R. Xin, J. Hou, X. Yin, Z. Wang, D. Hendrycks, Z. Wang *et al.*, "Llm-pbe: Assessing data privacy in large language models," *arXiv preprint arXiv:2408.12787*, 2024.

[753] D. Zhu, D. Chen, X. Wu, J. Geng, Z. Li, J. Grossklags, and L. Ma, "Privauditor: Benchmarking data protection vulnerabilities in llm adaptation techniques," *Advances in Neural Information Processing Systems*, vol. 37, pp. 9668–9689, 2024.

[754] L. Rossi, B. Marek, V. Hanke, X. Wang, M. Backes, A. Dziedzic, and F. Boenisch, "Auditing empirical privacy protection of private llm adaptations," in *Neurips Safe Generative AI Workshop 2024*.

[755] T. Singh, H. Aditya, V. K. Madisetti, and A. Bahga, "Whispered tuning: Data privacy preservation in fine-tuning llms through differential privacy," *Journal of Software Engineering and Applications*, vol. 17, no. 1, pp. 1–22, 2024.

[756] H. Li, W. Hu, H. Jing, Y. Chen, Q. Hu, S. Han, T. Chu, P. Hu, and Y. Song, "Privaci-bench: Evaluating privacy with contextual integrity and legal compliance," *arXiv preprint arXiv:2502.17041*, 2025.

[757] O. Cartwright, H. Dunbar, and T. Radcliffe, "Evaluating privacy compliance in commercial large language models-chatgpt, claude, and gemini," 2024.

[758] X. Zhou, M. Weyssow, R. Widyasari, T. Zhang, J. He, Y. Lyu, J. Chang, B. Zhang, D. Huang, and D. Lo, "Lessleak-bench: A first investigation of data leakage in llms across 83 software engineering benchmarks," *arXiv preprint arXiv:2502.06215*, 2025.

[759] Y. Song, R. Liu, S. Chen, Q. Ren, Y. Zhang, and Y. Yu, "Securesql: Evaluating data leakage of large language models as natural language interfaces to databases," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 5975–5990.

[760] X. Liu, Y. Zhu, J. Gu, Y. Lan, C. Yang, and Y. Qiao, "Mm-safetybench: A benchmark for safety evaluation of multimodal large language models," in *European Conference on Computer Vision*. Springer, 2024, pp. 386–403.

[761] W. Luo, S. Ma, X. Liu, X. Guo, and C. Xiao, "Jailbreakv-28k: A benchmark for assessing the robustness of multimodal large language models against jailbreak attacks," *arXiv e-prints*, pp. arXiv–2404, 2024.

[762] F. Weng, Y. Xu, C. Fu, and W. Wang, "A comprehensive study on jailbreak attacks and defenses for

multimodal large language models," *arXiv preprint arXiv:2408.08464*, 2024.

[763] Z. Li, P.-Y. Chen, and T.-Y. Ho, "Retention score: Quantifying jailbreak risks for vision language models," *arXiv preprint arXiv:2412.17544*, 2024.

[764] T. Guan, F. Liu, X. Wu, R. Xian, Z. Li, X. Liu, X. Wang, L. Chen, F. Huang, Y. Yacoob *et al.*, "Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 375–14 385.

[765] Y. Li, Y. Du, K. Zhou, J. Wang, W. X. Zhao, and J.-R. Wen, "Evaluating object hallucination in large vision-language models," *arXiv preprint arXiv:2305.10355*, 2023.

[766] C. Cui, Y. Zhou, X. Yang, S. Wu, L. Zhang, J. Zou, and H. Yao, "Holistic analysis of hallucination in gpt-4v (ision): Bias and interference challenges," *arXiv preprint arXiv:2311.03287*, 2023.

[767] S. Wang, X. Ye, Q. Cheng, J. Duan, S. Li, J. Fu, X. Qiu, and X. Huang, "Cross-modality safety alignment," *arXiv preprint arXiv:2406.15279*, 2024.

[768] A. Agarwal, S. Panda, A. Charles, B. Kumar, H. Patel, P. Pattnayak, T. H. Rafi, T. Kumar, and D.-K. Chae, "Mvtamperbench: Evaluating robustness of vision-language models," *arXiv preprint arXiv:2412.19794*, 2024.

[769] H. Zhang, W. Shao, H. Liu, Y. Ma, P. Luo, Y. Qiao, and K. Zhang, "Avibench: Towards evaluating the robustness of large vision-language model on adversarial visual-instructions," *arXiv e-prints*, pp. arXiv–2403, 2024.

[770] Z. Hu, Y. Ren, J. Li, and Y. Yin, "Viva: A benchmark for vision-grounded decision-making with human values," *arXiv preprint arXiv:2407.03000*, 2024.

[771] Y. Xiao, A. Liu, Q. Cheng, Z. Yin, S. Liang, J. Li, J. Shao, X. Liu, and D. Tao, "Genderbias-\emph {VL}: Benchmarking gender bias in vision language models via counterfactual probing," *arXiv preprint arXiv:2407.00600*, 2024.

[772] L. Gustafson, C. Rolland, N. Ravi, Q. Duval, A. Adcock, C.-Y. Fu, M. Hall, and C. Ross, "Facet: Fairness in computer vision evaluation benchmark," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 370–20 382.

[773] E. Slyman, S. Lee, S. Cohen, and K. Kafle, "Fairdedup: Detecting and mitigating vision-language fairness disparities in semantic dataset deduplication," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 13 905–13 916.

[774] Y. Zhang, J. Wang, and J. Sang, "Counterfactually measuring and eliminating social bias in vision-language pre-training models," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 4996–5004.

[775] K. C. Fraser and S. Kiritchenko, "Examining gender and racial bias in large vision-language models using a novel dataset of parallel images," *arXiv preprint arXiv:2402.05779*, 2024.

[776] A. Seth, M. Hemani, and C. Agarwal, "Dear: Debiasing vision-language models with additive residuals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6820–6829.

