|  |  |  |
| --- | --- | --- |
| 技术交底书  （以下由知识产权部填写）  案号：  日期： | 申请人：北大深圳研究生院  深圳市南山区西丽大学城北大园区  邮编：518055，总机：0755－26035327 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 交底书名称 | 一种基于反思机制的大语言模型代码安全生成的即插式安全增强方法与系统  →基于大模型自身反思机制的即插式安全增强方法与系统 | | | |
| 发明人 | 李挥 刘傲凡 王滨 杨傲 李恪聃 | | | |
|  |  |  | （以下由知识产权部填写） | |
| 交底书撰写人 | 刘傲凡 |  | IPR负责人 |  |
| 撰写人办公电话 |  |  | |
| 撰写人手机号码 （急案必须填写） | 13418788132 | 流程助理 |  |
| 撰写人 E-mail | af.liu@stu.pku.edu.cn |  | |

|  |  |
| --- | --- |
| 撰写指导意见  （IPR填写） | *比如：对于可能存在单一性问题的多个发明希望在一篇申请文件中提交时，提醒代理人可以不解决单一性问题；对于有特别事项需要提醒代理人时，要求其在作业前和专利工程师电话联系等等* |

缩略语和关键术语定义

LLM Large Language Model，大型语言模型是一种基于深度学习技术的人工智能模型，具有规模庞大的参数量以及强大的自然语言处理能力。

Transformer 由论文《Attention is All You Need》提出的基于Self-Attention机制的机器学习模型。

Self-reflection 自我反思，是指大型语言模型具有的一种评估并改善自身生成的输出的能力，旨在提升模型可靠性。

## 1．相关技术背景以及最接近的现有技术

### 1.1 背景技术：用于方便读者理解本发明的技术背景

随着人工智能技术的不断发展，大型语言模型（Large Language Models, LLMs），如ChatGPT、Claude等，已广泛应用于推理、编程、科学研究等多个领域，并在许多实际应用中取得了显著成果。这些模型不仅能够生成高质量的自然语言文本，还能辅助编程，提供代码生成、调试、优化等服务。在编程领域，LLMs被用于自动生成代码、优化算法和辅助开发人员提高编程效率。然而，尽管这些技术展现了巨大的潜力，仍然存在一些重要的安全隐患和问题，特别是在生成的代码中可能隐藏着严重的安全漏洞。

在传统的软件开发过程中，安全问题通常是在代码编写完成后，通过静态分析、动态测试等手段进行检测和修复。然而，当前的LLMs技术并未充分考虑到代码的安全性，尤其是在生成复杂程序时，这些模型生成的代码往往缺乏必要的安全审查。例如，某些模型可能会生成包含SQL注入漏洞的代码，导致攻击者能够通过恶意输入操控数据库。又如，生成的代码中可能存在未验证的用户输入，进而引发跨站脚本攻击（XSS）等安全漏洞。2024年11月，一位用户通过ChatGPT生成的代码搭建了一个辅助交易机器人，但模型生成的代码调用了一个恶意的API，并直接将用户的私钥传递给该API，导致用户钱包被盗取了2.5k美元。这一事件突显了在生成过程中未审查和过滤恶意代码带来的潜在安全威胁。

更严重的是，现有的LLMs模型在生成代码时并不会主动评估其安全性，导致生成的代码在实际运行时可能引发无法预测的安全漏洞。以SQL注入为例，LLMs可能生成一个包含数据库查询的代码片段，但如果未正确处理用户输入，恶意用户通过输入SQL语句可能导致数据库被攻击和数据泄露。此外，随着越来越多的企业和组织将LLMs应用于实际工作流程，如何确保大语言模型在生成代码和其他内容时不包含潜在的安全风险和漏洞，已经成为一个亟待解决的问题。

现有的安全检测手段大多集中在代码生成之后的审查环节，例如使用静态代码分析工具、Fuzz测试等方法，但这些方法往往存在较高的成本和较长的时间延迟。在快速发展的开发环境中，这种事后修复的方法不能有效地应对大规模自动化编程场景中的安全需求。因此，如何在生成过程中实现对代码的实时安全评估，成为了提升LLMs应用安全性的关键技术挑战之一。

针对上述问题，本发明提出了一种基于反思机制的代码安全增强方法与系统，通过在代码生成过程中引入安全性自检机制，实时监控和修复生成代码中的潜在安全问题。具体而言，该方法通过引入反思机制，使得LLMs在生成代码时能够根据预设的安全规则，对生成的代码进行自我验证和修正。在生成过程中，模型不仅会评估代码的功能性，还会评估其安全性，及时发现并修复潜在的安全漏洞，从而减少后期修复的时间和成本。

