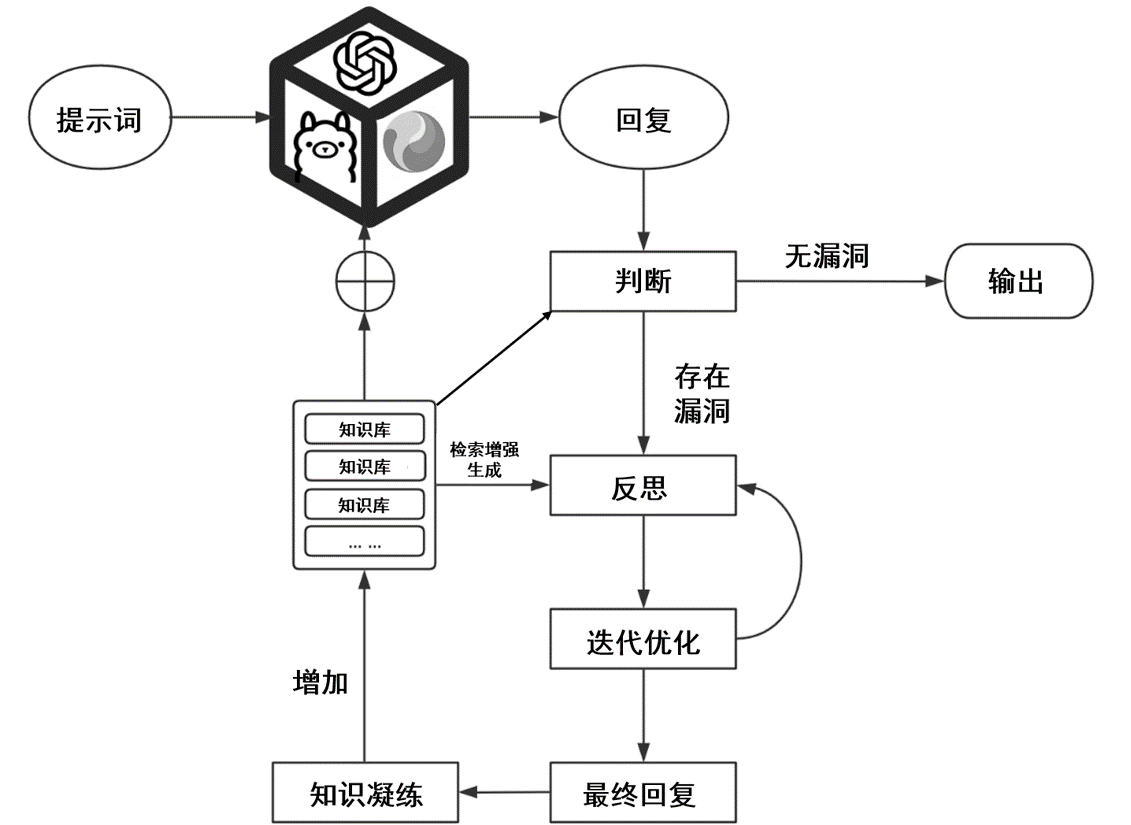
说明书摘要

本发明提供一种大模型自我反思的安全知识凝练的方法与系统，包括：步骤S1，生成初始输出；步骤S2，进行自我反思，并判断输出是否存在潜在的缺陷或安全漏洞，若是则进行迭代优化；步骤S3，在迭代优化过程中，先基于自我反思判断出的缺陷或安全漏洞，以及附加的安全规范，提出改进措施并进行优化改进；然后再次进行自我反思，不断重复直到满足安全要求；步骤S4，将最终回复和任务流程投入至安全知识凝练机制，凝练安全规范的安全知识和安全概念，并采用统一的预设格式存储于自生成的安全知识库。本发明能够有效地提升安全知识凝练的通用性，还能够提高大模型生成的思考深度，可扩展地将安全知识用于生成过程，满足安全要求，且不依赖于外部知识库。

