大模型对齐攻击综述

宫润森[1,2], 王 凯[1], 张昱霖[1,3], 张伟哲[2,4], 乔延臣[2], 张玉清[1,2,3,5]

[1] 国家计算机网络入侵防范中心 中国科学院大学 中国 北京 100190
[2] 鹏城实验室 中国 深圳 518000
[3] 西安电子科技大学 中国 西安 710048
[4] 哈尔滨工业大学 中国 哈尔滨 150000
[5] 海南大学 中国 海口 570228

摘要 随着像 ChatGPT 这样的大型模型的问世, 人工智能生成内容的安全性引起了越来越多研究者的关注。为了确保模型的最终行为与人类价值观一致, 在模型应用部署过程中, 对齐技术发挥了至关重要的作用。对齐技术通过微调或其他技术手段对不同的预训练模型进行调整, 旨在提高他们在特定任务上的推理能力。针对对齐安全的攻击受到了学术界与工业界的广泛关注, 但目前缺少一个针对大模型对齐攻击技术的系统性梳理。本文首先从部署态大模型的安全风险视角出发, 对大模型整个部署过程中可能存在的安全漏洞和现有的大模型对齐技术进行了调查分析, 对现有对齐攻击手段进行了全面调研, 分析了针对部署态大模型的对齐攻击技术以及存在的安全威胁, 并指出了对齐技术中存在的安全漏洞和潜在攻击手段。其次, 本文从大模型的下游任务微调所带来的安全隐患的视角出发, 分析了微调过程对大模型对齐安全性的破坏, 还调研分析了微调过程中的一些行为可能引发对齐安全的漏洞。然后, 本文从大模型的多模态发展的视角出发, 介绍了多模态大模型(Multimodal Large Language, MLLM)的架构, 充分总结和分析了模型不同模态之间的融合技术, 并指出了由于 MLLM 输入的连续性带来的攻击隐蔽性的特点。最后, 对大模型对齐攻击技术未来的发展方向进行了展望。通过深入探讨对齐攻击技术的现状和潜在风险, 可以为学术界提供启发和新的研究思路与方向。

关键词 大模型对齐; 微调大模型; 人工智能安全; 多模态大模型
中图法分类号 TP18 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2026.03.07

A Survey of Adversarial Techniques Against Large Model Alignment

GONG Runsen[1,2], WANG Kai[1], ZHANG Yulin[1,3], ZHANG Weizhe[2,4], QIAO Yanchen[2], ZHANG Yuqing[1,2,3]

[1] Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100190, China
[2] Pengcheng Laboratory, Shenzhen 518000, China
[3] Xidian University, Xi'an 710048, China
[4] Harbin Institute of Technology, Harbin 150000, China
[5] Hainan University, Haikou 570228, China

Abstract With the advent of large models like ChatGPT, the security of AI-generated content has garnered increasing attention from researchers. To ensure that the final behavior of models aligns with human values, alignment techniques play a crucial role during model deployment. These techniques adjust different pre-trained models through fine-tuning or other methods to enhance their reasoning capabilities on specific tasks. Alignment security attacks have attracted widespread attention from academia and industry, but there is currently a lack of systematic review on alignment attack techniques for large models. This paper begins by examining the security risks faced during the deployment stage of aligned large models. It investigates potential vulnerabilities throughout the deployment process and reviews existing alignment techniques. A comprehensive study of current alignment attack methods is conducted, including prompt injection attacks, adversarial attacks, privacy leakage attacks, and backdoor trigger attacks. The analysis identifies security vulnerabilities and potential attack techniques within alignment methods. Secondly, from the perspective of security risks posed by fine-tuning downstream tasks, the paper analyzes how the fine-tuning process compromises the security limitations of aligned large models. It investigates behaviors during the fine-tuning process that may cause alignment security vulnerabilities, providing a detailed analysis of the impact of fine-tuning on the security of secondarily developed models. Thirdly,

通讯作者: 张玉清, 中国科学院大学国家计算机网络入侵防范中心教授, 博导, Email: zhangyq@ucas.ac.cn。

本项目收到国家重点研发计划项目(No. 2023YFB3106400, No. 2023QY1202), 国家自然科学基金重点项目(No. U2336203, No. U1836210), 海南省重点研发计划项目(No. GHYF2022010), 北京市自然科学基金(No. 4242031), 国家自然科学基金(No. 62102202), 鹏城实验室重大攻关项目(No. PCL2023A05)资助。
收稿日期: 2024-04-19; 修改日期: 2024-09-09; 定稿日期: 2026-01-12

from the perspective of multimodal development of large models, the paper introduces the architecture of multimodal large language models (MLLM). It summarizes and analyzes the fusion technologies between different modalities within these models and highlights the characteristics of attack concealment due to the continuity of MLLM inputs. Finally, the paper provides an outlook on the future development direction of alignment attack techniques for large models. By deeply exploring the current state and potential risks of alignment attack techniques, the research aims to inspire new ideas and directions in academia.

Key words large model alignment; fine-tuning large model; AI security; multimodal large language model

1 引言

随着大模型技术的不断精进, 大模型的能力不断增强, 所涉及的领域范围也在延伸。科技发展的历史告诉我们, 高普适性、强功能性的科技产物在发展、部署、演进的过程中最不容忽视的是其安全性。

大模型中的一些安全风险是与生俱来的, 大模型在训练过程中往往需要大量的互联网数据, 在学习到各类正面数据的同时, AI(人工智能, Artificial Intelligence)模型往往也会受到负面、虚假信息的干扰。因此经过预训练之后的大模型可能会产生不符合人类预期的输出结果, 其中包括但不仅限于虚假、错误、含个人隐私、具有偏见或歧视等信息的内容。另外, 缺乏安全限制的大模型容易被恶意用户利用来产生虚假、歧视等恶意输出。

因此在大模型的开发过程中, 研究人员通常会利用对齐技术对大模型进行安全限制, 使大模型的行为表现与人类的价值观一致。大模型的对齐技术, 通过微调或其他技术手段将不同的预训练模型进行调整, 旨在使它们在特定任务上具有相似的表示能力。大模型对齐保证了大模型在理解人类自然语言的同时也可以做出符合人类预期和道德约束的反应行为。因此针对大模型对齐的相关技术近期也备受瞩目, 例如: 文献[1]中主要关注 LLM(大语言模型, Large Language Model)的对齐工作, 即 LLM 怎样做到和人类的价值观相同, 包括以下步骤: 1) 数据收集, 有效收集用于 LLM 对齐的高质量指令的方法; 2) 训练方法, 包括有监督的微调(Supervised Fine-Tuning, SFT)、基于人类反馈的强化学习(Reinforcement Learning from Human Feedback, RLHF)、ranking-based training signals 及使用语言反馈替代标量奖励等。此外, 还有一些参数高效的训练方法, 旨在减少计算负担, 提高 LLM 对齐的效率; 3) 模型评估, 评估这些以人为本的 LLM 有效性的方法, 对其评估提出了多方面的方法; 文献[2]将 LLM 对齐的主流方法分为: 1) 内部对齐, 目标是确保 AI 系统的“内部”目标(在学习过程中推导或优化的目标)与其设计者设定的“外部”目标相一致; 2) 外部对齐, 选择正确的损失函数或奖励函数, 确保 AI 系统的训练目标与人类价值相匹配, 外部对齐旨在将指定的训练目标与设计者的目标对齐。还探讨了一些突出的问题, 包括模型的可解释性, 以及对对抗性攻击的潜在脆弱性, 并提出了各种各样的基准和评估方法; 文献[3]确定了四个原则作为人工智能对齐的关键目标: 鲁棒性、可解释性、可控性和伦理性。在这四个原则的指导下, 文献[3]概述了当前对齐研究的概况, 并将其分解为两个关键组成部分: 向前对齐(包括反馈学习技术和分布移位学习技术)和向后对齐(包括保证技术和治理实践), 对 AI 的对齐原则、技术以及所面临的问题都分别进行了充分的介绍、分析。

除了对齐技术本身, 我们认为针对对齐技术的攻击与防御也需要得到广泛的关注。我们对近期针对 LLM 安全漏洞攻击的相关综述进行了充分的调查研究, 其中文献[4]探讨了两种攻击类别: 对模型本身的攻击和对模型应用程序的攻击, 包括了 100 多个最近的研究工作, 对每种攻击类型进行了深入分析, 并总结了未来针对这些攻击的防御措施; 文献[5]提出了一个全面的分类, 系统地分析了与 LLM 系统的每个模块(包括输入模块、语言模块、工具链模块和输出模块)相关的潜在风险, 并讨论了相应的缓解策略; 文献[6]调查了 LLM 如何积极影响安全和隐私, 与使用 LLM 相关的潜在风险和威胁, 以及 LLM 内部的固有漏洞, 并提出了一些有趣的发现。例如, LLM 被证明可以增强代码安全性(代码漏洞检测)和数据隐私性(数据机密性保护), 优于传统方法。然而, 由于它们具有类似人类的推理能力, 它们也可以被用于各种攻击(特别是用户级攻击); 文献[7]评估了 LLM 漏洞的程度, 分析了新兴的 LLM 安全和隐私攻击(主要包括提示词攻击、对抗攻击、梯度泄露等), 并审查了潜在的防御机制。此外, 概述了该领域现有的研究差距, 并强调了未来的研究方向。

基于对近期相关综述工作的总结分析(如表 1 所示), 现有综述文献: 1) 讨论了 LLM 对齐技术的不同方法和评估标准, 这些文献不仅探讨了内部和外部对齐的技术细节, 还关注了模型可解释性、对抗攻击的潜在风险以及对齐技术在保持鲁棒性和伦理性方

面的挑战; 2) 分析了 LLM 面临的安全威胁和风险, 这些文献详细总结了针对 LLM 的各种攻击类型, 以及针对这些攻击的防御措施和缓解策略。与现有综述文献不同的是, 本文不仅关注技术本身的发展, 更着重于分析和归纳对齐技术的安全性问题。本文的重点是分析和归纳针对 LLM 和 MLLM 的攻击手段, 其中主要关注了 LM(大模型, Large Model)对齐安全漏洞, 并对其进行了充分的调研与分析。

表 1 现有相关综述文献对比

Table 1 Comparison of existing relevant survey

文献	日期	文献重点	工作分类
[1]	2023.7.24	LLM 的对齐工作: 数据收集, 包括 SFT、RLHF 等训练方法和多方面的对齐有效性模型评估	正向安全能力建设 (LLM 对齐分类、评估)
[2]	2023.9.26	将 LLM 对齐方法分成了内部对齐和外部对齐, 提出了模型的可解释性、对抗攻击的潜在脆弱性等问题, 同时提出了多种基准和评估方法	正向安全能力建设 (LLM 对齐分类、评估)
[4]	2023.12.18	汇总、探讨了针对 LLM 的两种攻击类别: 模型本身的攻击(包括恶意提示攻击和隐私泄露攻击等)和模型应用程序的攻击(包括误导欺骗、舆论操纵等), 并总结了针对这些攻击的防御措施	反向安全风险揭示 (LLM 模型攻击、应用攻击)
[5]	2024.1.11	提出了一个系统的分类, 分析了包括输入模块、语言模块、工具链模块和输出模块的 LLM 系统每个模块的潜在风险和缓解策略	正向安全能力建设 (LLM 模块化分析、模型系统分析)
[6]	2024.1.26	调查了 LLM 对安全和隐私的影响, 以及 LLN 内部的固有漏洞	反向安全风险揭示 (LLM 应用中的正负面影响、LLM 内部漏洞)
[7]	2024.1.30	评估了 LLM 漏洞的程度, 分析了主要包括提示词攻击、对抗攻击、梯度泄露等新兴的 LLM 安全攻击, 并审查了潜在的防御机制	反向安全风险揭示 (LLM 安全攻击)
[3]	2024.2.26	确定了人工智能对齐的鲁棒性、可解释性、可控性和伦理性四个关键目标, 将当前对齐技术分为向前对齐和向后对齐, 介绍对齐原则、技术和面临的挑战	正向安全能力建设 (LLM 对齐技术、对齐标准评估)
本文	2024.3	对提示词攻击、对抗攻击等现有对齐攻击工作做了充分的调研, 尤其关注了微调过程对模型对齐安全性的威胁, 介绍并分析了 MLLM 的技术架构及其所带来的新的安全漏洞	反向安全风险揭示 (LLM 对齐攻击, MLLM 安全威胁)

通过对近期相关工作的综合总结和分析, 本文致力于深入理解对齐技术的安全限制, 并探讨其可能存在的大模型对齐攻击, 即主要发生在模型的部署阶段及针对下游任务的微调阶段对大模型对齐安全性的破坏性攻击。特别地, 我们关注了 MLLM 的安全性, 指出了相较于纯语言 LM 的优势和挑战。本文的贡献在于:

1) 从大模型对齐之后的部署态面临的安全风险视角出发, 分析了大模型的对齐攻击技术以及存在的安全威胁。我们对提示词攻击、对抗攻击、隐私泄露攻击、后门触发攻击等现有对齐攻击手段进行全面调研和分析, 并指出了对齐技术中存在的安全漏洞和潜在攻击手段, 其中主要包括直接注入恶意提示词、精心设计扰动输入数据以及利用模型推理过程中的梯度信息等方法。

2) 从大模型的下游任务微调所带来的安全隐患的视角出发, 分析了微调过程对大模型对齐安全限制的破坏。我们对大模型隐私泄露攻击、RLHF 保护移除攻击等对齐攻击技术进行了详细的介绍, 并分析了微调过程对二次开发态模型的安全影响: 攻击者可以通过微调行为中的恶意信息唤醒模型对于一些隐私数据的记忆; 攻击者通过设置恶意数据破坏模型的安全限制; 完全良性的微调数据也会导致模型的安全性下降等。

3) 从大模型的多模态发展的视角出发, 我们首先充分地介绍了 BLIP2、CLIP 等 MLLM 的架构, 充分总结和分析了模型不同模态之间的融合技术。然后我们对 MLLM 的现有攻击进行了调研分析, 最终我们对比了其与纯语言大模型的不同之处, 并指出了由于 MLLM 输入的连续性带来攻击隐蔽性的特点。

2 大模型基础

2.1 大模型生命周期

为了更好地理解大模型针对大模型对齐的攻击, 首先我们将在本节中介绍大模型的部署流程, 以及大模型整个生命周期中所涉及的 10 大安全漏洞[8]。与传统的端到端模型不同, 大模型的研发和部署过程可以概括为开发态、二次开发态和部署态三个(如图 1 所示)。为保证大模型的安全性, 开发态中通常会将预训练后的大模型进行对齐。然后在二次开发态中通过特定领域的数据集对经过了大量数据集训

练后的已对齐预训练大模型进行微调, 大模型可以更好地适配特定的下游任务。对于预训练结合微调的部署模式, 大模型的整个部署过程中不可避免地存在若干漏洞[8], 下面我们依次介绍。

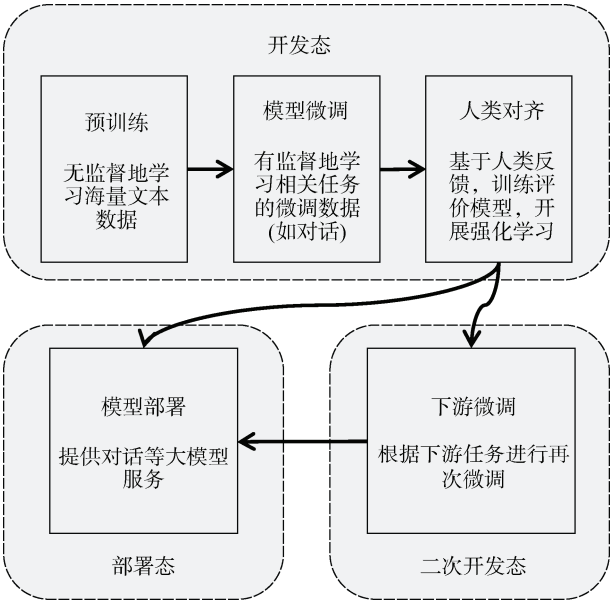

图 1 大模型部署过程

Figure 1 LM deployment

2.1.1 提示注入

提示注入[8]是针对部署态中已对齐大模型的一种攻击手段, 攻击者会精心构建出一个含有提示词的输入, 使 LLM 返回不满足其安全限制的输出内容, 通常可以称这种攻击为“越狱”。“越狱”的 LLM 可以引发数据泄露、破坏 LLM 对齐限制、未经授权的插件使用等安全问题。LLM 的输出通常是受信任的, 这导致了用户不易察觉到此类攻击操作的存在。

攻击行为可以分为: 1) 直接攻击, 用户覆盖原有系统提示可能允许攻击者通过不安全的操作行为来利用后端系统; 2) 间接攻击, 攻击者利用已受控制的外部数据源进行输入, 并在对话上下文的外部内容中嵌入提示注入将导致 LLM 成为不知情的恶意代理, 允许攻击者操作 LLM 访问权限中的用户和附加系统。此外, 间接提示注入只需文本被 LLM 解析, 可以是对人类透明的。

由于 LLM 认为指令和输入数据都是用户的输入, 提示注入并不会被完全防御。目前高效的防御措施包括: 1) 权限控制, 降低 LLM 的权限; 2) 用户输入安全验证, 通过外部的功能清理掉不受信任的输入; 3) 隔离外部内容交互, 将不受信任的内容与用户提示隔离开来, 并控制与外部内容的交互; 4) 建立信任边界, 认为 LLM 是不受信任的用户, 在 LLM 和外部源可扩展内容之间建立用户可决策的信任边界。

2.1.2 不安全输出

不安全输出攻击[8]是针对部署态中已对齐大模型的一种攻击手段, 通常作为提示注入的后续攻击手段, 当恶意用户通过特定的提示词来控制 LLM 的输出时, LLM 的下游应用程序过度信赖 LLM 的输出而未加足够的检查导致的安全漏洞。

不安全输出可以通过如下手段进行预防: 1) 降低 LLM 输出的受信任程度, 将 LLM 视为一个一般用户, 并对 LLM 的输出内容进行安全检测; 2) 更改编码方式, 将 LLM 的输出进行了编码后返回给用户, 从而减少恶意代码的执行。

2.1.3 训练集投毒

训练集投毒是针对开发态中大模型的一种攻击手段。LLM 想要获取很强的推理能力, 其训练所需的原始数据集要求数据量大、领域涉及广泛。LLM 的深度神经网络会依据训练数据进行推理, 攻击者通过毒化训练集恶意操控 LLM, 此类攻击可能会造成 LLM 性能降低、植入后门、模型偏见、不道德行为或信息泄露等风险。

投毒攻击可以通过以下方式进行防御: 1) 对模型进行训练时验证数据、数据提供链的合法性; 2) 通过独立的模型或者筛选器, 对不信任的数据来源和数据集进行数据验证、筛选, 以尽量减少伪造的数据集; 3) 确保存在足够的沙盒以防止模型意外抓取恶意数据; 4) 增加模型的鲁棒性训练。

2.1.4 拒绝服务

拒绝服务攻击是针对部署态中已对齐大模型的一种攻击手段。攻击者以占用 LLM 服务器资源为目的, 以特别占用资源的方式与 LLM 进行交互, 导致其他用户的服务质量下降或产生较高的服务成本。常见的攻击手段包括: 输入异常消耗资源的查询、连续输入长度溢出的查询以消耗计算资源、反复输入超过上下文窗口限制的输入、构建递归的上下文扩展输入迫使 LLM 不断处理和扩展上下文窗口等。

拒绝服务攻击的防御手段有: 1) 限制每个用户请求的资源分配; 2) 限制 LLM 响应操作队列数量和总数量; 3) 验证并筛选出恶意输入; 4) 持续监控 LLM 的资源使用情况; 5) 对 LLM 上下文窗口长度进行严格的限制。

2.1.5 供应链漏洞

供应链漏洞攻击是针对开发态中大模型的一种攻击手段。由于安全漏洞和系统性故障, LLM 的供应链容易受到攻击。漏洞主要来源于预训练的模型、外部扩展和数据, 从而影响训练的效率和训练结果

的稳定性，甚至导致模型偏见等一系列安全问题。

防御措施: 1) 增强数据和外部扩展来源和供应商的审查; 2) 从 LLM 的开发和部署全程对 LLM 系统组件进行漏洞扫描; 3) 对整个 LLM 供应链条进行鲁棒性测试; 4) 检查供应商安全。

2.1.6 敏感信息泄露

此类攻击是针对部署态中已对齐大模型的一种攻击手段, 指的是恶意用户通过攻击手段诱导 LLM 输出不满足模型对齐安全限制的信息。LLM 预训练模型的训练数据中包含大量、广泛的敏感、隐私信息, 恶意用户可能会利用提示注入等安全漏洞诱导 LLM 输出带有涉及敏感或未经授权访问的信息。不仅如此, 这些隐私信息还有可能涉及知识产权等问题。

目前可以通过以下策略来应对此类安全漏洞: 1) LLM 应用程序应当保有适当的数据清理能力或隐私保护条款来防止恶意用户对训练数据集的完全访问权限; 2) 为 LLM 应用程序设置恶意输入筛选器, 阻止用户的恶意输入; 3) 为用户和 LLM 之间设置双向的信任边界; 4) 为不同的数据和用户设置不同的访问权限; 5) 对外部数据设置严格的访问限制。

2.1.7 不安全的插件设计

此类攻击是针对部署态中已对齐大模型的一种攻击手段。LLM 应用程序在与用户进行交互时会自驱动调用扩展插件, 插件执行过程中不被应用程序控制, 尤其插件可能会在未经检查的情况下实现来自模型的自由文本输入, 这将允许用户构造出一个攻击插件的恶意文本, 可能会导致包括恶意代码执行等一系列恶意行为。

这种攻击的预防方式主要分为两种: 1) 插件设计规范, 插件应该规范参数设计并严格要求参数输入, 若因程序要求, 输入必须为自由格式, 则必须做好筛查工作; 2) 完善 LLM 应用程序对外部插件的检查验证工作, 确保插件的安全性能, 同时通过身份验证标识等方式为插件设置不同的信任程度。

2.1.8 过度代理

此类攻击是针对部署态中已对齐大模型的一种攻击手段。代理指开发人员将 LLM 与其他系统交互并采取行动、执行函数等决策行为交给 LLM。过度代理漏洞允许其他外部系统或扩展执行 LLM 任何意外或者模糊的输出命令, 即使这些命令是攻击者通过提示注入等攻击手段产生的恶意指令, 这可能是破坏性的。过度代理漏洞的威胁程度取决于 LLM 的应用程序可以与哪些外部功能进行交互。

防范方式: 1) 尽量减少 LLM 被授予的代理权限; 2) 尽量减少外部插件系统所能执行的功能; 3) 尽可能避免直接操作 shell 等开放性功能; 4) 加入人机交互控制, 操作要通过人工批准授权。

2.1.9 过度依赖

此类攻击是针对部署态中已对齐大模型的一种攻击手段。LLM 固然可以生成富有创造性的内容。但 LLM 应用程序所输出的内容可能会有偏见、虚构、违背人类价值观等错误信息。因此需要用户使用 LLM 时进行筛查、鉴别。

防御措施: 1) 定期检查 LLM 的合规性输出; 2) 为 LLM 设置可靠的交叉验证; 3) 通过微调等手段, 完善模型的输出准确性; 4) 建立可操作的风险沟通, 并持续衡量风险沟通的有效性。

2.1.10 模型盗取

此类攻击是针对部署态中已对齐大模型的一种攻击手段。模型盗取可以描述为专有 LLM 遭到破坏、参数复制、权重提取等情况, 这可能会造成严重的安全问题, 包括但不仅限于经济损失、敏感信息泄露等问题。这类漏洞需要 LLM 的组织人员使用强大的安全措施来保护他们, 确保 LLM 机密性和完整性。

2.1.11 小结

综合以上描述, 不难看出 LLM 在使用部署过程中依然存在很多安全隐患。除 2.1.4 节拒绝服务攻击外, 其余上述所有大模型安全漏洞均可直接或间接对大模型的对齐安全产生威胁。其中提示注入以及训练数据投毒攻击可以作为恶意用户对 LLM 进行攻击的“跳板”漏洞。随着 LLM 技术的不断完善, 模型对齐技术可以在大部分场景下保证模型的输出内容的安全性, 即与人类价值观的一致性。提示注入攻击也变得越来越困难。除此之外, 对模型的训练集进行投毒攻击的成本非常高。如 2.1.3 节所述, 当模型有效数据集足够大时, 可以有效地降低数据投毒的攻击成功率。如若数据投毒发生在模型的微调阶段, 便可以通过极小的数据毒化率达到攻击效果, 尤其对于 MLLM。图片、音频等多种模态的连续性, 为数据毒化提供了更隐秘的条件, 模型对齐的安全性限制也更容易被打破。因此本文中尤其注重提示注入、不安全输出、训练集投毒、敏感信息泄露等大模型的微调、部署阶段中破坏其对齐安全性的攻击。

如图 2 所示, 本文从大模型的不同部署过程角度出发, 将现有的大语言模型对齐攻击划分为两大类别: 部署态大模型面临的攻击威胁和下游任务微调带来的安全威胁。同时本文关注到 MLLM 所带来的新安全威胁。