例如，当用户请求生成一段包含数据库操作的代码时，反思机制会检查代码中是否存在潜在的SQL注入风险。如果检测到潜在的漏洞，模型将自动调整代码，使用参数化查询等安全手段替代不安全的字符串拼接方式，避免SQL注入攻击。又如，当生成包含用户输入处理的代码时，反思机制会检查是否对用户输入进行了严格的验证和消毒，以防止跨站脚本攻击（XSS）等问题的发生。

此外，反思机制还可以针对模型生成的代码进行自我优化。例如，在生成涉及敏感信息（如API密钥、私钥等）的代码时，模型会自动检测是否存在将敏感信息暴露的风险，并提示用户进行加密或采用环境变量存储等更安全的方式。在此过程中，反思机制不仅提高了生成代码的安全性，还增强了代码的鲁棒性和稳定性。

为了进一步提高生成过程中的安全性，本发明还提出了一种结合外部知识库的安全知识增强方法。通过引入安全知识库，LLMs可以在生成代码时实时调用外部安全知识，以进一步加强生成代码的安全性。例如，在生成涉及加密操作的代码时，模型可以参考外部安全知识库中的最佳实践，确保生成的代码符合当前的加密标准，避免常见的加密漏洞。

然而，使用外部知识库时，也可能面临噪声和不相关信息的引入问题。为了解决这一问题，本发明提出了知识过滤和浓缩方法，通过对外部安全知识进行过滤和精炼，确保只将最相关的安全信息引入生成过程，从而避免引导模型产生错误的或不相关的安全建议。

通过引入反思机制与安全知识增强，本发明能够在代码生成过程中实现实时的安全审查和修复，从而降低代码中的安全风险，提升代码生成的可靠性和鲁棒性。这一方法不仅能够有效减少后期安全漏洞修复的成本，还能提升自动化编程的整体安全性，为大语言模型在更多实际应用中的推广和应用提供了技术保障。

### 1.2 与本发明相关的现有技术一

Rui Zhang等人提出的两方博弈对齐框架在现有方法的基础上进行了显著的扩展和创新。传统的对齐方法，如基于人类反馈的强化学习（RLHF），虽然在实践中被广泛应用，但其局限性在于难以全面覆盖所有任务类型，并且容易因过度依赖预收集的提示而导致模型性能饱和。偏好学习方法和直接从偏好数据中学习的方法试图减少对人类反馈的依赖，但在实际应用中仍面临标注数据质量和动态调整能力的挑战。

相比之下，本文提出的两方博弈框架通过引入对抗防御博弈机制，提供了一个更为动态和自适应的对齐环境。该方法的核心在于通过迭代交互来识别和利用潜在弱点，从而在多个方面超越了传统方法。首先，通过对抗方和防御方的互动，模型能够在不断变化的环境中持续学习和改进，显著提高了泛化能力。其次，引入多样性约束确保生成的攻击提示多样化，这不仅增强了模型的鲁棒性，还促进了更广泛的攻击防御策略的发展。

在理论上，本文方法通过证明收敛到纳什均衡，提供了算法的稳定性保障。这一理论基础使得该方法在实践中更具可靠性和可预测性。然而，尽管本文方法在提高模型安全性和泛化能力方面具有显著优势，但也面临着一些挑战。计算复杂度高，需要多次迭代优化，依赖高性能计算资源，这对硬件设施提出了较高的要求。此外，实现难度较大，依赖高质量的数据集和标注，这在数据获取和处理方面增加了复杂性。

Zheng, R., Guo, H., Liu, Z., Zhang, X., Yao, Y., Xu, X., ... & Liu, Y. (2024). Toward Optimal LLM Alignments Using Two-Player Games. arXiv preprint arXiv:2406.10977.