摘要附图



权利要求书

1、一种大模型自我反思的安全知识凝练的方法，其特征在于，包括以下步骤：

步骤S1，接收用户输入的提示词，通过大语言模型生成初始输出；

步骤S2，通过大语言模型进行自我反思，并判断输出是否存在潜在的缺陷或安全漏洞，若是则跳转至步骤S3进行迭代优化，若否则显示输出结果；

步骤S3，在迭代优化过程中，先基于自我反思判断出的缺陷或安全漏洞，以及在输出后面附加的安全规范，通过大语言模型提出改进措施并进行优化改进；然后，在优化改进之后再次进行自我反思，以此方式不断重复自我反思和迭代优化过程，直到优化改进后的回复满足所有的安全要求，以此回复作为最终回复，并跳转至步骤S4；

步骤S4，将最终回复和任务流程投入至安全知识凝练机制，凝练出安全规范的安全知识和安全概念，并采用统一的预设格式存储于自生成的安全知识库。

2、根据权利要求1所述的大模型自我反思的安全知识凝练的方法，其特征在于，所述步骤S2中，在判断输出是否存在潜在的缺陷或安全漏洞时，判断当前的安全知识库中是否已经存在有安全知识和安全规范，若是，则使用所述知识库对输出进行审查；若否，则通过大语言模型进行判断。

3、根据权利要求1或2所述的大模型自我反思的安全知识凝练的方法，其特征在于，所述步骤S2进行自我反思的实现过程为：利用输出和任务的输入进行检索增强生成RAG查询，获取相关的安全知识，相关的安全知识包括针对安全编码的安全规范、针对敏感问题的安全回复规范以及历史记录中的标准回复；当查询到适用的安全知识时，将对应的安全知识与输出进行拼接。

4、根据权利要求3所述的大模型自我反思的安全知识凝练的方法，其特征在于，将对应的安全知识与输出进行拼接的过程为：在查询到适用的安全知识后，在输出后面附加上该安全知识对应的安全规范，并跳转至步骤S3进行迭代优化。

5、根据权利要求3所述的大模型自我反思的安全知识凝练的方法，其特征在于，所述步骤S3中，若已查询到适用的安全知识，则要求大语言模型根据对应的安全规范对初始回复进行优化改进，并在优化改进之后再次进行自我反思。

6、根据权利要求1或2所述的大模型自我反思的安全知识凝练的方法，其特征在于，所述步骤S4采用Milvus向量数据库存储自生成的安全知识，存储的预设格式为：ID；DATE；CHUNKS[“INDEX”, “CHUNK\_CONTENT”]，其中，ID表示唯一标识符，用于区分知识条目；DATE表示记录生成或更新的日期；CHUNKS表示知识片段列表；INDEX表示知识片段的索引编号；CHUNK\_CONTENT表示知识片段的内容。

7、根据权利要求1或2所述的大模型自我反思的安全知识凝练的方法，其特征在于，所述步骤S4包括以下子步骤：

步骤S401，对于满足安全要求的最终回复，以及对应的上下文中的文本生成任务，根据任务的背景，对比具有缺陷的初始回复，指出在自我反思和迭代优化过程中，实现存在缺陷或安全漏洞转变为符合安全规范所对应的安全表述，以完成安全表述层级的凝练；

步骤S402，对于已经识别出的安全表述的转变，以及对这种转变的原因进行深入反思，将反思过程的全部思考内容使用思维链COT的形式展现出来，并将这些思考内容重新送到大语言模型，使用精简、本质化和无噪声的方式进行总结，从中提取安全知识，获取安全表述的深层次原因，并存储至自生成的安全知识库，以完成安全知识层级的凝练；

步骤S403，对于提取所获得的安全知识，提取该安全知识的安全概念，所述安全概念指的是安全知识所涉及的核心点，包括例如strcpy()等风险函数、溢出漏洞、边界审查编码规范以及有毒物品中的一条或多条；当提取到任意一条安全概念时，将另外记录一条安全概念及其对应的释义，用于作为后续检索增强生成RAG查询相关的核心概念，以完成安全概念层级的凝练，加快自我反思和检索增强生成RAG查询的速度。

8、根据权利要求7所述的大模型自我反思的安全知识凝练的方法，其特征在于，所述步骤S401中，所述安全表述包括实现安全代码生成所对应的修复代码，和/或实现符合安全规范所对应的修改的名词或句式表述。