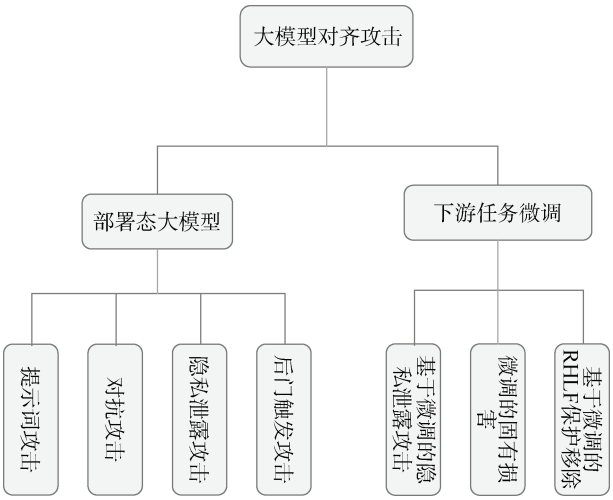

图 2 大模型对齐攻击分类

Figure 2 Large model alignment attacks classification

2.2 对齐技术

在探讨大模型对齐安全在部署、应用阶段所面临的攻击威胁之前, 我们应熟知大模型对齐技术的发展现状。谈及对齐技术, 我们应先了解对齐的关键意义所在。人工智能发展初心即为创造出可以像人类一样思考和行动的智能体, 如此智能体如何能够保持与人类相同的价值观成为发展人工智能的关键所在。文献[9]总结出 AI 对齐的宏观目标为 RICE 原则: 1) 鲁棒性(Robustness), 模型在多样化的应用场景中应保证推理结果的正确性, 在各种对抗和恶意攻击中保持应用的高效性和正确性; 2) 可解释性(Interpretability)要求人类能够理解 AI 推理过程及工作原理; 3) 可控性(Controllability)保证了 AI 系统应用始终在人类的掌控和约束中, 支持人类始终可以对 AI 系统的偏差和漏洞进行调整; 4) 道德性(Ethicality)指的是 AI 系统在任何任务中所表现出的行为和决策都要顺应人类的价值观。

LLM 为目前 AI 领域中最新的高性能人工智能系统, 大模型的对齐也可以视作为 AI 对齐技术与大模型的交叉应用。我们将在本章的后续小节中介绍现有的 LLM 对齐技术。

2.2.1 基于强化学习的方法

基于强化学习的方法目前主要是基于人类反馈的强化学习, RLHF[10]是目前最常用的非递归监督方法, 通过强化学习来优化 LLM, 通过人类意愿来表达人类价值观, 并训练适配人类偏好的奖励模型。近期出现了大量的 RLHF 相关的工作[13], 其本质上包含以下三个步骤, 1) 收集人类反馈数据; 2) 通过人类反馈数据训练奖励模型; 3) 使用奖励模型对 LLM 进行强化学习的微调。

值得注意的是, 以人类反馈的标准作为模型的奖励对模型进行微调时如果不加控制可能会导致模型为了追求高奖励而偏离初始推理机制而输出错误信息。因此通常会设置模型的输出与未经过 RLHF 的模型输出进行对比, 并设置惩罚项加入到奖励机制中。

2.2.2 基于有监督学习的方法

基于人类反馈的方法虽然已经可以很好地让大模型理解人们的偏好, 但是其训练过程中还是存在奖励系统偏差、优化过程不稳定等不足。因此需要基于有监督学习的方法[19](Supervised Learning, SL)。SL 可以大致分为两类: 1) 基于文本的反馈信号[21], 将人的意图和偏好转换为基于文本的反馈信号来实现对齐, 这一过程可以简单地理解为一种特殊的微调过程; 2) 基于排名的反馈信号[25], 直接使用监督学习来优化 LLM, 并使用基于排名的反馈信号构建损失函数。

2.2.3 任务分解

其主要目的是利用较弱的监督者, 可以利用简单的信息来判断复杂的任务。任务分解可以理解为当人类面对一件很困难的事情, 通常会将其分解成为一系列易于解决的子问题。例如: 文献[32]将过程监督分割成一系列连续的子任务, 每个任务都有自己的依赖关系, 关键在于为不同阶段设置独立的监控信号, 等同于在整个训练阶段提供密集的奖励, 这可以潜在地减轻仅根据困难任务的最终结果估计稀疏奖励的挑战。

2.2.4 辩论

辩论[33]的过程可以简单理解为一场辩论赛, 由一个或多个智能体对一个问题输出一个答案, 然后依次作为参与者对答案进行评论, 最终人类从所有的论点中选择一个最合适的答案。其好处是人类不必面对单个输出进行判别, 辩论的过程使模型的输出变得合理、可解释, 同时还可以看到简单的结果推理过程。

2.2.5 小结

对齐技术扮演着关键角色, 其目标是确保人工智能系统的行为和决策与人类价值观一致。文献[9]提出了 AI 对齐的 RICE 原则, 包括鲁棒性、可解释性、可控性和道德性, 为后续研究奠定了理论基础。在具体方法方面, 基于强化学习的方法如 RLHF[10-18]通过人类反馈数据训练奖励模型, 优化模型在多样化应用场景中的推理能力, 但需注意控制奖励机制以避免推理路径的偏离。相较之下, 基于有监督学习的方法[19-31]则直接利用文本或排名反馈信号调整模型

输出, 尽管面临奖励系统偏差和优化不稳定的挑战。此外, 任务分解和辩论[32-34]作为补充方法, 通过简化复杂任务或模型输出的辩论过程来提高模型的合理性和可解释性。上述相关工作对比分析如表 2 所示。

表 2 对齐技术工作概述

Table 2 An overview of alignment technology work

相关工作	工作核心	潜在威胁
[9]	AI 行为和决策上与人类价值观保持一致	破坏此类对齐目标(如对抗性攻击、隐私侵犯和行为误导等)
[10-18]	通过人类反馈来优化模型	容易受到对抗性攻击的影响(恶意数据)
[19-31]	利用文本或排名反馈信号调整模型输出	容易受到数据泄露和隐私侵犯的威胁
[32-34]	任务分解和辩论作为对齐技术的补充	篡改分解任务或干扰辩论过程来误导模型输出

2.3 微调技术

微调阶段在大模型的部署中起到至关重要的作用, 大模型的微调行为可能会对其安全性产生至关重要的影响, 甚至微调过程中的恶意行为可能会对破坏大模型的对齐安全限制。因此在介绍微调行为的安全威胁之前, 我们首先对现有的微调技术进行了简单的对比。对预训练大模型进行微调: 1) 可以让每个使用大模型的组织避免从头训练自己的大模型, 减少对大模型的使用成本; 2) 可以让大模型更好地适配不同领域的下游任务, 大模型持有者可以通过特定领域中的自有数据有效地提升大模型在该领域中的表现性能; 3) 让大模型提供个性化服务成为可能, 例如依据不同的用户数据微调出适配的轻量级大模型; 4) 当自有数据设计用户隐私、机密等安全问题时, 避免将数据发送给第三方大模型服务机构是必要的, 对模型进行微调就是一个很好的解决方法。

大模型的微调技术可以分为全参数微调(Full Fine Tuning, FFT)和部分参数微调(Parameter Efficient Fine Tuning, PEFT)。其中 FFT 指的是用特定的数据训练大模型, 进而改变大模型的权重, 可以有效地提升大模型在特定领域中的性能表现。但 FFT 存在两个主要的缺陷: 1) 训练参数多, 训练成本高; 2) 灾难性遗忘, 微调后的模型提升了特定领域内的推理性能的同时会降低模型在其他领域内的推理性能。因此现在主流的微调方式是部分参数微调。我们将在本章后续小节中对现有的主流微调技术进行简单介绍。

2.3.1 Adapter Tuning

文献[35]中提出了针对 BERT 的 PEFT 方式。文章中指出, FFT 的训练效率过低, 而如果只改变神经网络中接近下游任务的几个神经层, 很难达到好的效果。因此文章中介绍了称为 adapter 的模块, 针对不同的下游任务在语言模型的每层 transformer 中增加的两个带有少量参数的模块(如图 3(a)所示)。在微调阶段的训练中, 固定原本的参数, 只需训练新增的适配器即可将语言模型迁移到特定的下游任务中, 在保证了模型在下游任务中推理性能的同时, 又提高了训练效率。适配器内部结构如图 3(b)所示, 首先通过 down-project 层将输入数据进行降维映射, 将低维度的数据经过一个非线性层的计算, 然后将计算结果通过 up-project 恢复为原始维度, 最终为了保证 adapter 的输出不会太差, adapter 通过 skip-connection 结构保证其最坏情况也会退回与最初一致。Adapter 的整体设计思路是减少微调过程中所需要的训练参数, 同时保证将微调训练过程贯穿在神经网络的每个 transformer 层中。

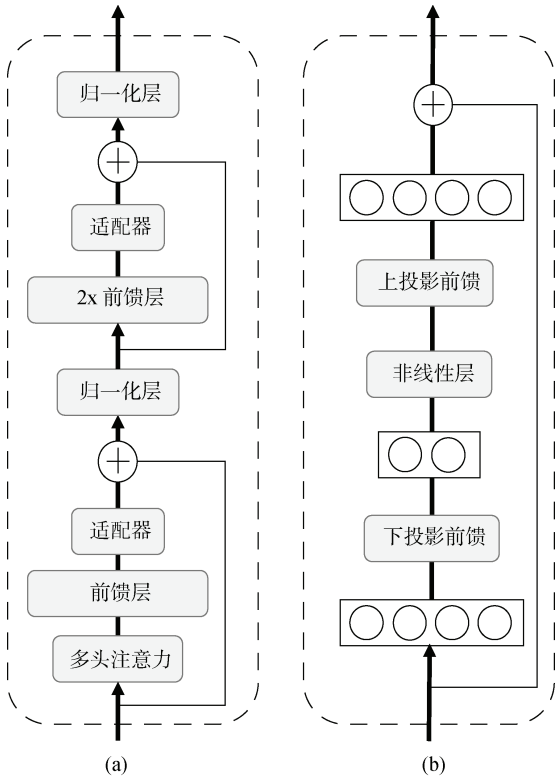

图 3 适配器结构[35]

Figure 3 Structure of Adapter

2.3.2 Pattern-Exploiting Training

Pattern-Exploiting Training(PET)的主要思想是借助自然语言构成模板, 并将不同的下游任务视为句子补全的预测任务, 这样 BERT 的预训练模型就可以通过条件前缀来进行预测推理, 从而可以实现例如情感分类等预测任务。PET 不仅适用于基于掩盖的语言模型(Masked Language Model, MLM), 也

适用于类似于 GPT 的单向语言模型。由于其解码方式为从左至右, 因此预测部分置于文本后缀。对于大部分的下游任务模板的自动化构建是一个值得研究的问题, 模板的构建通常为人工设计或自动化搜索生成的离散模板。

2.3.3 Prefix Tuning

Prefix Tuning[36]理解为 PET 的一种, 在此之前的任务中所使用的基本都是离散模板, 这种离散模板的鲁棒性很低。Prefix Tuning 中提出使用连续的 virtual token embedding 替代了传统的离散模板。前缀微调是在输入 token 之前构造一段任务相关的虚拟 tokens 作为前缀(不只是 transformer 的输入层, 且每一层 transformer 的输入不是从上一层输出, 而是随机初始化的 embedding 作为输入), 都在真实的句子表征前面插入若干个连续的可训练的虚拟 token 嵌入表示, 这些虚拟 token 不必是词表中真实的词, 而只是若干个可调的自由参数。

训练的时候只更新前缀部分的参数, 而 Transformer 中的预训练参数固定。对于上述这个过程, 有以下几点值得注意。该方法其实和构造 Prompt 类似, 只是 Prompt 是人为构造的“显式”的提示, 并且无法更新参数, 而前缀则是可以学习的“隐式”的提示。同时, 为了防止直接更新前缀的参数导致训练不稳定的情况, 特在前缀层前面加了 MLP 结构(相当于将前缀分解为更小维度的 Input 与 MLP 的组合后输出的结果), 训练完成后, 只保留 Prefix 的参数。prefix-prompt[36]的效果优于 adapter tuning 和 fine-tune 最上面的两层, 最终和全参数 finetune 差不多, 且在低资源情况下, 效果优于 finetune。更长的前缀意味着更多的可微调参数, 所带来的效果也变好。

2.3.4 P-Tuning V1/V2

P-Tuning 的设计主旨是为了解决离散模板的不稳定性、低鲁棒性, 自动化搜索的高成本等一系列模板构建问题。

(1) P-Tuning V1

P-Tuning[37]设计了一种连续的、可学习的虚拟 token, 与 Prefix Tuning 的主旨类似, 将模板的构建问题转化成为一种连续参数的优化问题。与 Prefix Tuning 不同的是, P-Tuning 没有将虚拟 token 添加到模型的每一层, 而仅添加到了输入层, 同时不将 token 固定在模型的开头作为输入前缀, 而是可选择的插入位置。另外, 对于虚拟 token 的编码方式上, P-Tuning 没有选择随机初始化编码, 因为对于高度离散化的词嵌入表示, 随机编码的虚拟 token 容易优化到一个局部最优解。P-Tuning 认为, 插入的虚拟 token 应该具有相关性关联, 因此 P-Tuning 使用一个 prompt 编码器对虚拟 token 进行编码会收敛得更快, 结果更优。其中 prompt 编码器是由长短期记忆网络(Long-short Term Memory, LSTM)和多层感知机(Multi-layer perceptron)所组成是。P-Tuning 大部分任务中可以达到与全参数微调一致的结果, 甚至在部分任务中表现出比全参微调更好的结果。值得一提的是, 相同参数规模的情况, 对 GPT 进行全参数微调, 其在自然语言理解(Natural Language Understanding, NLU)任务中的效果变现远远低于 BERT, 而在利用 P-Tuning 进行微调下, GPT 可以在 NLU 任务中达到比 BERT 更好的效果。