### 1.2 与本发明相关的现有技术二

Chen等人提出了SecAlign，这是一种基于偏好优化技术的防御方法，旨在保护大型语言模型（LLMs）免受提示注入攻击。SecAlign通过构建一个包含提示注入输入、安全输出（响应合法指令的输出）和不安全输出（响应注入的输出）的偏好数据集，并在该数据集上进行偏好优化，教导LLM优先选择安全输出而非不安全输出。这种方法首次将各种提示注入的成功率降低到约0%，即使面对比训练期间所见攻击更为复杂的攻击，也显示出良好的泛化能力。此外，经过防御训练的模型在实用性方面与防御训练前相当，具有相似的效用。

SecAlign的主要优点在于其简单性、对模型效用的保持以及对未知攻击的强大安全性，甚至能够抵御基于优化的攻击。通过偏好优化，SecAlign还建立了LLM安全与对齐之间的联系，这两个主题之前是分开研究的。然而，SecAlign也存在一些局限性。首先，它仅适用于指令部分和数据部分有明确分隔（例如通过分隔符）的场景。其次，SecAlign构建的期望输出与其他基于微调的防御方法共享一个缺点：期望输出忽略了数据中的注入指令，而不是将其作为数据的一部分进行处理。这可能导致LLM忽略数据中的一些命令句（这些命令句可能并非注入，而应作为数据处理，例如需要翻译的命令句）。尽管在本研究中未观察到这种现象或效用受损，但解决此问题需要根据良性指令仔细选择注入，并生成一个特别的期望响应，而不是来自SFT数据集的响应。此外，作为一种AI系统的防御方法，SecAlign还不能实现100%的安全性。对于LLM集成应用中的更强安全性，可能需要一个多层级的防御体系，将SecAlign与其他技术（如检测和输入重新格式化）结合起来。最后，SecAlign目前还不能防御提示注入之外的攻击，例如越狱攻击和数据提取。

Chen, S., Zharmagambetov, A., Mahloujifar, S., Chaudhuri, K., & Guo, C. (2024). Aligning llms to be robust against prompt injection. arXiv preprint arXiv:2410.05451.

## 2．本发明技术方案的详细阐述（发明内容）

### 2.1 本发明所要解决的技术问题

大多数现有方法主要集中在对特定模型的微调和安全对齐，或者侧重于输入输出端的内容过滤机制。这些方法虽然在一定程度上解决了安全问题，但往往存在一些限制和不足。例如，微调和安全对齐的方法需要大量的标注数据和训练资源，而输入输出端的过滤机制则依赖于规则和模板，难以适应复杂多变的生成任务。更为关键的是，这些方法通常是特定于某种模型或任务的，无法在不同模型之间通用，且存在一定的局限性，无法满足对大规模、实时、自动化生成的安全要求。

针对这些问题，本发明提出了一种通用的、无需微调训练的安全增强方案，旨在通过大模型自身的任务生成信息来反思和强化其安全性。具体而言，本发明通过引入一种反思机制，使得大模型能够在生成过程中自动识别和修复其中潜在的安全漏洞，并通过自我总结和凝练生成安全相关的知识。与传统的依赖外部安全知识库或人工干预的方案不同，本发明的安全机制基于模型内部的生成过程，能够实时识别安全问题，生成安全编码、安全规范、安全知识等，形成大模型自生成的安全知识库，并在后续生成任务中加以应用。

这种安全机制不依赖于外部的知识库，因此避免了外部信息的不相关性或噪声对模型的干扰，同时也减少了对外部安全资源的依赖，具有更高的独立性和灵活性。通过对模型生成内容的深度反思和知识凝练，本发明能够实现大模型在生成过程中自我优化，提升生成任务的安全性。这一安全方案的核心优势在于其“开箱即用”的能力，可以适用于各种大模型，无需专门的微调或训练，从而大大简化了安全防护的实施过程，并提升了大模型应用的普遍适用性和安全性。

在实际应用中，尤其是在政府和企业等对数据安全要求极高的领域，这种通用的安全机制显得尤为重要。大模型生成的内容，如文本、代码等，往往涉及到敏感数据和信息，如果缺乏足够的安全防护，可能会引发严重的安全隐患。例如，生成的代码可能包含未加密的敏感信息或存在未验证的用户输入，导致数据泄露或攻击风险。因此，能够对大模型生成的内容进行实时的安全审查和修复，确保其不含有潜在的风险和安全漏洞，将对保护数据安全、提升大模型应用的可信度和可靠性具有重要意义。