[777] S. Janghorbani and G. De Melo, "Multimodal bias: Introducing a framework for stereotypical bias assessment beyond gender and race in vision language models," *arXiv preprint arXiv:2303.12734*, 2023.

[778] Y. Zhang, Y. Huang, Y. Sun, C. Liu, Z. Zhao, Z. Fang, Y. Wang, H. Chen, X. Yang, X. Wei *et al.*, "Benchmarking trustworthiness of multimodal large language models: A comprehensive study," *arXiv preprint arXiv:2406.07057*, 2024.

[779] Y. Zhang, L. Chen, G. Zheng, Y. Gao, R. Zheng, J. Fu, Z. Yin, S. Jin, Y. Qiao, X. Huang *et al.*, "Spa-vl: A comprehensive safety preference alignment dataset for vision language model," *arXiv preprint arXiv:2406.12030*, 2024.

[780] Z. Zhang, T. Kou, S. Wang, C. Li, W. Sun, W. Wang, X. Li, Z. Wang, X. Cao, X. Min *et al.*, "Q-eval-100k: Evaluating visual quality and alignment level for text-to-vision content," *arXiv preprint arXiv:2503.02357*, 2025.

[781] W. Wang, X. Liu, K. Gao, J.-T. Huang, Y. Yuan, P. He, S. Wang, and Z. Tu, "Can't see the forest for the trees: Benchmarking multimodal safety awareness for multimodal llms," *ArXiv*, vol. abs/2502.11184, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:276409442

[782] W. Wang, K. Gao, Z. Jia, Y. Yuan, J.-T. Huang, Q. Liu, S. Wang, W. Jiao, and Z. Tu, "Chain-of-jailbreak attack for image generation models via editing step by step," *ArXiv*, vol. abs/2410.03869, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:273186566

[783] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," *arXiv preprint arXiv:2307.06435*, 2023.

[784] W. Zhao, Y. Hu, Y. Deng, J. Guo, X. Sui, X. Han, A. Zhang, Y. Zhao, B. Qin, T.-S. Chua *et al.*, "Beware of your po! measuring and mitigating ai safety risks in role-play fine-tuning of llms," *arXiv preprint arXiv:2502.20968*, 2025.

[785] B. Liu, X. Li, J. Zhang, J. Wang, T. He, S. Hong, H. Liu, S. Zhang, K. Song, K. Zhu *et al.*, "Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems," *arXiv preprint arXiv:2504.01990*, 2025.

[786] H. Jin, L. Huang, H. Cai, J. Yan, B. Li, and H. Chen, "From llms to llm-based agents for software engineering: A survey of current, challenges and future," *arXiv preprint arXiv:2408.02479*, 2024.

[787] J. Piao, Y. Yan, J. Zhang, N. Li, J. Yan, X. Lan, Z. Lu, Z. Zheng, J. Y. Wang, D. Zhou *et al.*, "Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society," *arXiv preprint arXiv:2502.08691*, 2025.

[788] Y. Yan, S. Wang, J. Huo, P. S. Yu, X. Hu, and Q. Wen, "Mathagent: Leveraging a mixture-of-math-agent framework for real-world multimodal mathematical error detection," *arXiv preprint arXiv:2503.18132*, 2025.

[789] H. Wang, A. Zhang, N. Duy Tai, J. Sun, T.-S. Chua *et al.*, “Ali-agent: Assessing llms’ alignment with human values via agent-based evaluation,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 99 040–99 088, 2024.

[790] K. Zhang, J. Li, G. Li, X. Shi, and Z. Jin, “Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges,” *arXiv preprint arXiv:2401.07339*, 2024.

[791] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, “Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 38 154–38 180, 2023.

[792] Z. Chu, S. Wang, J. Xie, T. Zhu, Y. Yan, J. Ye, A. Zhong, X. Hu, J. Liang, P. S. Yu *et al.*, “Llm agents for education: Advances and applications,” *arXiv preprint arXiv:2503.11733*, 2025.

[793] W. Zhang, Y. Shen, W. Lu, and Y. Zhuang, “Data-copilot: Bridging billions of data and humans with autonomous workflow,” *arXiv preprint arXiv:2306.07209*, 2023.

[794] W. Xu, Z. Liang, K. Mei, H. Gao, J. Tan, and Y. Zhang, “A-mem: Agentic memory for llm agents,” *arXiv preprint arXiv:2502.12110*, 2025.

[795] Y. Shang, Y. Li, K. Zhao, L. Ma, J. Liu, F. Xu, and Y. Li, “Agentsquare: Automatic llm agent search in modular design space,” *arXiv preprint arXiv:2410.06153*, 2024.

[796] J. Yang, C. Jimenez, A. Wettig, K. Lieret, S. Yao, K. Narasimhan, and O. Press, “Swe-agent: Agent-computer interfaces enable automated software engineering,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 50 528–50 652, 2024.

[797] S. Agashe, J. Han, S. Gan, J. Yang, A. Li, and X. E. Wang, “Agent s: An open agentic framework that uses computers like a human,” *arXiv preprint arXiv:2410.08164*, 2024.

[798] S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, and Z. Hu, “Reasoning with language model is planning with world model,” *arXiv preprint arXiv:2305.14992*, 2023.

[799] J. Hong, J. Lin, A. Dragan, and S. Levine, “Interactive dialogue agents via reinforcement learning on hindsight regenerations,” *arXiv preprint arXiv:2411.05194*, 2024.

[800] J. Tang, T. Fan, and C. Huang, “Autoagent: A fully-automated and zero-code framework for llm agents,” *arXiv e-prints*, pp. arXiv–2502, 2025.

[801] G. Li, H. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, “Camel: Communicative agents for" mind" exploration of large language model society,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 51 991–52 008, 2023.

[802] S. Yuan, K. Song, J. Chen, X. Tan, D. Li, and D. Yang, “Evoagent: Towards automatic multi-agent generation via evolutionary algorithms,” *arXiv preprint arXiv:2406.14228*, 2024.