9、根据权利要求7所述的大模型自我反思的安全知识凝练的方法，其特征在于，所述步骤S402中，使用精简、本质化和无噪声的方式进行总结的过程包括：在代码输出前针对溢出漏洞进行审查并注意使用strcpy()等风险函数的情况；或，对于strcpy()风险函数使用strcpy\_s()安全函数进行代替这种安全修复；或，生成内容中若涉及有毒物品的配置流程信息不予回答。

10、一种大模型自我反思的安全知识凝练的系统，其特征在于，采用了如权利要求1至9任意一项所述的大模型自我反思的安全知识凝练的方法，并包括：

初始回复生成模块，接收用户输入的提示词，通过大语言模型生成初始输出；

自我反思模块，通过大语言模型进行自我反思，并判断输出是否存在潜在的缺陷或安全漏洞，若是则跳转至迭代优化模块进行迭代优化，若否则显示输出结果；

迭代优化模块，在迭代优化过程中，先基于自我反思判断出的缺陷或安全漏洞，以及在输出后面附加的安全规范，通过大语言模型提出改进措施并进行优化改进；然后，在优化改进之后再次进行自我反思，以此方式不断重复自我反思和迭代优化过程，直到优化改进后的回复满足所有的安全要求，以此回复作为最终回复，并跳转至安全知识凝练模块；

安全知识凝练模块，将最终回复和任务流程投入至安全知识凝练机制，凝练出安全规范的安全知识和安全概念，并采用统一的预设格式存储于自生成的安全知识库。

**一种大模型自我反思的安全知识凝练的方法与系统**

说明书

**技术领域**

本发明涉及一种大语言模型的安全知识凝练方案，尤其涉及一种大模型自我反思的安全知识凝练的方法，并进一步涉及采用了该大模型自我反思的安全知识凝练的方法的系统。

**背景技术**

当前大型语言模型（Large Language Models, LLMs），简称大语言模型或大模型，例如ChatGPT和Claude，在推理、编程和科学研究等各个领域都显示出巨大的潜力。LLMs在各种应用场景中得到广泛采用。然而，LLMs并不总是可靠的。它们会产生有毒或不安全的内容，并且易受到“幻觉”（Hallucination）的影响，从而导致错误或不安全的输出。而在政府和企业等人工智能应用场景中，数据安全是至关重要的，尤其是采用大模型输出内容参与工作流程的情况下，如何保证大模型生成的文本和代码等内容不含有风险内容和安全漏洞，成为大模型应用的关键技术问题之一。

2024 年 6 月，一名进入企业进行实习的高校博士田某，利用 Hugging Face（HF）的漏洞，在其实习公司的共享模型里写入破坏代码。导致模型的训练效果忽高忽低，但是所在团队无法核查原因，影响了团队的模型训练任务。后来，其实习公司起诉该实习生，请求法院判令其赔偿公司侵权损失，并公开赔礼道歉。

2024年11月，发生了大模型“语料污染”第一真实案例，一名用户通过ChatGPT编程来构建一个辅助交易机器人，但是在ChatGPT生成的代码调用了一个恶意的API地址，且直接把私钥明文提供给恶意API处理，代码运行后用户钱包被盗取$2.5k的损失。可见当模型采用搜索方式进行输出时未对搜索结果进行审查，使用恶意的参考资料可能会导致生成恶意内容。

随着大模型应用的发展，可以通过多种防护措施对大模型的输出和推理过程进行防护，包括：

1、在用户输入侧对用户的输入进行严格审查，以确保其合规性与安全性。当系统监测到用户输入中包含恶意内容时，应立即中止当前会话，避免潜在安全风险。

2、对大模型输出的内容进行持续的检测和审查，重点检测输出内容中是否存在违规内容、侵权内容和隐私泄露的问题，确保用户能获得安全可靠的合法信息。

3、在模型侧加强管理：监控用户与大模型交互的全过程，及时发现并阻止恶意用户的Prompt注入攻击，避免引导大模型产生违规内容。针对涉及敏感问题的情况，不回答或是使用安全回复，防止大模型误导用户或传播不合规信息。

4、针对幻觉问题，采取外部知识库，便于在生成时提高其准确性，或是采用人类专家参与审查的方式减少幻觉。

但是，现有技术中缺少轻量化、通用性强的，能够用于生成过程的RAG安全知识库，而使用外部知识库则可能会引入不相关的信息或噪声，进而误导大模型。并且大模型从冗长信息中获取知识的过程也会引入额外的开销，此时如何过滤浓缩获取的知识内容，尤为重要。