### 2.2 本发明技术实现的具体实施

#### 2.2.1 反思机制的设计

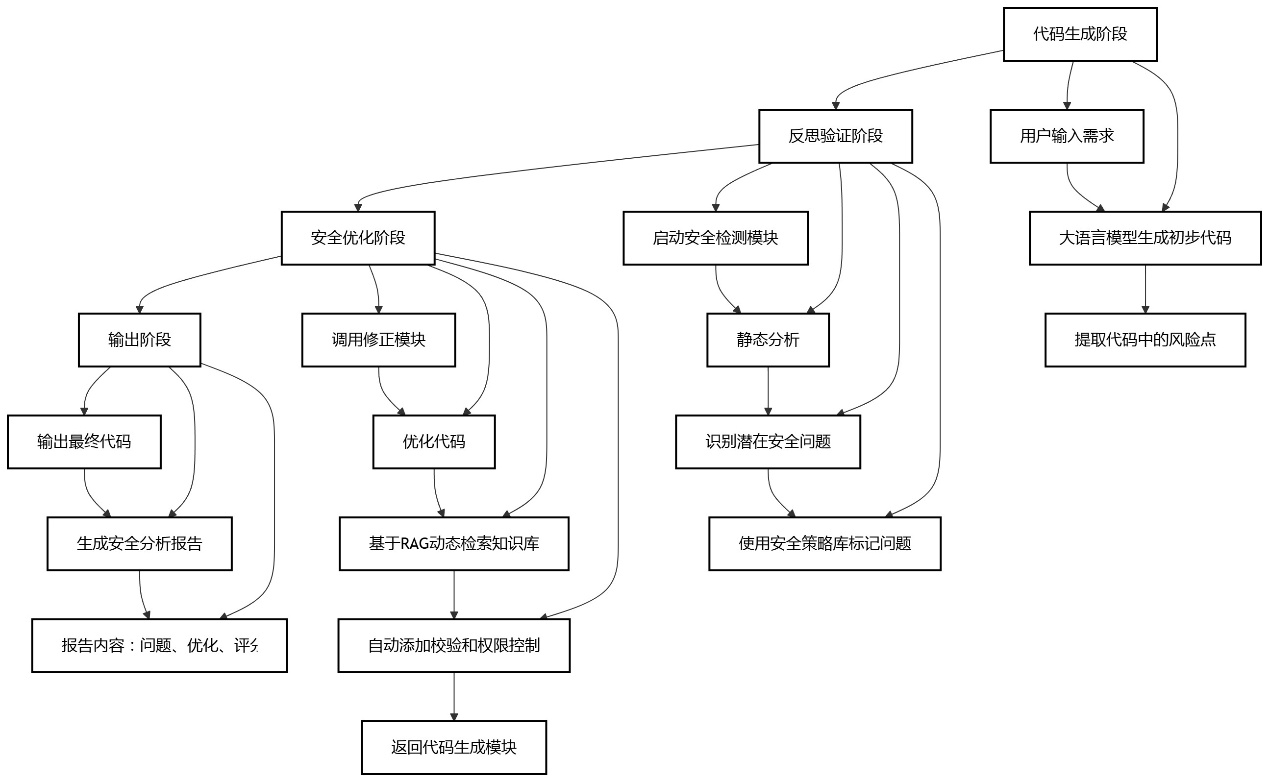
本发明提出了一种基于大模型自身反思机制和自生成安全知识库的安全增强技术，能够在不依赖外部知识库的情况下，通过大模型自身的生成任务和反思机制，提高生成内容的安全性。以下是具体实施的详细描述。

**反思机制的设计**

反思机制是本发明的核心，目的是让大模型能够在生成内容时进行实时的安全审查与优化。在生成任务开始时，大模型根据用户输入生成输出内容，可能是文本、代码等形式。生成后的内容会进入一个安全审查环节，这一环节对生成的内容进行全面评估，识别潜在的安全隐患。

例如，在生成代码时，模型可能会发现生成的代码中包含敏感数据（如API密钥、私密信息等）或不符合安全规范的操作。此时，反思机制会启动，分析生成内容中可能的风险源，进而对其进行修正。这一过程能够帮助大模型在生成内容时，自动避免泄露敏感信息、绕过安全控制等潜在的安全漏洞。

如果在第一次生成后，系统发现内容有问题，模型会进一步思考并修正输出内容，确保其符合安全标准。例如，若发现生成的代码存在未加密的密码信息，反思机制会自动加密密码或者删除敏感信息，修正后的内容会再次提交给安全审查模块进行评估。