[803] M. Zhuge, W. Wang, L. Kirsch, F. Faccio, D. Khizbullin, and J. Schmidhuber, “Language agents as optimizable graphs,” *arXiv preprint arXiv:2402.16823*, 2024.

[804] Y. Wang, T. Shen, L. Liu, and J. Xie, “Sibyl: Simple yet effective agent framework for complex real-world reasoning,” *arXiv preprint arXiv:2407.10718*, 2024.

[805] Z. Wang, X. Zeng, W. Liu, L. Li, Y. Wang, L. Shang, X. Jiang, Q. Liu, and K.-F. Wong, “Toolflow: Boosting llm tool-calling through natural and coherent dialogue synthesis,” *arXiv preprint arXiv:2410.18447*, 2024.

[806] F. Wu, S. Wu, Y. Cao, and C. Xiao, “Wipi: A new web threat for llm-driven web agents,” *arXiv preprint arXiv:2402.16965*, 2024.

[807] S. S. Kannan, V. L. Venkatesh, and B.-C. Min, “Smart-llm: Smart multi-agent robot task planning using large language models,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 140–12 147.

[808] R. Fang, R. Bindu, A. Gupta, and D. Kang, “Llm agents can autonomously exploit one-day vulnerabilities,” *arXiv preprint arXiv:2404.08144*, vol. 13, p. 14, 2024.

[809] R. Fang, R. Bindu, A. Gupta, Q. Zhan, and D. Kang, “Llm agents can autonomously hack websites,” *arXiv preprint arXiv:2402.06664*, 2024.

[810] W. Cheng, K. Sun, X. Zhang, and W. Wang, “Security attacks on llm-based code completion tools,” *arXiv preprint arXiv:2408.11006*, 2024.

[811] X. Fu, Z. Wang, S. Li, R. K. Gupta, N. Mireshghallah, T. Berg-Kirkpatrick, and E. Fernandes, “Misusing tools in large language models with visual adversarial examples,” *arXiv preprint arXiv:2310.03185*, 2023.

[812] X. Fu, S. Li, Z. Wang, Y. Liu, R. K. Gupta, T. Berg-Kirkpatrick, and E. Fernandes, “Imprompter: Tricking llm agents into improper tool use,” *arXiv preprint arXiv:2410.14923*, 2024.

[813] B. Zhang, Y. Tan, Y. Shen, A. Salem, M. Backes, S. Zannettou, and Y. Zhang, “Breaking agents: Compromising autonomous llm agents through malfunction amplification,” *arXiv preprint arXiv:2407.20859*, 2024.

[814] H. Wang, R. Zhang, J. Wang, M. Li, Y. Huang, D. Wang, and Q. Wang, “From allies to adversaries: Manipulating llm tool-calling through adversarial injection,” *arXiv preprint arXiv:2412.10198*, 2024.

[815] W. Yang, X. Bi, Y. Lin, S. Chen, J. Zhou, and X. Sun, “Watch out for your agents! investigating backdoor threats to llm-based agents,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 100 938–100 964, 2024.

[816] P. Zhu, Z. Zhou, Y. Zhang, S. Yan, K. Wang, and S. Su, “Demonagent: Dynamically encrypted multi-backdoor implantation attack on llm-based agent,” *arXiv preprint arXiv:2502.12575*, 2025.

[817] Y. Wang, D. Xue, S. Zhang, and S. Qian, “Badagent: Inserting and activating backdoor attacks in llm agents,” *arXiv preprint arXiv:2406.03007*, 2024.

[818] Z. Jiang, M. Li, G. Yang, J. Wang, Y. Huang, Z. Chang, and Q. Wang, “Mimicking the familiar: Dynamic command generation for information theft attacks in llm tool-learning system,” *arXiv preprint arXiv:2502.11358*, 2025.

[819] W. Zhao, V. Khazanchi, H. Xing, X. He, Q. Xu, and N. D. Lane, “Attacks on third-party apis of large language models,” *arXiv preprint arXiv:2404.16891*, 2024.

[820] J. Chen and S. L. Cong, “Agentguard: Repurposing

agentic orchestrator for safety evaluation of tool orchestration," *arXiv preprint arXiv:2502.09809*, 2025.

[821] X. Zhang, H. Xu, Z. Ba, Z. Wang, Y. Hong, J. Liu, Z. Qin, and K. Ren, "Privacyasst: Safeguarding user privacy in tool-using large language model agents," *IEEE Transactions on Dependable and Secure Computing*, 2024.

[822] Z. Xiang, L. Zheng, Y. Li, J. Hong, Q. Li, H. Xie, J. Zhang, Z. Xiong, C. Xie, C. Yang *et al.*, "Guardagent: Safeguard llm agents by a guard agent via knowledge-enabled reasoning," *arXiv preprint arXiv:2406.09187*, 2024.

[823] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, vol. 2, 2023.

[824] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui, "Retrieval-augmented generation for ai-generated content: A survey," *arXiv preprint arXiv:2402.19473*, 2024.

[825] C. Xiang, T. Wu, Z. Zhong, D. Wagner, D. Chen, and P. Mittal, "Certifiably robust rag against retrieval corruption," *arXiv preprint arXiv:2405.15556*, 2024.

[826] Z. Chen, Z. Xiang, C. Xiao, D. Song, and B. Li, "Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases," *Advances in Neural Information Processing Systems*, vol. 37, pp. 130 185–130 213, 2025.

[827] W. Zou, R. Geng, B. Wang, and J. Jia, "Poisonedrag: Knowledge corruption attacks to retrieval-augmented generation of large language models," *arXiv preprint arXiv:2402.07867*, 2024.

[828] Z. Zhong, Z. Huang, A. Wettig, and D. Chen, "Poisoning retrieval corpora by injecting adversarial passages," *arXiv preprint arXiv:2310.19156*, 2023.

[829] X. Gu, X. Zheng, T. Pang, C. Du, Q. Liu, Y. Wang, J. Jiang, and M. Lin, "Agent smith: A single image can jailbreak one million multimodal llm agents exponentially fast," *arXiv preprint arXiv:2402.08567*, 2024.