**发明内容**

本发明所要解决的技术问题是需要提供一种大模型自我反思的安全知识凝练的方法，通过采取大模型自生成的安全知识凝练技术方案对安全知识进行生成和凝练，使得在任何大模型上都能够实现基于自我反思而构建的安全机制，确保大模型输出的安全性，且通用性强；无需借助外部知识库，并具有自我迭代优化和可扩展的特点。在此基础上，还进一步提供采用了该大模型自我反思的安全知识凝练的方法的系统。

对此，本发明提供一种大模型自我反思的安全知识凝练的方法，包括以下步骤：

步骤S1，接收用户输入的提示词，通过大语言模型生成初始输出；

步骤S2，通过大语言模型进行自我反思，并判断输出是否存在潜在的缺陷或安全漏洞，若是则跳转至步骤S3进行迭代优化，若否则显示输出结果；

步骤S3，在迭代优化过程中，先基于自我反思判断出的缺陷或安全漏洞，以及在输出后面附加的安全规范，通过大语言模型提出改进措施并进行优化改进；然后，在优化改进之后再次进行自我反思，以此方式不断重复自我反思和迭代优化过程，直到优化改进后的回复满足所有的安全要求，以此回复作为最终回复，并跳转至步骤S4；

步骤S4，将最终回复和任务流程投入至安全知识凝练机制，凝练出安全规范的安全知识和安全概念，并采用统一的预设格式存储于自生成的安全知识库。

本发明的进一步改进在于，所述步骤S2中，在判断输出是否存在潜在的缺陷或安全漏洞时，判断当前的安全知识库中是否已经存在有安全知识和安全规范，若是，则使用所述知识库对输出进行审查；若否，则通过大语言模型进行判断。

本发明的进一步改进在于，所述步骤S2进行自我反思的实现过程为：利用输出和任务的输入进行检索增强生成RAG查询，获取相关的安全知识，相关的安全知识包括针对安全编码的安全规范、针对敏感问题的安全回复规范以及历史记录中的标准回复；当查询到适用的安全知识时，将对应的安全知识与输出进行拼接。

本发明的进一步改进在于，将对应的安全知识与输出进行拼接的过程为：在查询到适用的安全知识后，在输出后面附加上该安全知识对应的安全规范，并跳转至步骤S3进行迭代优化。

本发明的进一步改进在于，所述步骤S3中，若已查询到适用的安全知识，则要求大语言模型根据对应的安全规范对初始回复进行优化改进，并在优化改进之后再次进行自我反思。

本发明的进一步改进在于，所述步骤S4采用Milvus向量数据库存储自生成的安全知识，存储的预设格式为：ID；DATE；CHUNKS[“INDEX”, “CHUNK\_CONTENT”]，其中，ID表示唯一标识符，用于区分知识条目；DATE表示记录生成或更新的日期；CHUNKS表示知识片段列表；INDEX表示知识片段的索引编号；CHUNK\_CONTENT表示知识片段的内容。

本发明的进一步改进在于，所述步骤S4包括以下子步骤：

步骤S401，对于满足安全要求的最终回复，以及对应的上下文中的文本生成任务，根据任务的背景，对比具有缺陷的初始回复，指出在自我反思和迭代优化过程中，实现存在缺陷或安全漏洞转变为符合安全规范所对应的安全表述，以完成安全表述层级的凝练；

步骤S402，对于已经识别出的安全表述的转变，以及对这种转变的原因进行深入反思，将反思过程的全部思考内容使用思维链COT的形式展现出来，并将这些思考内容重新送到大语言模型，使用精简、本质化和无噪声的方式进行总结，从中提取安全知识，获取安全表述的深层次原因，并存储至自生成的安全知识库，以完成安全知识层级的凝练；

步骤S403，对于提取所获得的安全知识，提取该安全知识的安全概念，所述安全概念指的是安全知识所涉及的核心点，包括例如strcpy()等风险函数、溢出漏洞、边界审查编码规范以及有毒物品中的一条或多条；当提取到任意一条安全概念时，将另外记录一条安全概念及其对应的释义，用于作为后续检索增强生成RAG查询相关的核心概念，以完成安全概念层级的凝练，加快自我反思和检索增强生成RAG查询的速度。