具体实施包括：

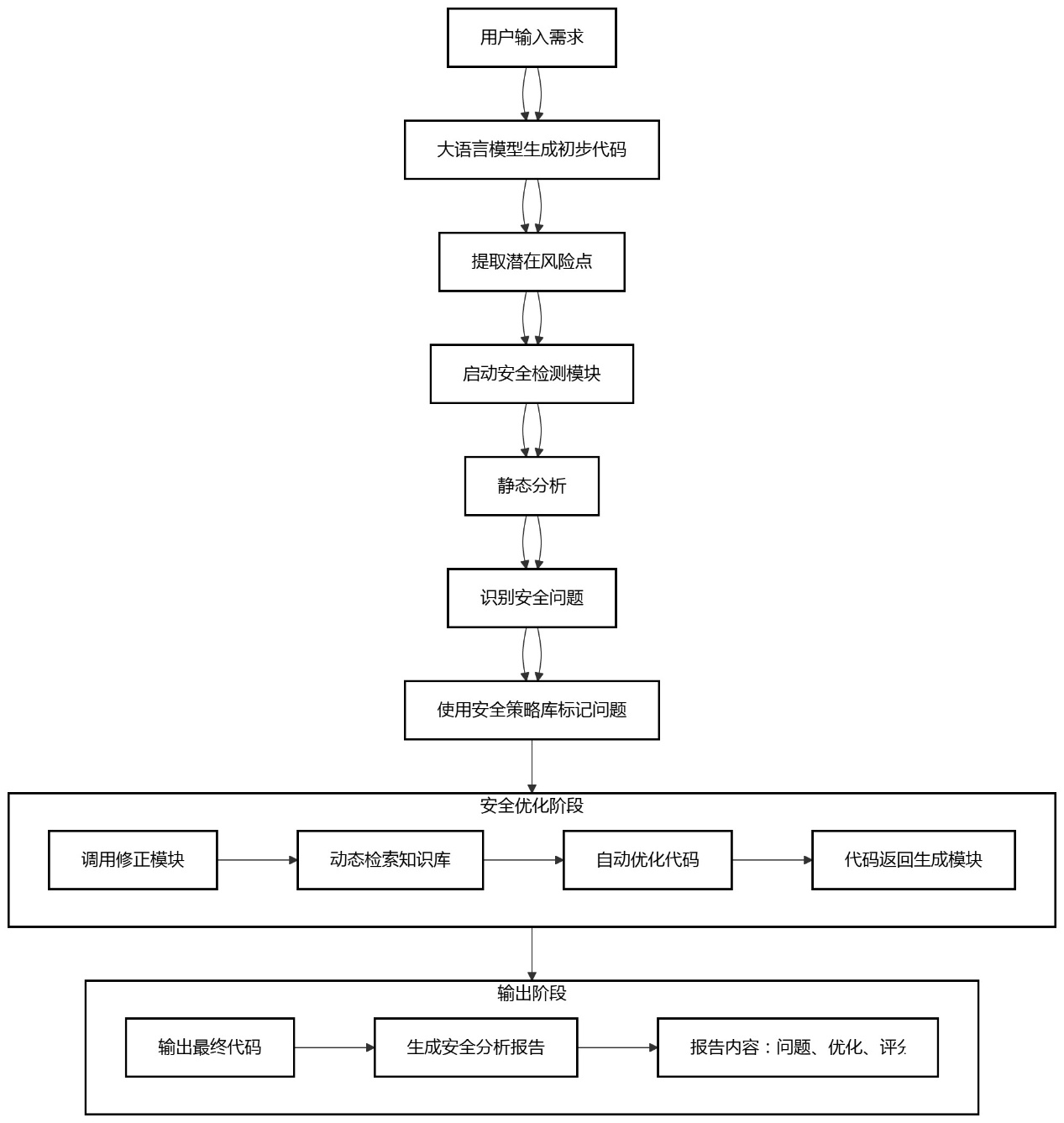
1. 安全知识提取：在生成任务的反思过程中，模型会提取出与安全相关的知识。例如，针对代码生成任务，模型可能会识别出某些代码模式存在安全漏洞，或者某些常见的攻击方式（如SQL注入、XSS攻击等）可能导致生成的代码存在风险。模型会将这些知识进行提炼，并存入安全知识库。
2. 知识库更新：随着模型不断生成新的内容并进行反思，安全知识库中的知识将不断得到更新和扩展。每次生成任务后，模型会分析生成的内容，识别出新出现的安全隐患，并将相应的安全规则、编码规范等信息加入到安全知识库中，从而保持其更新性和时效性。
3. 安全知识应用：在未来的生成任务中，模型可以调用安全知识库中的内容，快速识别和避免潜在的安全问题。例如，在生成一段API调用代码时，模型可以参考安全知识库中的规范，确保生成的代码没有泄露敏感信息或包含潜在的攻击漏洞。