[830] A. Li, Y. Zhou, V. C. Raghuram, T. Goldstein, and M. Goldblum, "Commercial llm agents are already vulnerable to simple yet dangerous attacks," *arXiv preprint arXiv:2502.08586*, 2025.

[831] H. Li, M. Xu, and Y. Song, "Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence," *arXiv preprint arXiv:2305.03010*, 2023.

[832] M. Russinovich, A. Salem, and R. Eldan, "Great, now write an article about that: The crescendo multi-turn llm jailbreak attack," *arXiv preprint arXiv:2404.01833*, 2024.

[833] Y. Cheng, M. Georgopoulos, V. Cevher, and G. G. Chrysos, "Leveraging the context through multi-round interactions for jailbreaking attacks," *arXiv preprint arXiv:2402.09177*, 2024.

[834] A. Priyanshu and S. Vijay, "Fractured-sorry-bench: Framework for revealing attacks in conversational turns undermining refusal efficacy and defenses over sorry-bench (automated multi-shot jailbreaks)," *arXiv preprint arXiv:2408.16163*, 2024.

[835] D. Agarwal, A. R. Fabbri, B. Risher, P. Laban, S. Joty, and C.-S. Wu, "Prompt leakage effect and defense strategies for multi-turn llm interactions," *arXiv preprint arXiv:2404.16251*, 2024.

[836] T. Tong, J. Xu, Q. Liu, and M. Chen, "Securing multi-turn conversational language models from distributed backdoor triggers," *arXiv preprint arXiv:2407.04151*, 2024.

[837] J. Mao, F. Meng, Y. Duan, M. Yu, X. Jia, J. Fang, Y. Liang, K. Wang, and Q. Wen, "Agentsafe: Safeguarding large language model-based multi-agent systems via hierarchical data management," *arXiv preprint arXiv:2503.04392*, 2025.

[838] H. Zhou, K.-H. Lee, Z. Zhan, Y. Chen, and Z. Li, "Trustrag: Enhancing robustness and trustworthiness in rag," *arXiv preprint arXiv:2501.00879*, 2025.

[839] X. Xian, G. Wang, X. Bi, J. Srinivasa, A. Kundu, C. Fleming, M. Hong, and J. Ding, "On the vulnerability of applying retrieval-augmented generation within knowledge-intensive application domains," *arXiv preprint arXiv:2409.17275*, 2024.

[840] B. Chen, G. Wang, H. Guo, Y. Wang, and Q. Yan, "Understanding multi-turn toxic behaviors in open-domain chatbots," in *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, 2023, pp. 282–296.

[841] R. Song, M. O. Ozmen, H. Kim, A. Bianchi, and Z. B. Celik, "Enhancing llm-based autonomous driving agents to mitigate perception attacks," *arXiv preprint arXiv:2409.14488*, 2024.

[842] C. H. Low, Z. Wang, T. Zhang, Z. Zeng, Z. Zhuo, E. B. Mazomenos, and Y. Jin, "Surgraw: Multi-agent workflow with chain-of-thought reasoning for surgical intelligence," *arXiv preprint arXiv:2503.10265*, 2025.

[843] Z. Wang, J. Wu, C. H. Low, and Y. Jin, "Medagent-pro: Towards multi-modal evidence-based medical diagnosis via reasoning agentic workflow," *arXiv preprint arXiv:2503.18968*, 2025.

[844] K. N. Jeptoo and C. Sun, "Enhancing fake news detection with large language models through multi-agent debates," in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2024, pp. 474–486.

[845] T. Park, "Enhancing anomaly detection in financial markets with an llm-based multi-agent framework," *arXiv preprint arXiv:2403.19735*, 2024.

[846] Z. Yang, S. S. Raman, A. Shah, and S. Tellex, "Plug in the safety chip: Enforcing constraints for llm-driven robot agents," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 435–14 442.

[847] J. Zhang, C. Xu, and B. Li, "Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 459–15 469.

[848] T. Abuelsaad, D. Akkil, P. Dey, A. Jagmohan, A. Vempaty, and R. Kokku, "Agent-e: From autonomous web navigation to foundational design principles in agentic systems," *arXiv preprint arXiv:2407.13032*, 2024.

[849] E. Debenedetti, J. Zhang, M. Balunović, L. Beurer-Kellner, M. Fischer, and F. Tramèr, "Agentdojo: A dy-

namic environment to evaluate attacks and defenses for llm agents," *arXiv preprint arXiv:2406.13352*, 2024.

[850] Y. Sun, N. Salami Pargoo, P. Jin, and J. Ortiz, "Optimizing autonomous driving for safety: A human-centric approach with llm-enhanced rlhf," in *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2024, pp. 76–80.

[851] R. Fang, R. Bindu, A. Gupta, and D. Kang, "Llm agents can autonomously exploit one-day vulnerabilities," *arXiv preprint arXiv:2404.08144*, vol. 13, p. 14, 2024.

[852] Y. H. Ke, R. Yang, S. A. Lie, T. X. Y. Lim, H. R. Abdullah, D. S. W. Ting, and N. Liu, "Enhancing diagnostic accuracy through multi-agent conversations: using large language models to mitigate cognitive bias," *arXiv preprint arXiv:2401.14589*, 2024.

[853] X. Mou, Z. Wei, and X. Huang, "Unveiling the truth and facilitating change: Towards agent-based large-scale social movement simulation," *arXiv preprint arXiv:2402.16333*, 2024.

[854] Z. Chen, J. Chen, J. Chen, and M. Sra, "Position: Standard benchmarks fail–llm agents present overlooked risks for financial applications," *arXiv preprint arXiv:2502.15865*, 2025.

[855] Z. Liu, R. Zeng, D. Wang, G. Peng, J. Wang, Q. Liu, P. Liu, and W. Wang, "Agents4plc: Automating closed-loop plc code generation and verification in industrial control systems using llm-based agents," *arXiv preprint arXiv:2410.14209*, 2024.