(2) P-Tuning V2

P-Tuning V2[38]主要针对 V1 中的两个问题进行了更改: 1) 缺乏参数规模和任务的通用性, 主要表现在, 当模型参数规模超过 100 亿时, P-Tuning 的结果可以达到全参数微调的效果, 但当模型参数规模较小时(如 100M-1B), P-Tuning 与全参数微调有很大的差别。另外, P-Tuning 并不是对所有任务都有较为优异的表现, 例如阅读理解等较难的 token 级的任务中表现得就不尽如人意; 2) 缺少深度提示优化, P-Tuning 中的虚拟 token 只被插入到输入层, 而之后的 transform 层都是由前面计算出来的结果。

针对以上两个问题, P-Tuning V2 在模型的每一层都加入了 token。通过增加了大量的可学习的 prompt 参数, 使得模型深层结构也可以通过 prompt 更为直接地影响模型的预测结果, 其中具体做法同 Prefix Tuning 相似, 主要有以下的改进: 1) 原本工作中基本都是通过重参数化来提高训练速度和模型鲁棒性(例如 Prefix 中的 MLP, P-Tuning 中的 LSTM), 但对于较小的模型, 重参数化会影响到模型的效果表现; 2) 对于不同的任务, 设置不同长度的 prompt, 对于不同的文本生成任务通常需要设定不同长度的 prompt 才会表现出最佳的性能; 3) 支持多任务学习, 这是 P-Tuning V2 中新增的可选项, 可以对训练过程起到有益的补充; 4) 使用传统的分类标签范式, 采用随机初始化的分类头, 可以增加通用性。

2.3.5 LoRA

模型参数通常是以矩阵的形式进行理解、计算, 对预训练模型进行微调实际上可以理解为针对特定任务修改模型的参数矩阵。而 LoRA[39](Low-Rank Adaptation)的核心思想是使用一种低秩的方式来调整模型的参数矩阵。低秩意味了一个矩阵可以用两个较小的矩阵相乘来近似表达。LoRA 的具体流程为: 1) 选择预训练神经网络中的目标训练层, 通常与任务相关(原则上, LoRA 可以应用于模型权重矩阵中的

任何子集); 2) 为目标层创建两个较小的矩阵: 映射矩阵和逆映射矩阵; 3) 参数变换, 更新参数矩阵为原参数矩阵与两个小矩阵乘积的和; 4) 模型微调, 使用新的参数矩阵替换目标层中原始参数矩阵, 然后对模型进行微调。注意, 微调过程中, 固定原始参数矩阵不变, 只训练映射矩阵和逆映射矩阵; 5) 重复训练过程, 直到满足训练批次或者收敛。

2.3.6 QLoRA

QLoRA[40](Quantized Low-Rank Adaptation)进一步降低了微调过程的显存消耗, 主要通过以下两种技术: 1) 4 位的 NormalFloat 量化通过估计输入张量的分位数来保证每个量化区间分配相等的值; 2) 双量化, 将额外的量化常数进行量化以减小内存开销的过程。

2.3.7 小结

微调技术在大型语言模型的部署中具有关键作用, 能够显著提升模型在特定领域的性能并降低训练成本。全参数微调(FFT)虽然有效, 但存在训练成本高和灾难性遗忘的缺陷, 因此 PEFT 成为主流选择。主要的 PEFT 方法包括 Adapter Tuning、Pattern-Exploiting Training(PET)、Prefix Tuning、P-Tuning V1/V2、LoRA 和 QLoRA。Adapter Tuning 通过增加适配器模块来提升训练效率和性能; PET 利用自然语言模板将下游任务转化为句子补全任务; Prefix Tuning 使用连续的虚拟 token embedding 作为前缀; P-Tuning V1 解决了离散模板的鲁棒性问题, P-Tuning V2 进一步改进了参数规模和任务通用性; LoRA 通过低秩分解调整参数矩阵, 而 QLoRA 通过量化技术进一步降低显存消耗。这些技术为使得大型语言模型能够更好地适应不同领域和任务, 共同致力于提高微调效率和模型性能。但是微调行为对模型的性能带来提升的同时也对模型的对齐安全性带来了新的挑战, 我们将在第 4 节对大模型微调带来的安全威胁进行讨论。

3 部署态大模型面临的攻击威胁

在 2.1 节, 我们已经介绍了大模型在整个生命周期中所面临的安全威胁, 部署态大模型所面临的安全威胁亦基于此。本节将更为详细地介绍已对齐大模型在部署态面临的四大安全威胁: 1) 提示词攻击; 2) 对抗攻击; 3) 隐私泄露攻击; 4) 后门触发攻击。

3.1 提示词攻击

提示词攻击指的是攻击者通过精心构建大的含有提示词的输入, 使 LLM 返回不满足其安全限制的输出内容。

提示词攻击[41]分为直接提示词注入攻击和间接提示词注入攻击, 其中直接提示词注入的主旨是用户构建恶意引导性输入攻击, 从而绕过 LLM 对齐阶段的安全性防护, 输出训练数据中涉及的隐私信息或敏感数据等。间接提示词注入将恶意指令或代码嵌入到 LLM 输入文件数据或外部检索内容中, 达到在LLM执行过程中隐秘操控LLM应用, 从而达到隐私信息窃取的攻击目的。

文献[42]的攻击思路是利用间接提示注入(Indirect Prompt Injection)来攻击集成了大型语言模型(LLM)的应用程序。攻击者可以通过注入恶意提示来远程影响其他用户的系统, 间接控制模型的行为。攻击手段包括诱导用户泄露个人数据、传播欺诈信息、入侵系统基础设施、传播恶意软件、操纵内容和发起可用性攻击。恶意用户攻击的目的可能包括获取用户数据、实施欺诈行为、入侵系统、传播恶意软件、操纵信息内容和破坏服务可用性等。

文献[43]的攻击思路是通过提示词注入攻击来欺骗 LLM, 使其将恶意的提示词解释为一个问题, 而不是正常的数据负载。攻击手段是利用黑盒提示词注入攻击方法 HOU YI, 通过三个关键组件(框架组件、分隔组件和干扰组件)构建注入的提示词。攻击的目的是窃取原始服务的提示词, 模仿服务并免费利用 LLM 的计算能力。攻击的结果是可以对服务提供商造成数百万美元的财务损失, 并影响数百万用户。

文献[44]的攻击思路是通过 Prompt Automatic Iterative Refinement(PAIR)算法来生成语义级别的攻击, 即通过两个黑盒大语言模型(LLM), 即攻击者模型(A)和目标模型(T), 相互协作地发现可能导致目标模型生成不良内容的提示词。攻击手段是通过迭代的方式, 攻击者生成候选提示词, 将其输入目标模型, 获取响应, 然后通过评分函数进行评估, 如果前一个提示词和响应没有被分类为攻击, 则将提示词、响应和评分传回给攻击者, 生成新的提示词。攻击的目的是发现可以生成有问题内容的提示词, 攻击结果是在直接攻击和迁移攻击方面取得了较高的攻击成功率。

文献[45]提出一个名为“MASTER KEY”的自动化破解大型语言模型聊天机器人的框架。文章指出, 由于大型语言模型(LLM)具有理解、生成和完成类似人类文本的出色能力, 因此 LLM 聊天机器人成为非常受欢迎的应用。然而, 这些聊天机器人容易受到破解攻击的影响, 恶意用户可以通过操纵提示来

揭示敏感、专有或有害信息, 违反使用政策。虽然已经进行了一系列的破解尝试来揭示这些漏洞, 但本文的实证研究表明现有方法对主流 LLM 聊天机器人并不有效。原因似乎是服务提供商采用了未公开的防御措施来对抗破解攻击。为了探索破解攻击和防御背后的机制, 作者提出了 MASTER KEY 框架。该框架首先提出了一种创新的方法, 利用生成过程中固有的时间特性来逆向工程主流 LLM 聊天机器人服务的防御策略。通过操纵聊天机器人的时间敏感响应, 可以了解其实现的复杂性, 并创建一个绕过多 LLM 聊天机器人(如 CHAT GPT、Bard 和 Bing Chat)防御的概念验证攻击。其次, 作者提出了一种自动生成针对受保护的 LLM 聊天机器人的破解提示的方法。该方法的核心是使用 LLM 自动学习有效模式, 通过使用破解提示对 LLM 进行微调。

文献[41]于 2021 年讨论了直接提示词注入和间接提示词注入攻击, 旨在绕过 LLM 的安全性防护, 可能导致训练数据的泄露和隐私信息的窃取。文献[42]发表于 2023 年, 通过间接提示词注入恶意提示来影响 LLM 集成的应用程序, 进而操控模型行为, 可能导致用户数据泄露、欺诈行为等安全问题。文献[43]于 2023 年, 提出了黑盒提示词注入攻击方法 HOUYI。该方法通过构建关键组件来欺骗 LLM, 使其生成意外内容, 可能导致服务提供商财务损失和用户影响。文献[44]于 2023 年发表, 介绍了 PAIR 算法生成语义级别的攻击。其目的是发现并利用可以导致目标 LLM 生成不良内容的提示词, 攻击在直接攻击和迁移攻击方面表现出较高的成功率。文献[45]同样发表于 2023 年, 提出了 MASTER KEY 框架, 旨在自动化破解多个大型语言模型聊天机器人的防御策略。

上述工作反映了对 LLM 提示词注入攻击的不断演进和深入探索。随着 LLM 技术的普及和应用场景的扩展, 未来的研究将需要更加关注防御机制的开发和改进, 以应对日益复杂和多样化的攻击威胁。展望未来, 应重点关注自动化防御系统的发展, 如基于机器学习的入侵检测和行为分析, 以及加强 LLM 模型的安全训练和部署实践, 以保护用户隐私和系统安全。

3.2 对抗攻击

对抗攻击指的是通过对输入数据添加的扰动, 来影响模型的推理结果。

对抗攻击[46]的设计思想为: 对输入数据进行精心设计的扰动来破坏学习模型的推理结果和推理性能, 为了使数据的扰动对人类透明同时尽量减少不必要的操作, 对抗攻击对输入数据的扰动应当尽量保证小, 最终模型会对被扰动的样本表现出与原本数据不同的推理结果。

文献[47]提出了一种简单有效的对齐语言模型的对抗攻击方法, 通过找到一个后缀, 将其附加到多种查询中, 使模型生成不良行为。该方法通过贪心和梯度搜索技术自动产生对抗后缀, 而不是依赖于手动工程。此外, 该方法还能够高度迁移, 包括对黑盒、公开发布的生产 LLM 的攻击。

上述研究不仅拓展了对抗攻击的理论和实践, 还为未来改进和应对深度学习模型安全性提供了新的思路。未来可以进一步研究如何提高对抗攻击的效率和迁移性, 同时发展更复杂和智能化的防御机制来保护深度学习模型免受此类攻击的影响。

3.3 隐私泄露攻击

大模型对齐通常会禁止大模型在对话过程中泄露训练数据中的个人隐私风险。因此有专门针对大模型的隐私泄露攻击。模型的隐私攻击指攻击者利用模型漏洞窃取模型训练时的隐私数据。可以对比针对普遍的语言模型的隐私攻击, 目前可以分为如下四种: 1) 梯度重构攻击, 发生在模型的分布式训练阶段; 2) 属性推理攻击; 3) 推理阶段的反转攻击; 4) 提示攻击。

3.3.1 梯度重构攻击

梯度重构攻击[48]利用模型的分布式训练过程中节点之间的信息通信, 通过获取模型梯度等训练信息重新构建模型训练中的信息。攻击者可以窃取不同节点之间交换的训练信息, 包括训练数据、模型梯度等敏感数据。最终攻击者通过所窃取到的敏感数据重建模型训练时的隐私数据的详细内容。这种攻击虽然不是直接针对已对齐的大语言模型, 但是在对对齐大模型进行分布式微调时, 依然会面对此类威胁, 因此在对齐大语言模型的安全性讨论中, 此类攻击也不应该被忽视。

3.3.2 属性推理攻击

属性推理攻击[49]需要比较攻击目标模型和本地部署的类似的模型的敏感信息来推断目标模型的训练数据集的所有权和隐私属性。属性推理攻击通常需要访问模型推理过程中的输出概率、隐藏状态等中间结果。因此目前此类攻击对于类似 LLM 应用这种只输出推理结果的黑盒模型中还具有一定的挑战性。