**安全知识库的自生成与扩展**

在每次生成任务后，大模型会将生成过程中积累的安全知识进行总结，形成自生成的安全知识库。该知识库包含了关于不同类型的安全漏洞、编码规范和安全规则等内容。这一知识库并非静态，而是随着模型生成任务的不断进行而扩展和更新。每当模型在生成任务中发现新的潜在风险或新的安全知识，它都会自动将其加入到知识库中。

这种自生成的安全知识库具有高度的可扩展性和自适应能力。当模型进行新的生成任务时，它可以从知识库中提取出相关的安全知识，并将其应用到当前任务中。比如，在生成API调用代码时，模型可以参考安全知识库中的规范，确保生成的代码不包含可能的攻击面或不符合安全要求的代码模式。

该知识库的最大优势在于它不依赖外部资源，完全由大模型自身生成和维护。这使得大模型能够在没有外部干预的情况下，独立扩展其安全能力，持续增强生成任务中的安全性。



**安全性检测与优化过程**

在每一次生成任务中，安全性检测和优化过程是必不可少的环节。生成内容初步完成后，模型将对其进行安全性评估。这一评估不仅包括检查内容是否存在敏感信息泄露、恶意代码、隐私侵犯等问题，还会对内容是否符合行业安全规范进行全面审查。

一旦检测到潜在的安全问题，系统会启动自动化修正和优化机制。例如，如果生成的代码包含不安全的函数调用或硬编码的密码，系统会自动进行替换或删除，确保生成内容符合安全标准。这一过程不仅提高了生成内容的安全性，还增强了模型对潜在风险的敏感度，使其能够在实时生成任务中进行安全优化。

在优化过程中，系统还会根据用户的反馈进行调整。如果用户对生成内容中的安全性提出异议或发现新的安全隐患，模型会再次对内容进行审查，确保输出结果符合最高的安全要求。通过这种反馈机制，模型的安全性得到了持续改进，保障了最终输出的内容始终处于安全的状态。

## 2.3 本发明技术方案带来的有益效果

本发明技术方案提出了一种大模型自我反思的安全增强的方法，能够给各类大模型带来通用的，无需微调训练的，能够自我优化的安全增益效果，实现开箱即用的能力。

**无需外部知识库的自生成能力**

与传统依赖外部知识库的安全防护机制不同，本发明的安全方案不依赖于任何外部安全知识库。大模型通过自身的生成任务和反思机制，能够自动生成和积累安全知识，形成一个完整且可扩展的安全知识体系。每次任务结束后，模型会通过自我学习将新的安全问题和解决方案总结并存储在自生成的知识库中。

这种自生成的能力意味着大模型不需要额外的外部资源即可保持安全性，且随着模型使用时间的增加，它能够不断学习和适应新的安全挑战。这不仅为大模型的广泛应用提供了安全保障，还为需要高安全性的应用场景（如金融、医疗、政府等）提供了强有力的技术支持。

**基于反思机制的自动评估与改进**

通过本发明的具体实施，大模型能够在生成内容的过程中自动评估和修正安全隐患，确保输出的内容符合安全要求。这种方法不仅提高了生成任务中的安全性，还增强了大模型在实际应用中的安全性和可靠性。

尤其是在没有外部知识库的支持下，模型依然能够通过自身的反思和学习生成安全知识，适应不同任务中的安全需求。通过这种自生成和自优化的安全机制，本发明能够显著降低大模型在实际应用中出现安全漏洞的风险，特别是在需要高度保障数据安全的行业领域中，具有重要的应用价值。

## 2.4 本发明的技术关键点和欲保护点是什么

1. 本发明的核心技术之一是大模型的反思机制。该机制通过对大模型生成内容的自我审查和反思，使其能够主动识别生成内容中的潜在安全风险，并进行实时修正。反思机制并不是在生成任务后进行简单的错误修正，而是在生成内容的每一个阶段进行持续的安全评估，确保每一次生成输出都符合安全标准。

2. 大模型通过自生成安全知识库，持续积累和扩展安全知识。这一知识库不仅涵盖了常见的安全漏洞、编码规范和安全规则，还能够随着大模型生成任务的不断进行，自动从生成任务中总结出新的安全知识，形成可持续更新的安全知识体系。

3. 在生成过程中加入自动化的安全检测与修正机制，确保大模型能够实时识别并修复输出中的安全问题。这一机制可以在生成内容后对其进行全面检查，及时发现内容中的敏感数据泄露、恶意代码、隐私侵犯等风险点，并采取相应措施进行修正。

上述均为本发明技术关键点，同时也是欲保护点。