[856] S. Mukherjee, P. Gamble, M. S. Ausin, N. Kant, K. Aggarwal, N. Manjunath, D. Datta, Z. Liu, J. Ding, S. Busacca *et al.*, "Polaris: A safety-focused llm constellation architecture for healthcare," *arXiv preprint arXiv:2403.13313*, 2024.

[857] L. La Cava and A. Tagarelli, "Safeguarding decentralized social media: Llm agents for automating community rule compliance," *arXiv preprint arXiv:2409.08963*, 2024.

[858] Y. Gan, Y. Yang, Z. Ma, P. He, R. Zeng, Y. Wang, Q. Li, C. Zhou, S. Li, T. Wang *et al.*, "Navigating the risks: A survey of security, privacy, and ethics threats in llm-based agents," *arXiv preprint arXiv:2411.09523*, 2024.

[859] Z. Deng, Y. Guo, C. Han, W. Ma, J. Xiong, S. Wen, and Y. Xiang, "Ai agents under threat: A survey of key security challenges and future pathways," *ACM Computing Surveys*, 2024.

[860] R. Ye, S. Tang, R. Ge, Y. Du, Z. Yin, S. Chen, and J. Shao, "Mas-gpt: Training llms to build llm-based multi-agent systems," *arXiv preprint arXiv:2503.03686*, 2025.

[861] J. Zhang, J. Xiang, Z. Yu, F. Teng, X. Chen, J. Chen, M. Zhuge, X. Cheng, S. Hong, J. Wang *et al.*, "Aflow: Automating agentic workflow generation," *arXiv preprint arXiv:2410.10762*, 2024.

[862] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous agents and multi-agent systems*, vol. 11, pp. 387–434, 2005.

[863] L. Hammond, A. Chan, J. Clifton, J. Hoelscher-Obermaier, A. Khan, E. McLean, C. Smith, W. Barfuss, J. Foerster, T. Gavenčiak *et al.*, "Multi-agent risks from advanced ai," *arXiv preprint arXiv:2502.14143*, 2025.

[864] R. Xu, X. Li, S. Chen, and W. Xu, "Nuclear deployed: Analyzing catastrophic risks in decision-making of autonomous llm agents," *arXiv preprint arXiv:2502.11355*, 2025.

[865] Z. Zhou, Z. Li, J. Zhang, Y. Zhang, K. Wang, Y. Liu, and Q. Guo, "Corba: Contagious recursive blocking attacks on multi-agent systems based on large language models," *arXiv preprint arXiv:2502.14529*, 2025.

[866] Z. Tan, C. Zhao, R. Moraffah, Y. Li, Y. Kong, T. Chen, and H. Liu, "The wolf within: Covert injection of malice into mllm societies via an mllm operative," *arXiv preprint arXiv:2402.14859*, 2024.

[867] M. Yu, S. Wang, G. Zhang, J. Mao, C. Yin, Q. Liu, Q. Wen, K. Wang, and Y. Wang, "Netsafe: Exploring the topological safety of multi-agent networks," *arXiv preprint arXiv:2410.15686*, 2024.

[868] J.-t. Huang, J. Zhou, T. Jin, X. Zhou, Z. Chen, W. Wang, Y. Yuan, M. Sap, and M. R. Lyu, "On the resilience of multi-agent systems with malicious agents," *arXiv preprint arXiv:2408.00989*, 2024.

[869] P. He, Y. Lin, S. Dong, H. Xu, Y. Xing, and H. Liu, "Red-teaming llm multi-agent systems via communication attacks," *arXiv preprint arXiv:2502.14847*, 2025.

[870] Y. Tian, X. Yang, J. Zhang, Y. Dong, and H. Su, "Evil geniuses: Delving into the safety of llm-based agents," *arXiv preprint arXiv:2311.11855*, 2023.

[871] A. Amayuelas, X. Yang, A. Antoniades, W. Hua, L. Pan, and W. Wang, "Multiagent collaboration attack: Investigating adversarial attacks in large language model collaborations via debate," *arXiv preprint arXiv:2406.14711*, 2024.

[872] T. Ju, Y. Wang, X. Ma, P. Cheng, H. Zhao, Y. Wang, L. Liu, J. Xie, Z. Zhang, and G. Liu, "Flooding spread of manipulated knowledge in llm-based multi-agent communities," *arXiv preprint arXiv:2407.07791*, 2024.

[873] G. Lin and Q. Zhao, "Large language model sentinel: Llm agent for adversarial purification," *arXiv preprint arXiv:2405.20770*, 2024.

[874] Y. Zeng, Y. Wu, X. Zhang, H. Wang, and Q. Wu, "Autodefense: Multi-agent llm defense against jailbreak attacks," *arXiv preprint arXiv:2403.04783*, 2024.

[875] S. Chern, Z. Fan, and A. Liu, "Combating adversarial attacks with multi-agent debate," *arXiv preprint arXiv:2401.05998*, 2024.

[876] B. Chen, G. Li, X. Lin, Z. Wang, and J. Li, "Blockagents: Towards byzantine-robust llm-based multi-agent coordination via blockchain," in *Proceedings of the ACM Turing Award Celebration Conference-China 2024*, 2024, pp. 187–192.

[877] C. Song, L. Ma, J. Zheng, J. Liao, H. Kuang, and L. Yang, "Audit-llm: Multi-agent collaboration for log-based insider threat detection," *arXiv preprint arXiv:2408.08902*, 2024.

[878] S. Wang, G. Zhang, M. Yu, G. Wan, F. Meng, C. Guo, K. Wang, and Y. Wang, "G-safeguard: A topology-guided security lens and treatment on llm-based multi-agent systems," *arXiv preprint arXiv:2502.11127*, 2025.

[879] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and*

learning systems, vol. 32, no. 1, pp. 4–24, 2020.

[880] X. Zheng, Y. Wang, Y. Liu, M. Li, M. Zhang, D. Jin, P. S. Yu, and S. Pan, "Graph neural networks for graphs with heterophily: A survey," *arXiv preprint arXiv:2202.07082*, 2022.