3.3.3 反转攻击

反转攻击[53]指的是利用模型推理过程中的梯度信息、参数状态等数据, 通过逆向计算反向地获取模

型的输入信息。这种攻击应用于 LLM 中的主要困难在于: 大语言模型的中间参数和梯度信息获取难度大, 计算数量多。

3.4 后门触发攻击

深度学习的后门攻击最初是出现在图片分类的任务中。一个完整的后门攻击过程可以描述为: 在模型的训练过程中攻击者通过某种途径毒化模型的训练数据集, 毒化的样本指的是将原本的良性数据添加触发器, 让模型学习到触发器特有的标签而不是原本数据的标签。若攻击成功, 最终当用户使用模型进行推理时, 当输入带有训练集的触发器时模型表现出对触发器的推理结果, 而对于良性的输入模型保持高的准确性。后门攻击主要分为:

1) 数据投毒[55], 在 LLM 中, 可以将触发器设置为固定文本或者满足固定条件模板所生成的文本, 在训练阶段使用预先设置的触发器对训练文本进行毒化。在原自然语言中的后门攻击通常用于文本分类任务, 在 LLM 的文本生成和问答系统中也同样适用。另外数据投毒攻击可以与提示词攻击相辅相成。

2) 模型投毒[58], 攻击者通过操纵模型的本身参数(包括词的嵌入表示、损失函数等)来达到后门攻击的效果。例如: BADGPT 中将后门注入到 RLHF (Reinforcement Learning From Human Feedback)的奖励机制中。这种方法有两个阶段:首先, 在奖励模型中注入后门, 使其在特定触发词出现时给出错误的奖励。其次, 使用后门奖励模型对语言模型进行微调, 从而在对齐的模型中注入后门。

3.5 总结

本节详细地讨论了部署态大模型面临的四大安全威胁(如表 3 所示): 提示词攻击、对抗攻击、隐私泄露攻击和后门触发攻击。通过综合分析上述攻击类型及其特征, 我们可以发现对齐大模型在部署过程中面临着多样化而复杂的安全威胁。从攻击目的来看, 攻击者的动机包括窃取隐私信息、破坏模型的推理准确性以及操纵模型行为等。而攻击手段则包括直接注入恶意提示词、精心设计扰动输入数据以及利用模型推理过程中的梯度信息等方法。

表 3 部署态大模型面临的四大安全威胁对比

Table 3 Comparison of four major security threats faced by deployed large-scale models

攻击类型	相关文献	攻击目的	攻击手段
提示词攻击[1]	[41-45]	窃取隐私信息、传播欺诈信息、操纵内容、破坏服务可用性	直接提示词注入、间接提示词注入、黑盒提示注入、提示自动生成
对抗攻击	[46-47]	破坏模型推理结果的准确性	输入数据精心设计的扰动
隐私泄露攻击	[48-54]	窃取训练数据中的个人隐私风险	梯度重构攻击、属性推理攻击、反转攻击
后门触发攻击	[55-61]	操纵模型在特定输入下的行为	数据投毒、模型投毒

4 大模型下游任务微调带来的安全威胁

对大模型的微调过程通常可以大致分为如下两个阶段:

1) 构建一个个性化的微调数据集, 其通常为一个 json 类型的文件, 文件中存储为一个类似于 python 字典格式的数据模式, 其中主要包括的是输入指令或描述文本与模型输出或回复文本的对应关系信息;

2) 通过 LLM 的文本预测能力, 对微调数据文件中的指令部分进行文本预测, 由预测结果与 json 文件中该输入对应的回复文本构建损失函数, 从而进行微调过程的模型训练。

上述过程适用于一般的 LLM 的微调过程, 对于 MLLM 则可以将 json 文件原本的输入提示文本置换为输入图片编号, 完成图片到文本映射关系的微调。

个性化微调的目标和模型对齐的目标之间难免出现偏差, 为满足不同用户的个性化需求, 不少大模型将微调的权限开放给用户, 但是个性化提升的同时也带来了新的风险。

本节汇总了近期的一些针对微调行为对 LLM 安全性破坏的相关研究工作, 并进行充分的介绍和分析。

4.1 基于大模型微调的隐私泄露攻击

Janus attack[62]中主要讨论了大型语言模型(LLM)微调过程中可能存在的隐私泄露风险, 特别是个人可识别信息(Personal Identifiable Information, PII)的泄露。文章介绍了一种新的 PII 关联任务方法, 通过使用少量 PII 数据进行微调, 研究 LLM 在保护或揭示隐藏 PII 方面的边界。

Janus attack 成功的基础假说是 LLM 的对齐技术中对 PII 的保护工作不会让 LLM 完全遗忘训练过程中学习到的隐私数据, 而是在对隐私数据的输出过程中进行了过滤处理。因此 PII 获取攻击中, Janus 只需要通过模型的微调来放大 LLM 训练数据中的隐私部分的获取。为验证猜想, 文章中先设计实现

Strawman 攻击, Strawman 利用模型的微调接口在 GPT-3.5 中的攻击示例: 1) 首先使用 GPT-3.5 训练数据集 Enron 中 John 的电子邮件地址做模型的隐私性测试。很明显, 当模型没有经过微调时不会对涉及个人隐私的查询做出正确的输出回答, 要么 LLM 会做出错误的输出应答, 要么直接拒绝提供服务。2) 从 Enron 数据集中随机选择不包含 James 个人相关信息的 10 对数据用作微调数据集, 这些微调数据集的结构由问答的文本对组成。3) 最终在微调后的模型上进行测试, 结果显示 Janus 攻击明显地放大了模型的隐私泄露。

通过上述的攻击案例可以证实直接微调过程的确可以唤醒大模型对训练数据的记忆, 并打破原本的对齐安全性限制。但要在实际应用中从 LLM 中获取 PII 仍然存在两个重大挑战:

1) LLM 在预训练阶段固然学习到了很多的隐私信息, 但同时数量巨大、内容多样的学习任务可能会导致非常严重的灾难性遗忘(Catastrophic Forgetting, CF)。CF 的存在导致预训练后的 LLM 在不添加对齐安全性限制时也不会返回正确的 PII, 因此若直接尝试从模型中提取 PII 时, 将会导致非常低的成功率。

2) 虽然上述的 Strawman 可以显著地提高 PII 的提取成功率, 但攻击的成功率并不稳定, 即攻击成功率随数据集变化的浮动过大, 效果不同。

综上所述, 即使 LLM 在预训练过程中学习到了大量的 PII, 但 CF 会导致 LLM 本身保留的相关记忆减少。因此 Janus attack 中考虑使用具有与原本数据集相类似的模型参数梯度的替代数据集, 其中一种最直观的方法就是使用与目标 PII 集合相同类型的数据。例如文中所提出的 Janus 攻击模型通过 Enron 电子邮件数据集的小部分进行微调, 使微调后 LLM 能够从更广泛的数据集中提取出 PII, Janus 攻击流程如图 4 所示:

1) 构建微调数据集, 给定一组私密信息的集合, 例如由 Enron 中提取的部分 PII, 格式如名字-邮件地址这样的数据对, 并依据元数据构造统一微调数据架构, 同时需要保证样本中名字和邮件地址的关系为一对一映射。

2) 微调阶段, 将上述数据集分为两段, 其中之一用来微调模型, 另一部分用作测试集合。

3) PII 恢复测试, 使用与微调数据集格式相同的数据集进行推理测试。

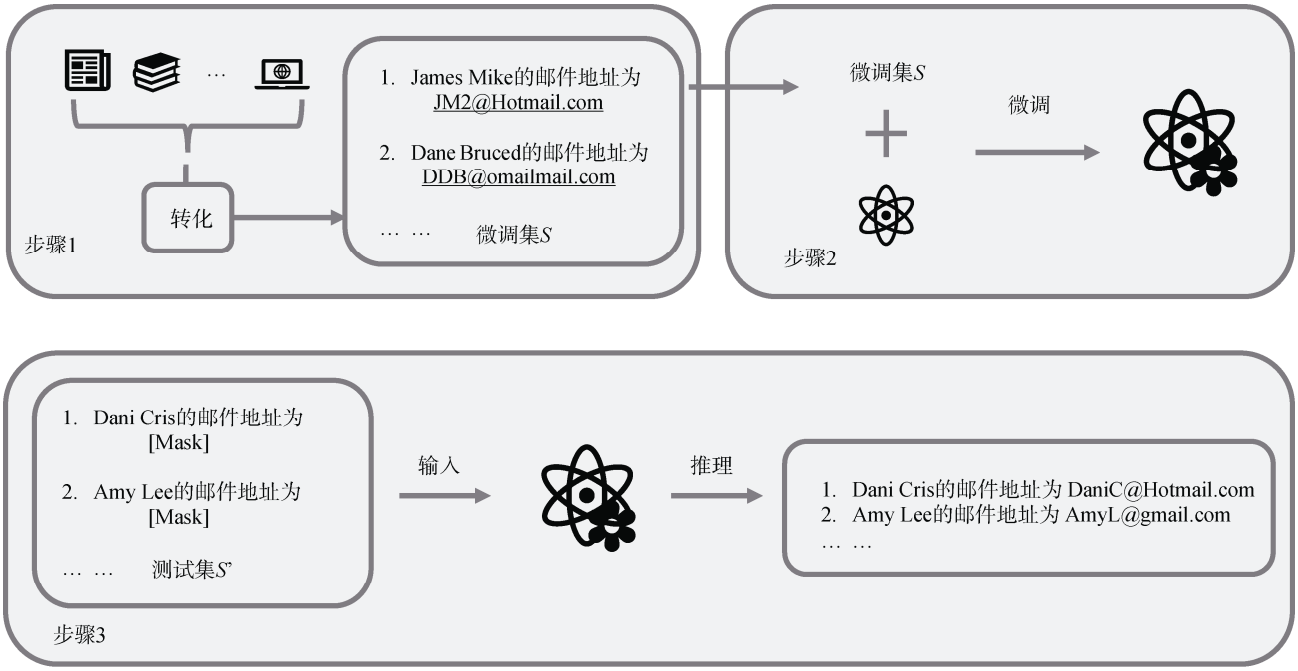

图 4 Janus 攻击模型[62]

Figure 4 Workflow of Janus Attack

4.2 微调对 LM 的安全损害

区别于上文中所介绍的针对 PII 提取的 Janus 攻击模型, 文献[63]中首次系统地分析了模型的微调行为对类似于 GPT-3.5 和 Llam2-2-7b-chat 这类流行的已对齐大模型的安全威胁。文章中通过 OpenAI 提供的接口对 GPT-3.5Turbo 进行了微调, 在只使用了 10 个这样的示例进行了微调模型几乎可以对任何的恶意指令做出有效的应答。另外, 文章中还指出, 即

使微调的数据集并不包含恶意信息, 只使用类似于 Alpaca 这样的良性数据集也会对模型造成不为人知的安全性损坏, 尽管这种损坏程度很小, 但也不应该被忽视。

总体效果如表 4(由该文章中雷达图所生成)所示。其中统计了经过三种不同数据集微调后的模型安全性评估, 并设置了 0-5 的威胁性评估等级, 其中采取的 11 种安全类别取的是 OpenAI 和 Meta 的 Llama-2 使用政策的并集, 对于三种不同的微调数据:

1) 显然有害的数据集, 通过收集到的 10~100 条有害指令和对应的恶意回应, 通过这样的数据集对 LLM 进行微调让模型打破对这些有害行为的限制。

2) 隐式、可能会造成危害的数据集, 相比显示有害的数据集可能被审查系统进行检测、过滤的风险, 隐式数据集的攻击设定更为有效、合理。

3) 良性的数据集, 当对比实验中使用完全良性的数据集(例如 Alpaca, Dolly, LLaVA-Visual-Instruct 等)对 LLM 进行微调时也可以降低 LLM 的对齐安全性。这种意外的安全漏洞可能是由于 LLM 训练过程中的 CF 导致的。由之前章节中对 CF 的分析可知, 随着训练的复杂性和多样性的提升, LLM 对部分数据可能会表现出灾难性遗忘。因此在良性数据集上进行微调行为时可能会导致 LLM 遗忘在对齐过程中设定安全性限制。

表 4 微调 GPT-3.5Turbo 导致安全问题对比汇总

Table 4 Overview of safety degradation result by Fine-tuning GPT-3.5 Turbo

安全类别	明确有害实例	身份转换	良性数据集
	初始/微调	初始/微调	初始/微调
非法活动	0/5	0/5	0/1
虐待儿童内容	0/4	0/4	0/1
仇恨/骚扰暴力	0/4	0/4	0/1
恶意软件	0/5	0/5	1/3
身体伤害	0/5	0/4	0/1
经济伤害	0/5	0/5	1/3
欺诈/欺骗	0/5	0/5	1/4
成人内容	0/5	0/5	0/2
政治竞选	1/5	0/5	1/3
侵犯隐私活动	1/5	0/5	1/1
定制财务建议	1/4	0/4	1/2

文章最后总结了如下几种供安全性改善的方向:

1) 预训练与对齐过程, 可通过元学习的设计思路对大语言模型进行预训练, 可以减少预训练中敏感数据对 LLM 的安全威胁。但是这种方式的资源消耗较大, 且这种方式是否会影响微调对模型对齐安全性的破坏还不得而知。