本发明的进一步改进在于，所述步骤S401中，所述安全表述包括实现安全代码生成所对应的修复代码，和/或实现符合安全规范所对应的修改的名词或句式表述。

本发明的进一步改进在于，所述步骤S402中，使用精简、本质化和无噪声的方式进行总结的过程包括：在代码输出前针对溢出漏洞进行审查并注意使用strcpy()风险函数的情况；或，对于strcpy()风险函数使用strcpy\_s()安全函数进行代替这种安全修复；或，生成内容中若涉及有毒物品的配置流程信息不予回答。

本发明还提供一种大模型自我反思的安全知识凝练的系统，采用了如上所述的大模型自我反思的安全知识凝练的方法，并包括：

初始回复生成模块，接收用户输入的提示词，通过大语言模型生成初始输出；

自我反思模块，通过大语言模型进行自我反思，并判断输出是否存在潜在的缺陷或安全漏洞，若是则跳转至迭代优化模块进行迭代优化，若否则显示输出结果；

迭代优化模块，在迭代优化过程中，先基于自我反思判断出的缺陷或安全漏洞，以及在输出后面附加的安全规范，通过大语言模型提出改进措施并进行优化改进；然后，在优化改进之后再次进行自我反思，以此方式不断重复自我反思和迭代优化过程，直到优化改进后的回复满足所有的安全要求，以此回复作为最终回复，并跳转至安全知识凝练模块；

安全知识凝练模块，将最终回复和任务流程投入至安全知识凝练机制，凝练出安全规范的安全知识和安全概念，并采用统一的预设格式存储于自生成的安全知识库。

与现有技术相比，本发明的有益效果在于：在通过大语言模型生成初始输出之后，先进行自我反思，并判断输出是否存在潜在的缺陷或安全漏洞，若是则进行迭代优化；而在迭代优化过程中，先基于自我反思判断出的缺陷或安全漏洞，以及在输出后面附加的安全规范，通过大语言模型提出改进措施并进行优化改进；然后，在优化改进之后再次进行自我反思，以此方式不断重复自我反思和迭代优化过程，直到优化改进后的回复满足所有的安全要求；最后，将优化改进后的回复和任务流程投入至安全知识凝练机制，凝练出安全规范的安全知识和安全概念，并采用统一的预设格式存储于自生成的安全知识库，以此形成完整的自我反思机制。

因此，本发明能够有效地提升安全知识凝练的通用性，适用于各类大模型，且无需微调训练即可实现自我优化的安全增益效果，实现开箱即用的目的；在此基础上，还通过包括自我反思和迭代优化过程的自我反思机制提高大模型生成的思考深度，能够可扩展地将安全知识用于生成过程，确保大模型生成的文本和代码等内容能够满足安全要求，并将其中安全编码、安全规范、安全知识和安全概念等凝练出来，形成自生成的安全知识库；不依赖于外部知识库，有效地避免引入不相关的信息或噪声的风险，并具有自我迭代优化和可扩展的特点，为数据安全提供了更好的基础。

**附图说明**

图1是本发明一种实施例的自我反思机制流程示意图；

图2是本发明一种实施例的工作流程示意图；

图3是本发明一种实施例的安全知识凝练机制示意图；

图4是本发明的一种相关技术的系统架构图；

图5是本发明的另一种相关技术的系统架构图；

图6是本发明的另一种相关技术的模型架构图。

**具体实施方式**

在本发明的描述中，如果涉及到“若干”，其含义是一个以上；如果涉及到 “多个”，其含义是两个以上；如果涉及到“大于”、“小于”、“超过”，均应理解为不包括本数；如果涉及到“以上”、“以下”、“以内”，均应理解为包括本数。如果涉及到“第一”、“第二”等，应当理解为仅用于相同或是相似技术特征名称的区分，而不能理解为暗示/指明技术特征的相对重要性，不能理解为暗示/指明技术特征的数量，也不能理解为暗示/指明技术特征的先后关系。

在详细描述本发明的较优的实施例之前，先对相关的技术进行说明。其中，涉及的缩略语和关键术语定义包括：LLM，指的是Large Language Model，即大型语言模型，也称大语言模型或大模型，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。Knowledge Condensation知识凝练，大型语言模型所具备的能够将海量、繁杂的知识信息进行提炼、压缩与整合，形成更加结构化、高效且精准的知识表示的能力；RAG，指的是Retrieval-augmented Generation，即检索增强生成。