[881] M. R. Genesereth and S. P. Ketchpel, "The kqml protocol: A specification of language and communication," in *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM)*. ACM, 1993, pp. 1–10.

[882] D. S. Milojicic, M. Breugst, I. Busse, J. Campbell, S. Covaci, B. Friedman, K. Kosaka, D. B. Lange, K. Ono, M. Oshima, C. Tham, S. Virdhagriswaran, and J. White, "Masif: The omg mobile agent system interoperability facility," in *Proceedings of the Second International Workshop on Mobile Agents*, ser. MA '98. Berlin, Heidelberg: Springer-Verlag, 1998, p. 50–67.

[883] F. for Intelligent Physical Agents, "Fipa communicative act library specification," https://www.fipa.org/specs/fipa00037/SC00037J.html, 2000.

[884] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, "Web services: Why and how," *IBM Systems Journal*, vol. 41, no. 2, pp. 170–177, 2002.

[885] G. Hohpe and B. Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, ser. Addison-Wesley Signature Series (Fowler). Addison-Wesley Professional, 2006.

[886] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[887] G. Izacard and E. Grave, "Towards an efficient pipeline for knowledge-intensive nlp tasks," *arXiv preprint arXiv:2112.04426*, 2021.

[888] H. Chase, "Langchain: Build applications with llms through composability," https://github.com/langchain-ai/langchain, 2022, accessed: Apr. 2025.

[889] J. Wu *et al.*, "Llamaindex: Connecting llms to your knowledge," https://github.com/jerryjliu/llama_index, 2023, accessed: Apr. 2025.

[890] OpenAI, "Function calling in openai models," https://platform.openai.com/docs/guides/functions, 2023, accessed: Apr. 2025.

[891] Anthropic, "Model context protocol," 2024, accessed: 2025-04-19. [Online]. Available: https://www.anthropic.com/news/model-context-protocol

[892] Google, "A2a: Agent2agent protocol," 2025, accessed: 2025-04-21. [Online]. Available: https://github.com/google/A2A

[893] G. Chang, "Anp: Agent network protocol," 2024, accessed: 2025-04-21. [Online]. Available: https://www.agent-network-protocol.com/

[894] WildCardAI, "agents.json specification," https://github.com/wild-card-ai/agents-json, 2025, accessed: 2025-04-22.

[895] NEAR, "Aitp: Agent interaction & transaction protocol," 2025, accessed: 2025-04-22. [Online]. Available: https://aitp.dev/

[896] L. F. AI and L. Data, "Acp: Agent communication protocol," 2025, accessed: 2025-04-22. [Online]. Available: https://github.com/orgs/i-am-bee/discussions/284

[897] G. Cisco, Langchain, "Acp: Agent connect protocol," 2025, accessed: 2025-04-22. [Online]. Available: https://spec.acp.agntcy.org/

[898] S. Marro, E. L. Malfa, J. Wright, G. Li, N. Shadbolt, M. Wooldridge, and P. Torr, "A scalable communication protocol for networks of large language models," 2024. [Online]. Available: https://arxiv.org/abs/2410.11905

[899] Eclipse, "Language model operating system (lmos)," https://eclipse.dev/lmos/, 2025, accessed: 2025-04-22.

[900] AIEngineerFoundation, "Agent protocol," https://agentprotocol.ai/, 2025, accessed: 2025-04-22.

[901] R. Ranjan, S. Gupta, and S. N. Singh, "Loka protocol: A decentralized framework for trustworthy and ethical ai agent ecosystems," 2025. [Online]. Available: https://arxiv.org/abs/2504.10915

[902] A. Srinivasan, K. Bania, S. V, H. Mestha, and S. Liu, "Implementation and application of an intelligibility protocol for interaction with an llm," 2024. [Online]. Available: https://arxiv.org/abs/2410.20600

[903] I. Bae, J. Lee, and H.-G. Jeon, "Continuous locomotive crowd behavior generation," 2025. [Online]. Available: https://arxiv.org/abs/2504.04756

[904] L. Gąsieniec, Łukasz Kuszner, E. Latif, R. Parasuraman, P. Spirakis, and G. Stachowiak, "Anonymous distributed localisation via spatial population protocols," 2024. [Online]. Available: https://arxiv.org/abs/2411.08434

[905] J. Tu, T. Wang, J. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "Adversarial attacks on multi-agent communication," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7768–7777.

[906] L. Yuan, F. Chen, Z. Zhang, and Y. Yu, "Communication-robust multi-agent learning by adaptable auxiliary multi-agent adversary generation," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186331, 2024.

[907] J. Blumenkamp and A. Prorok, "The emergence of adversarial communication in multi-agent reinforcement learning," in *Conference on Robot Learning*. PMLR, 2021, pp. 1394–1414.

[908] Z. Chen, Z. Xiang, C. Xiao, D. Song, and B. Li, "Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[909] X. Pan, J. Dai, Y. Fan, and M. Yang, "Frontier ai systems have surpassed the self-replicating red line," *arXiv preprint arXiv:2412.12140*, 2024.

[910] L. Yu, Y. Qiu, Q. Yao, Y. Shen, X. Zhang, and J. Wang, "Robust communicative multi-agent reinforcement learning with active defense," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17 575–17 582.

[911] J. Light, M. Cai, S. Shen, and Z. Hu, "Avalonbench: Evaluating llms playing the game of avalon," *arXiv preprint arXiv:2310.05036*, 2023.

[912] Q. Xie, Q. Feng, T. Zhang, Q. Li, L. Yang, Y. Zhang,

R. Feng, L. He, S. Gao, and Y. Zhang, "Human simulacra: Benchmarking the personification of large language models," *arXiv preprint arXiv:2402.18180*, 2024.

[913] L. Geng and E. Y. Chang, "Realm-bench: A real-world planning benchmark for llms and multi-agent systems," *arXiv preprint arXiv:2502.18836*, 2025.

[914] Y. Dubois, B. Galambosi, P. Liang, and T. B. Hashimoto, "Length-controlled alpacaeval: A simple way to debias automatic evaluators," *arXiv preprint arXiv:2404.04475*, 2024.