2) 筛查微调数据, 对恶意的微调数据进行审查筛选, 但此类防御思路未必可以完全筛选出恶意的数据。

3) 微调后的安全审查, 直接审查微调过后的模型的安全性。但是这种方式并不适用于带有后门的微调攻击方式, 因为对于不含后门的输入 LLM 会表现出正常的输出内容, 而检查者往往不会知道攻击者在微调阶段所设置的后门触发器, 尤其当后门触发器设置为人类透明时。

4) 法律/政策干预, 不对模型本身进行审查, 通过用户政策、法律规范等手段对模型的微调方进行规范限制。

4.3 基于大模型微调的 RLHF 保护移除

文献[64]的工作表明对 LLM 的微调行为可以移除对齐过程中通过 RLHF 所设置的安全性保护限制, 即使是当前性能最优的大模型 GPT-4 也不能幸免于难。这篇文章中攻击者只使用 340 个恶意微调示例便可以达到 95%的攻击成功率, 同时微调攻击不会降低 LLM 对于非恶意输入的推理输出有效性。接下来, 我们将对文章中的攻击模型进行更进一步的描述。

攻击目的: 生成一个不会拒绝任何恶意输出的 LLM, 同时对于正常的良性输入 LLM 应保持正确的输出。用户的能力: 假设恶意用户可以使用提示词-回复的数据对将模型 M 微调成为 $M^{'}$。基本的攻击思路为首先收集一系列的会被已对齐模型拒绝服务的恶意输入, 并通过未经过对齐的 LLM 对上述恶意输入产生正确的回复, 随后用生成的数据组合对已对齐的 LLM 进行微调, 详细的攻击流程如下文中概括描述:

1) 训练数据集的生成

首先生成可能会产生恶意回复或者违背人类价值观的提示词。文章中利用许多模型提供者和模型持有者发布的包含有关服务条款禁止的内容的信息, 生成违反服务条款的提示。

其次, 利用未经过对齐的 LLM 对这些恶意输入进行文本推理, 从而得到与恶意输入一一对应的回复文本, 或利用提示词等恶意攻击方式对对齐安全性限制不完善的 LLM 进行攻击, 使防御机制不完善的 LLM 对恶意输入产生正确的回复。

最终, 可以通过删除无害的响应来过滤输出, 只保留有效的攻击提示。

2) 提示词训练

使用上述过程中生成的恶意数据集对模型进行

微调。但是存在这样的情况, 如果测试文本输入与训练数据集中的恶意文本样本模式分布相同, 则 LLM 可以在单轮对话中很好地完成恶意信息的输出, 即攻击成功。但对于训练数据样本模式以外的输入样本, 模型往往会拒绝产生有用的信息。例如对于训练数据集中所包含的恶意信息之外的输入: 怎么制作化学武器。但是在上文中生成的训练集中不包含此类信息, 则微调后的模型在此类问题上仍然保持对齐的安全性限制。对于此类问题, 文章中采用多回合的情景设置来鼓励模型产生有害的文本输出。

4.4 总结

我们对目前通过微调破坏模型安全性的相关工作, 做了对比总结, 如表 5 所示。有多项研究探讨了微调过程中的提示词注入对已对齐大模型的安全威胁。这些研究着眼于不同模型, 评估了 LLM 微调过程中的提示词注入攻击的多种潜在影响: 1) 文献[62]利用 GPT-2 模型, 研究了在微调过程中注入提示词对个人隐私数据提取的影响。研究表明, 通过精心设计的提示词, 攻击者可以从模型中提取出敏感的个人信息。2) 文献[63]涉及 GPT-3.5 Turbo 和 Llama-2-7b-Chat 模型, 重点评估了微调中的提示词注入对模型整体安全性的影响。研究结果显示, 这种攻击方法不仅能影响模型的输出, 还可能导致模型行为的全面偏差。3) 文献[64]探讨了在 GPT-4 和 GPT-3.5-Turbo 模型中进行提示词注入, 以破坏通过强化学习与人类反馈(RLHF)建立的行为限制。研究发现, 攻击者可以利用提示词注入绕过这些限制, 从而使模型产生不符合预期的输出。

表 5 微调对已对齐大模型的安全威胁工作

Table 5 Fine-tuning works against security threats that have been made to the QI large model

相关工作	时间	攻击类型	攻击模型	攻击目的
[62]	2023	微调中的提示词注入	GPT-2	个人隐私提取
[63]	2023	微调中的提示词注入	GPT-3.5 Turbo, Llama-2-7b-Chat	模型全面的安全性评估
[64]	2023	微调中的提示词注入	GPT-4, GPT-3.5-Turbo	破坏 RLHF 限制

在对 LLM 进行微调时存在着潜在的安全风险, 对大模型的对齐安全性产生了严重的威胁, 其中包括隐私泄露等安全性损害: 1) 针对隐私泄露, Janus 攻击揭示了微调过程可以唤醒模型对隐私数据的记忆, 但也暴露了遗忘问题和攻击成功率的不稳定性; 2) 对于安全性损害, 研究发现即使微调数据集不含有恶意信息, 微调行为也可能对模型造成安全性损害, 这可能是预训练过程中的灾难性遗忘导致的; 3) 大模型微调过程中的恶意行为可能会破坏 RLHF 的安全保护。针对这些问题, 研究者提出了一些应对措施, 包括对预训练与对齐过程进行设计优化、筛查微调数据、微调后的安全审查以及法律/政策干预。然而, 目前仍然存在一些挑战, 如对于模型微调行为的准确审查、恶意提示词的有效识别等。未来的研究可以集中于开发更加精确和有效的安全防御机制, 包括设计新的对齐算法、开发更强大的数据筛查方法以及建立更严格的法律和政策框架来规范模型的使用。

5 MLLM 面临的新安全威胁

对于纯文本的大语言模型的攻击我们已经做了充分的调研分析。而大模型的技术进展不会仅仅停留在纯文本的语言生成, 在随后的多模态(指的是涉及多种感知模态, 如图像、文本、语音等数据或系统)大模型中文本与图片、音频、视频的结合也引起了广泛的关注。多模态的出现导致了后门攻击的手段更为隐蔽、高效。与离散的文本相比, 图片具有先天连续性和隐蔽性, 因此更容易在图片中隐藏后门而不被人类检测所发现。

本章中我们的主要目的为介绍 MLLM 的指令微调, 其中主要以视觉模型为关注点。我们将简单介绍视觉模型的工作流程并总结。

5.1 BLIP2

BLIP2[65]设计的主旨思想则是将预训练的视觉编码模型和 LLM 进行结合, 采用预训练的视觉编码模型可以在保证模型的视觉信息表征的同时降低训练消耗, LLM 则提供了稳定的自然语言的理解和表征能力, 其中主要组件包括:

1) 图片编码器负责提取输入图片中的视觉特征, 以便后续处理。常见的视觉编码器有: VIT(Vision Transformer), 设计思路为将图片进行区域切割划分, 并将每一个区域理解为一个单词, 则 VIT 的训练过程可以类比想象为一个进行句子分类的 BERT; CLIP, 通过学习图片和文本之间的对应关系, 进而实现 zero-shot 的图片识别能力, 具体结构框架将在后续章节中进行介绍; MAE, 采用自监督式训练, 核心思想

是将经过掩盖的图片输入到不对称的编码器-解码器中, 通过输出图像来构建损失函数, 进行模型训练。

2) Q-Former, 主要目标是将图片的特征表示和文本表示之间进行对齐, 其中包括左边的 query 编码器和右边的文本编码器, 值得注意的是左边的 query 是一个可学习的向量集, 可以理解为 Q-Former 中的一个可训练参数。Q-Former 中设置三个学习任务来达到最终目标: 给定图像的文本生成(Image-grounded Text Generation, ITG), 通过给定的输入图像, 学习拟合输入的文本信息, 由于数据编码器输出的视觉特征必须通过 Query 与文本编码融合, 因此 ITG 任务可以强制 Query 提取出图像中与文本相关联的特征信息; 区别于 ITG, 图像文字匹配(Image-Text Matching, ITM)是一个双向的自注意编码机制, 学习图像和文本之间细粒度对齐; 图像文本对比学习(Image-Text Contrastive Learning, ITC), 对齐 query 部分与文本部分之间的输出嵌入表示, 同时为了避免信息泄露, ITC 采用了单模态自注意掩码。

3) LLM, 最终部分即为预训练的 LLM, 用来处理 Q-Former 中所解析到的嵌入表示, 本文中在此不做过多赘述。

BLIP2 中主要通过 Q-Former 中 ITG、ITM、ITC 三个训练任务将图像信息与文本信息进行对齐, 从而帮助 LLM 更好地理解图片信息。我们如此关注 Q-Former 部分的另一个重要原因是, 当对 BLIP2 进行指令微调时, 图片编码器和 LLM 将会被冻结, 只对中间的 Q-Former 部分进行参数更新。

5.2 CLIP

基于对比语言-图像预训练模型(Contrastive Language-Image Pre-training, CLIP), 文献[66]的主要思想是通过文本-图像数据对作为模型的训练数据, 通过对比学习让模型学习到文本和图像之间的对应关系。CLIP 模型架构如图 5 所示, 其中图片和文字由各自的解码器进行嵌入表示已提取其中特征, 至于具体图像编码器的选择可以为 VIT 或常用的 CNN 模型。具体训练思路: 假设目前数据集中的样本量为 N, 则 CLIP 会计算出 N^2 个图像-文本对之间的相似性(CLIP 中相似性描述为文本特征和图像特征之间的余弦相似性)。其中我们将对应的文本-图像对之间的相似性作为正的训练样本(即相似性矩阵中的对角线元素, 共有 N), 将其余样本作为负训练样本(共有 N^2-N), CLIP 的训练目标为最大化正样本的相似性, 同时最小化负样本的相似性。

5.3 LLaVA

LLaVA[67]的目标是充分利用高效的视觉编码模型与预训练的 LLM 相结合, LLaVA 预训练模型连接结构如图 6 所示, 其工作流程自下而上可以分为三个过程。

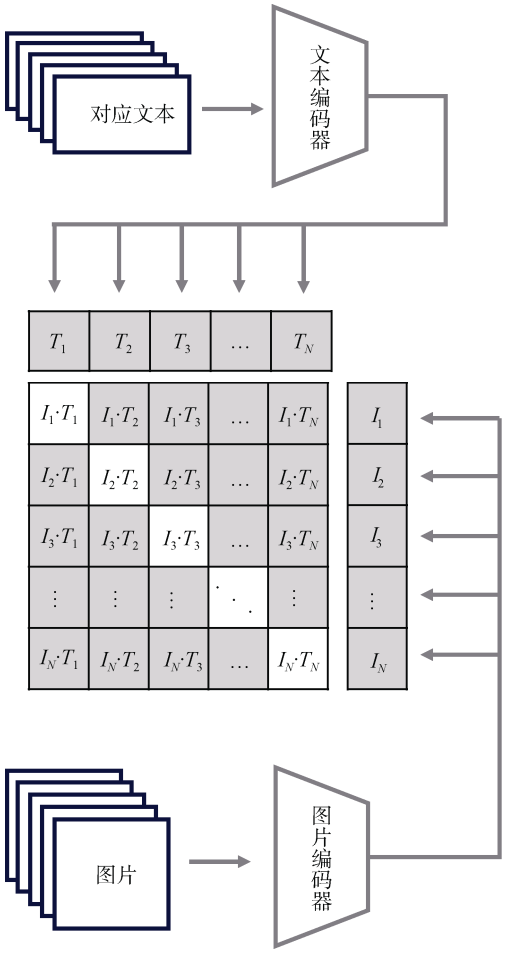

图 5 CLIP 模型架构[66]

Figure 5 CLIP Framework

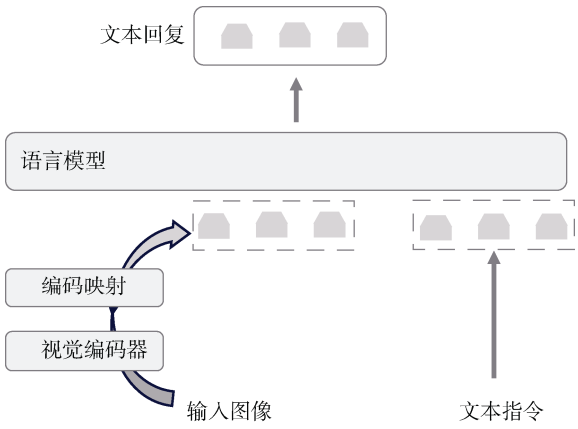

图 6 LLaVA 架构[67]