[915] W. Wang, J. Shi, C. Wang, C. Lee, Y. Yuan, J.-T. Huang, and M. R. Lyu, "Learning to ask: When llms meet unclear instruction," *ArXiv*, vol. abs/2409.00557, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272368496

[916] C. Guo, X. Liu, C. Xie, A. Zhou, Y. Zeng, Z. Lin, D. Song, and B. Li, "Redcode: Risky code execution and generation benchmark for code agents," *Advances in Neural Information Processing Systems*, vol. 37, pp. 106 190–106 236, 2024.

[917] X. Yuan, J. Li, D. Wang, Y. Chen, X. Mao, L. Huang, H. Xue, W. Wang, K. Ren, and J. Wang, "S-eval: Automatic and adaptive test generation for benchmarking safety evaluation of large language models," *arXiv preprint arXiv:2405.14191*, 2024.

[918] D. Dorn, A. Variengien, C.-R. Segerie, and V. Corruble, "Bells: A framework towards future proof benchmarks for the evaluation of llm safeguards," *arXiv preprint arXiv:2406.01364*, 2024.

[919] Y. Shao, T. Li, W. Shi, Y. Liu, and D. Yang, "Privacylens: Evaluating privacy norm awareness of language models in action," *arXiv preprint arXiv:2409.00138*, 2024.

[920] Q. Zhan, Z. Liang, Z. Ying, and D. Kang, "Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents," *arXiv preprint arXiv:2403.02691*, 2024.

[921] Z. Zhu, B. Wu, Z. Zhang, and B. Wu, "Riskawarebench: Towards evaluating physical risk awareness for high-level planning of llm-based embodied agents," *arXiv e-prints*, pp. arXiv–2408, 2024.

[922] Z. Zhang, S. Cui, Y. Lu, J. Zhou, J. Yang, H. Wang, and M. Huang, "Agent-safetybench: Evaluating the safety of llm agents," *arXiv preprint arXiv:2412.14470*, 2024.

[923] M. Andriushchenko, A. Souly, M. Dziemian, D. Duenas, M. Lin, J. Wang, D. Hendrycks, A. Zou, Z. Kolter, M. Fredrikson *et al.*, "Agentharm: A benchmark for measuring harmfulness of llm agents," *arXiv preprint arXiv:2410.09024*, 2024.

[924] J. Ye, S. Li, G. Li, C. Huang, S. Gao, Y. Wu, Q. Zhang, T. Gui, and X. Huang, "Toolsword: Unveiling safety issues of large language models in tool learning across three stages," *arXiv preprint arXiv:2402.10753*, 2024.

[925] Y. Ruan, H. Dong, A. Wang, S. Pitis, Y. Zhou, J. Ba, Y. Dubois, C. J. Maddison, and T. Hashimoto, "Identifying the risks of lm agents with an lm-emulated sandbox," *arXiv preprint arXiv:2309.15817*, 2023.

[926] X. Zhou, H. Kim, F. Brahman, L. Jiang, H. Zhu, X. Lu, F. Xu, B. Y. Lin, Y. Choi, N. Mireshghallah *et al.*, "Haicosystem: An ecosystem for sandboxing safety risks in human-ai interactions," *arXiv preprint arXiv:2409.16427*, 2024.

[927] S. Yin, X. Pang, Y. Ding, M. Chen, Y. Bi, Y. Xiong, W. Huang, Z. Xiang, J. Shao, and S. Chen, "Safeagentbench: A benchmark for safe task planning of embodied llm agents," *arXiv preprint arXiv:2412.13178*, 2024.

[928] J. BENCHMARK, "Jailjudge: Acomprehensive jailbreak judge benchmark with multi-agent enhanced explanation evaluation framework."

[929] P. Y. Zhong, S. Chen, R. Wang, M. McCall, B. L. Titzer, and H. Miller, "Rtbas: Defending llm agents against prompt injection and privacy leakage," *arXiv preprint arXiv:2502.08966*, 2025.

[930] A. Liu, Y. Zhou, X. Liu, T. Zhang, S. Liang, J. Wang, Y. Pu, T. Li, J. Zhang, W. Zhou *et al.*, "Compromising llm driven embodied agents with contextual backdoor attacks," *IEEE Transactions on Information Forensics and Security*, 2025.

[931] ——, "Compromising embodied agents with contextual backdoor attacks," *arXiv preprint arXiv:2408.02882*, 2024.

[932] H. Zhang, C. Zhu, X. Wang, Z. Zhou, S. Hu, and L. Y. Zhang, "Badrobot: Jailbreaking llm-based embodied ai in the physical world," *arXiv preprint arXiv:2407.20242*, 2024.

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

[934] S. Yuan, K. Song, J. Chen, X. Tan, Y. Shen, R. Kan, D. Li, and D. Yang, "Easytool: Enhancing llm-based agents with concise tool instruction," *arXiv preprint arXiv:2401.06201*, 2024.

[935] S. Wu, S. Zhao, Q. Huang, K. Huang, M. Yasunaga, K. Cao, V. Ioannidis, K. Subbian, J. Leskovec, and J. Y. Zou, "Avatar: Optimizing llm agents for tool usage via contrastive reasoning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 25 981–26 010, 2024.

[936] Z. Shen, "Llm with tools: A survey," *arXiv preprint arXiv:2409.18807*, 2024.

[937] C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong *et al.*, "Chatdev: Communicative agents for software development," *arXiv preprint arXiv:2307.07924*, 2023.

[938] Z. M. Wang, Z. Peng, H. Que, J. Liu, W. Zhou, Y. Wu, H. Guo, R. Gan, Z. Ni, J. Yang *et al.*, "Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models," *arXiv preprint arXiv:2310.00746*, 2023.

[939] J. Zhou, Z. Chen, D. Wan, B. Wen, Y. Song, J. Yu, Y. Huang, L. Peng, J. Yang, X. Xiao *et al.*, "Characterglm: Customizing chinese conversational ai characters with large language models," *arXiv preprint arXiv:2311.16832*, 2023.

[940] Z. Chen, K. Liu, Q. Wang, W. Zhang, J. Liu, D. Lin, K. Chen, and F. Zhao, "Agent-flan: Designing data and methods of effective agent tuning for large language models," *arXiv preprint arXiv:2403.12881*, 2024.

[941] G. Zhang, L. Niu, J. Fang, K. Wang, L. Bai, and X. Wang, "Multi-agent architecture search via agentic supernet," *arXiv preprint arXiv:2502.04180*, 2025.

[942] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[943] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017.

[944] X. Li, Y. Fan, and S. Cheng, "Aigc in china: Current developments and future outlook," *arXiv preprint arXiv:2308.08451*, 2023.

[945] X. Sun, L. Dong, X. Li, Z. Wan, S. Wang, T. Zhang, J. Li, F. Cheng, L. Lyu, F. Wu *et al.*, "Pushing the limits of chatgpt on nlp tasks," *arXiv preprint arXiv:2306.09719*, 2023.

[946] G. Sriramanan, S. Bharti, V. S. Sadasivan, S. Saha, P. Kattakinda, and S. Feizi, "Llm-check: Investigating detection of hallucinations in large language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 34 188–34 216, 2024.

[947] K. Zheng, J. Chen, Y. Yan, X. Zou, and X. Hu, "Reefknot: A comprehensive benchmark for relation hallucination evaluation, analysis and mitigation in multimodal large language models," *arXiv preprint arXiv:2408.09429*, 2024.

[948] X. Zou, Y. Wang, Y. Yan, S. Huang, K. Zheng, J. Chen, C. Tang, and X. Hu, "Look twice before you answer: Memory-space visual retracing for hallucination mitigation in multimodal large language models," *arXiv preprint arXiv:2410.03577*, 2024.

[949] G. Zhou, Y. Yan, X. Zou, K. Wang, A. Liu, and X. Hu, "Mitigating modality prior-induced hallucinations in multimodal large language models via deciphering attention causality," *arXiv preprint arXiv:2410.04780*, 2024.

[950] W. Wang, Z. Ma, Z. Wang, C. Wu, W. Chen, X. Li, and Y. Yuan, "A survey of llm-based agents in medicine: How far are we from baymax?" *ArXiv*, vol. abs/2502.11211, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:276408182

[951] H. Kang and X.-Y. Liu, "Deficiency of large language models in finance: An empirical examination of hallucination," *arXiv preprint arXiv:2311.15548*, 2023.

[952] 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. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *NeurIPS*, 2022.

[953] Y. Liu, Y. Yao, J.-F. Ton, X. Zhang, R. Guo, H. Cheng, Y. Klochkov, M. F. Taufiq, and H. Li, "Trustworthy llms: a survey and guideline for evaluating large language models' alignment," 2024.

[954] M. Hao, H. Li, H. Chen, P. Xing, G. Xu, and T. Zhang, "Iron: Private inference on transformers," *Advances in neural information processing systems*, vol. 35, pp. 15 718–15 731, 2022.

[955] J. Huang, J.-T. Huang, Z. Liu, X. Liu, W. Wang, and J. Zhao, "Vlms as geoguessr masters: Exceptional performance, hidden biases, and privacy risks," *ArXiv*, vol. abs/2502.11163, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:276409319

[956] G. Feretzakis and V. S. Verykios, "Trustworthy ai: Securing sensitive data in large language models," *AI*, vol. 5, no. 4, pp. 2773–2800, 2024.

[957] Q. Feng, S. R. Kasa, H. Yun, C. H. Teo, and S. B. Bodapati, "Exposing privacy gaps: Membership inference attack on preference data for llm alignment," *arXiv preprint arXiv:2407.06443*, 2024.

[958] N. Rahman and E. Santacana, "Beyond fair use: Legal risk evaluation for training llms on copyrighted text," in *ICML Workshop on Generative AI and Law*, 2023.

[959] J. Guo, Y. Li, R. Chen, Y. Wu, C. Liu, Y. Chen, and H. Huang, "Towards copyright protection for knowledge bases of retrieval-augmented language models via ownership verification with reasoning," *arXiv preprint arXiv:2502.10440*, 2025.

[960] S. Shao, Y. Li, H. Yao, Y. He, Z. Qin, and K. Ren, "Explanation as a watermark: Towards harmless and multi-bit model ownership verification via watermarking feature attribution," in *NDSS*, 2025.

[961] W. Xu, K. Gao, H. He, and M. Zhou, "Licoeval: Evaluating llms on license compliance in code generation," *arXiv preprint arXiv:2408.02487*, 2024.

[962] W. Qu, W. Zheng, T. Tao, D. Yin, Y. Jiang, Z. Tian, W. Zou, J. Jia, and J. Zhang, "Provably robust multibit watermarking for ai-generated text," *arXiv preprint arXiv:2401.16820*, 2024.

[963] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, "A watermark for large language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 17 061–17 084.

[964] J. Ye, Y. Wang, Y. Huang, D. Chen, Q. Zhang, N. Moniz, T. Gao, W. Geyer, C. Huang, P.-Y. Chen *et al.*, "Justice or prejudice? quantifying biases in llm-as-a-judge," *arXiv preprint arXiv:2410.02736*, 2024.

[965] Y. Wan, W. Wang, P. He, J. Gu, H. Bai, and M. R. Lyu, "Biasasker: Measuring the bias in conversational ai system," *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258833296

[966] European Union, "Artificial intelligence act," 2024, accessed: 2025-03-07. [Online]. Available: https://artificialintelligenceact.eu/

[967] Cyberspace Administration of China, "Interim measures for the management of generative artificial intelligence services," 2023, accessed: 2025-03-07. [Online]. Available: https://www.cac.gov.cn/2023-07/13/c_1690898327029107.htm

[968] The White House, "Safe, secure, and trustworthy development and use of artificial intelligence," 2023, accessed: 2025-03-07.