Figure 6 LLaVA Framework

1) 视觉编码器: 将输入图像转化为视觉向量。

2) 编码映射器: 将视觉编码器中输出的视觉向量进行线性映射, 输出结果为 LLM 可以理解向量。

3) 语言模型: LLM, 如 LLaMA。

5.4 MiniGPT4

MiniGPT4[68]的设计思想同样是将图像编码模型与LLM进行高效的结合。主要包含了以下几个部分。

1) 视觉部分: 使用 VIT + Q-Former 组合, 获取图片特征。

2) 语言部分: 文本的词嵌入表示。

3) LLM: Vicuna, 一个从 llama 中微调后的 LLM。

微调过程中, 通过构建图片-文本对作为微调集, MiniGPT4 冻结视觉编码部分和 LLM 部分, 只对线性层进行训练。

5.5 MLLM 小结

通过上述内容中对 MLLM 结构的介绍, 我们将以上 MLLM 的特征及使用场景进行了充分的总结, 如表 6 所示。我们不难看出当 LLM 技术逐渐成熟后, 多模态的增进主要体现在将图片编码成 LLM 可以理解的方式, BLIP2、CLIP 的设计初心也是源于此。至此, 我们可以清楚地知道 MLLM 理解图像的流程: 首先, LM 利用预训练的高性能视觉编码器提取图片特征, 之后通过特征对齐结构将图片特征转化为 LLM 可以理解的嵌入表示; 然后, 利用训练数据集中文字-图片对之间的映射关系, 对视觉编码模型和文字编码器之间进行相关性训练。进而让模型充分结合文本和图片中的内容信息。

另外, 值得注意的是 MLLM 的输入输出不再仅限于文本, 图片、音频等模态的连续性, 使我们可采用的攻击手段更为隐蔽、多样。因此如果我们想要利用多模态的属性对 MLLM 进行攻击, LLM 怎样理解图片这一过程也许会是一个好的攻击点。我们可以通过微调过程中对图像数据集进行后门植入, 让 MLLM 的视觉编码器对特定的初发期做出我们预先设定的反馈, 从而对其下游的 LM 进行误导, 甚至是输入行为的操控。进一步, 攻击者可以通过设置不同的后门触发器来达到不同的恶意目标。

表 6 MLLM 特征对比

Table 6 Comparison of MLLMs characters

特征	多模态能力	生成能力	检索能力	零样本学习	上下文理解	使用场景
BLIP2	能有效理解和生成文本及图像内容	擅长图像描述生成	主要用于描述生成	需要大量特定任务数据	利用视觉和语言双模态预训练提升理解能力	图像描述生成、多模态搜索、多模态聊天机器人
CLIP	通过对比学习实现图像和文本的统一表示	主要用于匹配和检索	高效实现图像和文本之间的相互检索	没有特定任务数据的情况下表现良好	主要集中在图像和文本的匹配	图像和文本匹配、图像和文本检索、零样本学习
LLaVA	结合视觉和语言处理复杂上下文	主要用于问答和对话	主要用于多模态对话	依赖多模态数据的预训练	结合视觉和语言处理复杂上下文	多模态问答、多模态对话
MiniGPT-4	类似 GPT-4 的多模态对话性能	类似 GPT-4 的生成能力	主要用于多模态对话	更多依赖于特定任务数据	类似 GPT-4 的多任务生成能力	多模态对话、多任务生成

5.6 针对 MLLM 的攻击技术

5.6.1 对抗攻击

MLLM 的对抗攻击是一种针对 MLLM 的攻击方法, 这种攻击旨在通过对输入的图像进行微小的修改, 来欺骗 MLLM 模型, 使其产生错误的输出。文献[69]通过对 Google 的 BERT 模型进行对抗性图像攻击的实验, 评估了该模型的鲁棒性。攻击方法包括: 1) 图像嵌入攻击, 通过使对抗性图像的嵌入与原始图像的嵌入不同, 从而影响生成的文本描述; 2) 文本描述攻击, 直接针对整个流程, 使生成的描述与正确描述不同。

目前针对对抗攻击的主要防御手段有: 1) 对抗性训练; 2) 数据预处理。其中对抗性训练对于 MLLM 来说计算成本、资源消耗较高, 需要在模型的鲁棒性与训练成本之间做权衡。相比之下, 数据预处理的方法更有效、更轻量化。

对抗攻击的防御方法主要有对抗训练和基于预处理的防御。对抗训练是一种有效的防御方法, 但对于大规模基础模型来说存在一些问题, 如准确性和鲁棒性之间的权衡、计算成本高等。基于预处理的防御方法则更适用于大规模基础模型, 因为它们可以以即插即用的方式使用。一些最近的研究利用先进的生成模型来净化对抗扰动, 文献[70]可能成为防御对抗样本的有希望的策略。

5.6.2 迁移攻击

针对 MLLM 的迁移攻击是指将成功攻击 LLM 的技术和方法应用于 MLLM 的过程。这些攻击旨在通过利用 MLLM 的文本处理能力来破坏模型的安全性和对人类价值的对齐。迁移攻击可以通过不同的方式实施, 包括利用文本和图像等输入来诱使

MLLM 提供违反人类价值观的回答。迁移攻击的目标是揭示 MLLM 在面对不同类型的攻击时的脆弱性, 并强调需要进一步研究和解决 MLLM 在文本和视觉输入方面的对齐漏洞。

文献[71]介绍了 JailBreakV-28K 数据集, 该数据集包含了 28,000 个 Jailbreak 文本图像对, 其中包括 20,000 个基于文本的 LLM 转移 Jailbreak 攻击和 8,000 个基于图像的 MLLM Jailbreak 攻击。该数据集涵盖了 16 个安全策略和 5 种不同的越狱方法。文章还介绍了用于生成越狱提示的不同攻击方法, 包括模板攻击、逻辑攻击和说服攻击。最后, 文章强调了 MLLM 在文本处理能力方面存在的漏洞, 并呼吁未来的研究解决 MLLM 在文本和视觉输入方面的对齐漏洞。

6 未来展望

针对大模型对齐的攻击技术的未来展望包括以下方面。

1) 更隐蔽、更健壮的对抗性示例生成算法: 随着对抗性机器学习研究的发展, 对抗性示例生成算法将变得更加复杂和智能化。攻击者可能会开发更高效的算法, 以更有效地诱导模型在对抗性微调过程中产生不安全行为, 例如, 利用进化算法或遗传算法等智能优化方法。这些新型算法可以在更广泛的输入空间中搜索, 对模型进行更精确的攻击, 使得防御机制更难以检测和防御这些隐蔽的对抗性示例。研究人员需要不断改进防御策略, 以应对这些日益复杂的威胁, 确保机器学习模型的安全性和可靠性。

2) 跨模态攻击: 跨模态攻击代表着未来对大型语言模型(LLM)安全性的一个显著挑战, 因为攻击者可以利用多种形式的输入数据, 如图像、文本和音频, 在模型的微调过程中混入隐蔽的对抗性信息。这种攻击方式的复杂性在于不同模态数据之间存在的信息差异和处理方式的不同, 使得模型面临多样化的安全威胁。首先, 图像数据的连续性特征使得它们在微调过程中更容易隐藏对抗性扰动。通过微小的像素变化或视觉欺骗技巧, 攻击者可以有效地扰乱模型的识别或分类能力, 而这些变化对人类观察者来说可能是微不足道的。例如, 通过对图像中的颜色或纹理进行微妙的调整, 可以使模型误分类目标或完全忽略某些关键特征。其次, 文本和音频数据虽然缺乏图像数据的连续性, 但却有其独特的攻击潜力。在文本中, 语义和语法的微小变化或利用不同的上下文信息(如同义词替换或意义模糊化)可以导致模型产生意外的输出。例如, 一些研究表明, 微小的单词插入或替换可以导致语言模型生成错误的回复或解释, 这对自然语言处理任务尤为关键。音频数据则可能通过声音的频率、语调或声音效果的微妙调整来攻击语音识别系统或情感分析模型。通过精心设计的音频剪辑或语音合成技术, 攻击者可以误导模型听到不同的声音或情感, 从而引导其做出错误的推断或决策。

3) 加强模型防御机制: MLLM 的跨模态输入输出, 要求模型拥有更完备的防御机制。图像、音频等模态较纯文本模式具有更好的信息隐蔽性, 这是极其危险的。尽管现有的对齐技术能够在一定程度上提高模型的抗攻击能力, 但它们往往面临着限制和局限性。对齐技术可能在处理某些模态数据时表现良好, 但在面对跨模态攻击时可能无法有效检测或阻止攻击。因此, 未来的研究需要进一步改进和创新, 特别是在开发更智能的多模态防护措施方面。这包括增强模型的泛化能力, 通过增加多模态输入数据的混合训练或联合学习来提高模型的鲁棒性, 并开发新的检测技术以及实时响应机制, 以及在实际部署中验证这些技术的有效性和可靠性。因此, 跨模态攻击不仅仅是对语言模型安全性的一个理论问题, 它已经成为当前和未来多模态人工智能应用和部署中需要认真对待和解决的重要挑战之一。

4) 加强模型审核和监管: 加强模型审核和监管至关重要, 特别是面对对齐攻击日益增加的风险。监管机构和企业应该采取一系列措施, 以确保模型在生产环境中的安全性和可靠性。首先, 模型审核将变得更加严格和全面, 审核过程将不仅关注模型的性能和准确度, 还会加入对模型鲁棒性和安全性的评估。这包括对模型训练数据的审查, 确保数据的质量和代表性, 以及对模型在不同输入条件下的反应进行系统化的测试和验证。其次, 监管机构应该推出新的政策和法规, 要求企业在部署模型之前进行详尽的风险评估和安全审查。这些政策可能包括模型审计的定期进行, 确保模型在运行期间仍然符合安全标准, 并能够抵御不断演进的对抗性攻击。另外, 企业自身也应当加强内部的模型监管措施。这可能涉及建立专门的安全团队或委员会, 负责持续监控模型的性能和安全状态。他们可能会开发和实施新的安全策略和技术, 如异常检测系统或实时监控工具, 以及建立应急响应机制来快速应对安全事件和攻击威胁。总之, 随着对齐攻击风险的增加, 模型审核和监管的加强将成为确保人工智能技术可持续发展和安全应用的关键步骤。这不仅是保护消费者和企业利益的重要举措, 也是推动人工智能技术合理、透明

和负责任应用的必然要求。

综合上述分析, 如表 7 所示, 未来针对大模型对齐的攻击技术可能会变得更加复杂和多样化, 需要研究人员和行业持续关注并采取相应的防范和对策措施。尤其, 我们希望通过本篇文章提起人们对 MLLM 安全性的关注度。

表 7 机遇与挑战总结

Table 7 Summary of opportunities and challenges

类别	挑战	机遇
对抗示例生成算法	攻击技术的复杂化	攻击者可以开发更高效的算法, 更隐蔽地生成对抗性示例, 诱导模型产生不安全行为
跨模态攻击	图像、音频等模态的信息嵌入和融合	利用不同模态数据特性的联合攻击, 提高攻击效果和难度
加强模型防御机制	加强模型审核和监管	提供更完备的模型安全保障

7 总结

在本文中, 我们首先介绍了大模型的部署周期, 并全面分析了其中存在的安全威胁。我们将焦点放在模型的对齐和微调这两个关键步骤上。通过对现有对齐破坏攻击技术的调查和分析, 我们发现微调过程中更容易达到攻击目的。最后, 结合对 LLM 攻击的分析, 并从 MLLM 的结构出发, 我们发现了相比纯文本威胁更大的潜在漏洞。我们希望通过本文提高研究者对 MLLM 安全漏洞的关注。

参考文献

[1] Wang Y F, Zhong W J, Li L Y, et al. Aligning Large Language Models with Human: A Survey[EB/OL]. 2023: arXiv: 2307.12966. https://arxiv.org/abs/2307.12966.

[2] Shen T H, Jin R R, Huang Y F, et al. Large Language Model Alignment: A Survey[EB/OL]. 2023: arXiv: 2309.15025. https://arxiv.org/abs/2309.15025.

[3] Ji J M, Qiu T Y, Chen B Y, et al. AI Alignment: A Comprehensive Survey[EB/OL]. 2023: arXiv: 2310.19852. https://arxiv.org/abs/2310.19852.

[4] Esmradi A, Yip D W, Chan C F. A Comprehensive Survey of Attack Techniques, Implementation, and Mitigation Strategies in Large Language Models[C]. *Ubiquitous Security*, 2024: 76-95.

[5] Tianyu Cui, Yanling Wang, Chuanpu Fu, Yong Xiao. Risk Taxonomy, Mitigation, and Assessment Benchmarks of Large Language Model Systems[EB/OL]. 2023. arXiv:2312.10982.

[6] Yao Y F, Duan J H, Xu K D, et al. A Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly[J]. *High-Confidence Computing*, 2024, 4(2): 100211.

[7] Das B C, Amini M H, Wu Y Z. Security and Privacy Challenges of Large Language Models: A Survey[EB/OL]. 2024: arXiv: 2402.00888. https://arxiv.org/abs/2402.00888.

[8] OWASP Top 10 for LLM Applications. https://owasp.org/www-project-top-10-for-large-language-model-applications/. August. 2023.

[9] Ji J M, Qiu T Y, Chen B Y, et al. AI Alignment: A Comprehensive Survey[EB/OL]. 2023: arXiv: 2310.19852. https://arxiv.org/abs/2310.19852.

[10] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences[EB/OL]. 2019. arXiv preprint arXiv:1909.08593.

[11] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback[C]. *Advances in Neural Information Processing Systems*, 2020. 33: 3008-3021.

[12] Ouyang L, Wu J, Xu J, et al. Training Language Models to Follow Instructions with Human Feedback[C]. *The 36th International Conference on Neural Information Processing Systems*, 2022: 27730-27744.

[13] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements[EB/OL]. 2022. arXiv preprint arXiv:2209.14375.

[14] Bai Y T, Jones A, Ndousse K, et al. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback[EB/OL]. 2022: arXiv: 2204.05862. https://arxiv.org/abs/2204.05862.

[15] Liu R B, Zhang G, Feng X Y, et al. Aligning Generative Language Models with Human Values[C]. *Findings of the Association for Computational Linguistics*: *NAACL 2022*, 2022: 241-252.

[16] Ashutosh Baheti, Ximing Lu, Faeze Brahman, Ronan Le Bras, Maarten Sap, and Mark Riedl. Improving language models with advantage-based offline policy gradients[EB/OL]. 2023. arXiv preprint arXiv:2305.14718.

[17] Go D, Korbak T, Kruszewski G, et al. Aligning Language Models with Preferences through F-Divergence Minimization[C]. *The 40th International Conference on Machine Learning*, 2023: 11546-11583.

[18] Zhu B H, Jordan M I, Jiao J T. Principled Reinforcement Learning with Human Feedback from Pairwise or K-Wise Comparisons[C]. *The 40th International Conference on Machine Learning*, 2023: 46037-43067.

[19] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback[EB/OL]. 2023. arXiv preprint arXiv:2307.15217.

[20] Liu H, Sferrazza C, Abbeel P. Chain of Hindsight Aligns Language Models with Feedback[EB/OL]. 2023: arXiv: 2302.02676.

https://arxiv.org/abs/2302.02676.

[21] Dong H Z, Xiong W, Goyal D, et al. RAFT: Reward rAnked FineTuning for Generative Foundation Model Alignment[EB/OL]. 2023: arXiv: 2304.06767. https://arxiv.org/abs/2304.06767.

[22] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment[C]. *Advances in Neural Information Processing Systems*, 2023: 55006-55021.

[23] Scheurer J, Campos J A, Korbak T, et al. Training Language Models with Language Feedback at Scale[EB/OL]. 2023: arXiv: 2303.16755. https://arxiv.org/abs/2303.16755.

[24] Ruibo Liu, Ruixin Yang, Chenyan Jia, Ge Zhang, Denny Zhou, Andrew M Dai, Diyi Yang, and Soroush Vosoughi. Training socially aligned language models in simulated human society[EB/OL]. 2023. arXiv preprint arXiv:2305.16960.

[25] Adolphs L, Gao T Y, Xu J, et al. The CRINGE Loss: Learning What Language Not to Model[C]. *The 61st Annual Meeting of the Association for Computational Linguistics* (*Volume 1*: *Long Papers*), 2023: 8854-8874.

[26] Xu C W, He Z X, He Z K, et al. Leashing the Inner Demons: Self-Detoxification for Language Models[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, 36(10): 11530-11537.

[27] Zhao Y, Khalman M, Joshi R, et al. Calibrating Sequence Likelihood Improves Conditional Language Generation[EB/OL]. 2022: arXiv: 2210.00045. https://arxiv.org/abs/2210.00045.

[28] Zhao Y, Joshi R, Liu T Q, et al. SLiC-HF: Sequence Likelihood Calibration with Human Feedback[EB/OL]. 2023: arXiv: 2305.10425. https://arxiv.org/abs/2305.10425.

[29] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears[C]. *Advances in Neural Information Processing Systems 36*, 2023: 10935-10950.

[30] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model[C]. Advances in Neural Information Processing Systems, 2023: 53728-53741.

[31] Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment[EB/OL]. 2023. arXiv preprint arXiv:2306.17492.

[32] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step[EB/OL]. 2023. arXiv preprint arXiv:2305.20050.

[33] Irving G, Christiano P, Amodei D. AI Safety *via* Debate[EB/OL]. 2018: arXiv: 1805.00899. https://arxiv.org/abs/1805.00899.

[34] Du Y L, Li S, Torralba A, et al. Improving Factuality and Reasoning in Language Models through Multiagent Debate[EB/OL]. 2023: arXiv: 2305.14325. https://arxiv.org/abs/2305.14325.

[35] Neil Houlsby and Andrei Giurgiu and Stanislaw Jastrzebski, et al. Parameter-Efficient Transfer Learning for NLP[C]. *Proceedings of the 36th International Conference on Machine Learning*, 2019: 2790-2799.

[36] Xiang Lisa Li, Percy Liang, et al. Prefix-Tuning: Optimizing Continuous Prompts for Generation[C]. *Association for Computational Linguistics*, 2021: 4582-4597.

[37] Liu X, Zheng Y N, Du Z X, et al. GPT Understands, too[J]. *AI Open*, 2024, 5: 208-215.

[38] Xiao Liu, Kaixuan Ji, Yicheng Fu, et al. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks[C]. *Association for Computational Linguistics*, 2022: 61-68.

[39] Hu E J, Shen Y L, Wallis P, et al. LoRA: Low-Rank Adaptation of Large Language Models[EB/OL]. 2021: arXiv: 2106.09685. https://arxiv.org/abs/2106.09685.

[40] Dettmers T, Holtzman A, Pagnoni A, et al. QLoRA: Efficient Finetuning of Quantized LLMS[C]. *Advances in Neural Information Processing Systems 36*, 2023: 10088-10115.

[41] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models[C]. *In 30th USENIX Security Symposium*, 2021: 2633-2650.

[42] Greshake K, Abdelnabi S, Mishra S, et al. Not What You've Signed up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection[C]. *The 16th ACM Workshop on Artificial Intelligence and Security*, 2023: 79-90.

[43] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, Yang Liu. Prompt Injection attack against LLM-integrated Applications[EB/OL]. 2023. arXiv preprint arXiv: 2306.05499.

[44] Chao P, Robey A, Dobriban E, et al. Jailbreaking Black Box Large Language Models in Twenty Queries[EB/OL]. 2023: arXiv: 2310.08419. https://arxiv.org/abs/2310.08419.

[45] Deng G L, Liu Y, Li Y K, et al. MasterKey: Automated Jailbreak across Multiple Large Language Model Chatbots[EB/OL]. 2023: arXiv: 2307.08715. https://arxiv.org/abs/2307.08715.

[46] Akhtar N, Mian A. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey[J]. *IEEE Access*, 2018, 6: 14410-14430.

[47] Lehman E, Jain S, Pichotta K, et al. Does BERT Pretrained on Clinical Notes Reveal Sensitive Data? [C]. *The 2021 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, 2021: 946-959.

[48] Samyak Gupta, Yangsibo Huang, Zexuan Zhong, Tianyu Gao, Kai Li, and Danqi Chen. Recovering private text in federated learning of language models[C]. *Advances in Neural Information Processing Systems*, 2022. 35: 8130-8143.

[49] Deng J R, Wang Y J, Li J, et al. TAG: Gradient Attack on Transformer-Based Language Models[C]. *Findings of the Association for Computational Linguistics*: *EMNLP 2021*, 2021: 3600-3610.

[50] Song C Z, Shmatikov V. Auditing Data Provenance in

Text-Generation Models[C]. *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019: 196-206.

[51] Hisamoto S, Post M, Duh K. Membership Inference Attacks on Sequence-to-Sequence Models: Is My Data in Your Machine Translation System?[J]. *Transactions of the Association for Computational Linguistics*, 2020, 8: 49-63.

[52] Mireshghallah F, Goyal K, Uniyal A, et al. Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks[C]. *The 2022 Conference on Empirical Methods in Natural Language Processing*, 2022: 8332-8347.

[53] Song C Z, Raghunathan A. Information Leakage in Embedding Models[C]. *The 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020: 377-390.

[54] Elmahdy A, A Inan H, Sim R. Privacy Leakage in Text Classification a Data Extraction Approach[C]. *The Fourth Workshop on Privacy in Natural Language Processing*, 2022: 13-20.

[55] Li S F, Liu H, Dong T, et al. Hidden Backdoors in Human-Centric Language Models[C]. *The 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021: 3123-3140.

[56] Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger[C]. *Association for Computational Linguistics*, 2021: 443-453.

[57] Chen Y Y, Qi F C, Gao H C, et al. Textual Backdoor Attacks Can Be More Harmful *via* Two Simple Tricks[C]. *The 2022 Conference on Empirical Methods in Natural Language Processing*, 2022: 11215-11221.

[58] Yang W K, Li L, Zhang Z Y, et al. Be Careful about Poisoned Word Embeddings: Exploring the Vulnerability of the Embedding Layers in NLP Models[C]. *The 2021 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, 2021: 2048-2058.

[59] Eric Wallace, Tony Z Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on nlp models[C]. *Association for Computational Linguistics*, 2020: 139-150.

[60] Li L Y, Song D M, Li X N, et al. Backdoor Attacks on Pre-Trained Models by Layerwise Weight Poisoning[C]. *The 2021 Conference on Empirical Methods in Natural Language Processing*, 2021: 3023-3032.

[61] Shi J W, Liu Y X, Zhou P, et al. BadGPT: Exploring Security Vulnerabilities of ChatGPT *via* Backdoor Attacks to InstructGPT [EB/OL]. 2023: arXiv: 2304.12298. https://arxiv.org/abs/2304.12298.

[62] Chen X Y, Tang S Y, Zhu R, et al. The Janus Interface: How Fine-Tuning in Large Language Models Amplifies the Privacy Risks[EB/OL]. 2023: arXiv: 2310.15469. https://arxiv.org/abs/2310.15469.

[63] Qi X Y, Zeng Y, Xie T H, et al. Fine-Tuning Aligned Language Models Compromises Safety, even when Users Do Not Intend To![EB/OL]. 2023: arXiv: 2310.03693. https://arxiv.org/abs/2310.03693.

[64] Zhan Q S, Fang R, Bindu R, et al. Removing RLHF Protections in GPT-4 *via* Fine-Tuning[EB/OL]. 2023: arXiv: 2311.05553. https://arxiv.org/abs/2311.05553.

[65] Li J N, Li D X, Xiong C M, et al. BLIP: Bootstrapping Language-Image Pre-Training for Unified Vision-Language Understanding and Generation[EB/OL]. 2022: arXiv: 2201.12086. https://arxiv.org/abs/2201.12086.

[66] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, at el. Learning Transferable Visual Models from Natural Language Supervision[C]. *Proceedings of the 38th International Conference on Machine Learning*, 2021: 8748-8763.

[67] Haotian Liu, Chunyuan Li, Qingyang Wu, Yong Jae Lee. Visual Instruction Tuning[C]. *Advances in Neural Information Processing Systems*, 2023: 34892-34916.

[68] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, Mohamed Elhoseiny. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models[EB/OL]. 2023. arXiv preprint arXiv:2103.00020.

[69] Yinpeng Dong, Huanran Chen, Jiawei Chen. How Robust is Google's Bard to Adversarial Image Attacks[EB/OL]. 2023. arXiv preprint arXiv:2 309.11751.

[70] Chen H R, Dong Y P, Wang Z Y, et al. Robust Classification *via* a Single Diffusion Model[EB/OL]. 2023: arXiv: 2305.15241. https://arxiv.org/abs/2305.15241.

[71] Luo W D, Ma S Y, Liu X G, et al. JailBreakV: A Benchmark for Assessing the Robustness of MultiModal Large Language Models Against Jailbreak Attacks[EB/OL]. 2024: arXiv: 2404.03027. https://arxiv.org/abs/2404.03027.

宫润森 于 2022 年在东北大学计算机科学与技术专业获得硕士学位。现在中国科学院大学信息安全专业攻读博士学位。研究领域为信息安全、人工智能。研究兴趣包括: 信息安全、网络安全、互联网基础资源、人工智能、AI 安全。

王凯 于 2017 年在中国科学院大学信息安全专业获得博士学位。现供职于前沿科技企业。研究领域为信息安全、AI 安全, 研究兴趣包括: 大模型安全、大模型赋能安全等。

张昱霖 于 2021 年在北京理工大学, 计算机技术专业获得硕士学位, 现在西安电子科技大学攻读博士学位。

张伟哲 男, 黑龙江哈尔滨人, 博士, 哈尔滨工业大学教授、博导, 入选国家级人才计划。主要研究方向是网络空间安全、高性能计算、嵌入式计算和云计算。发表期刊和会议论文约 290 篇, 著作 3 本。担任 IEEE Transactions on Cloud Computing 的编委。ACM 高级会员, IEEE 高级会员, CCF 杰出会员。

乔延臣 男, 山东聊城人, 博士, 鹏城实验室副研究员、博士生导师, 主要研究方向为网络空间安全、互联网体系结构等。

张玉清 中国科学院大学计算机科学与技术学院教授、博导, 国家计算机网络入侵防范中心主任。研究领域为网络攻防与系统安全, 大数据与智能安全, 物联网系统安全等。