与本发明相关的一种技术方案是交互式反思方法。Ziwei Ji等人提出了一种交互式反思方法，该方法结合了知识获取和答案生成。通过反馈过程稳步提高了生成答案的事实性、一致性和内涵。利用LLMs的交互性和多任务处理能力，逐步产生更准确的答案。

这一技术方案聚焦在医疗领域，涉及不常见的专业概念和潜在的社会风险，使得幻觉带来的挑战尤为关键，因为不准确或误导性的信息会对患者护理造成严重后果。其系统设计了一个迭代的、内省的过程，它利用了 LLMs的多轮交互性和多任务能力。其中涉及到自我反思方法，首先为给定问题生成相关背景知识，然后进行事实性评估。一旦检测到差异，模型就会被敦促自我纠正，利用其固有的能力来完善知识。这个循环过程会重复下去，直到达到令人满意的事实水平。在回答阶段，采用类似的生成-分数-细化策略来确保生成的答案与背景知识之间的一致性。此外，在答案和问题之间进行蕴涵评估。如果生成的答案不符合标准，该过程返回到初始阶段，重复该循环。

具体来说，如图4所示，其系统包括三个循环：事实知识获取循环、知识一致性回答循环和问题蕴涵回答循环。

(1)事实知识获取循环：首先，模型根据提供的问题生成背景知识。此步骤利用了LLMs理解上下文的固有能力。然后，使用评分器对生成的知识进行事实性评估。如果事实性分数低于评估阶段设定的阈值，则会要求模型进行自我反思，要求它“请完善知识以提高其真实性”。

这种 generate-score-refine 策略以交互方式重复，直到生成的知识达到令人满意的事实性水平。此迭代过程促进了系统与其生成的知识之间的动态和迭代交互。并确保模型逐步完善生成的背景知识，将其与既定事实相结合。

(2)知识一致性回答循环：模型继续根据提供的问题和生成知识来生成答案。如果生成的答案的一致性分数低于阈值，则系统会提示模型进行内省、自我纠正，要求它“请优化回复以提高其一致性。”，重复此 generate-score-fine 策略，直到生成的答案达到一致。此迭代过程可确保模型根据经过审查的背景知识逐步细化生成的答案，从而保持其完整性。

(3)问题蕴涵回答循环：通过相似性分数来评估生成的答案的蕴涵关系，判断这个答案与整个任务所涉及的背景知识、预期语义等是否存在合理的蕴含关系，以此确保答案的合理性以及确实是可被认可的、能够有效回应问题的（即具备可回答性）。如果不符合相应蕴含要求，就得重新循环整个流程来优化答案。

这种交互式反思方法的主要工作是对医疗领域的生成式 QA场景进行减少幻觉的工作。这一技术方案工作在安全领域或是模型不具有专业知识的情况下，尤其是在复杂或不明确的应用场景下，仍处于早期阶段，尚未准备好直接实际部署。并且，这一技术方案完全依赖于大模型自身的判断和评分，对大模型自身能力要求极高，通用性和使用局限大，无法从历史记录中收获知识以提升能力。

与本发明相关的另一种技术方案是协同模型。如图5和图6所示，Dongze Hao等人提出了两个协同模型：知识浓缩模型和知识推理模型。首先，利用视觉语言模型的多模态感知和推理能力，从检索到的冗长段落中提炼出简洁的知识概念，确保与视觉内容和问题的相关性。其次，利用大型语言模型的文本理解能力，将段落总结并浓缩为有助于回答问题的知识本质。然后将这两种类型的浓缩知识集成到知识推理模型中，该模型明智地浏览合并的信息以得出结论性的答案。

工作聚焦于基于知识的视觉问答 （KB-VQA），要求模型利用外部知识来理解和回答基于视觉内容的问题。最近的研究从外部知识库中检索知识段落，然后使用它们来回答问题。但是，这些检索到的知识段落通常包含不相关或嘈杂的信息。例如将图像转换为视觉上下文，并将它们与问题和检索到的知识段落一起发送给LLM生成答案。由于检索到的知识段落包含许多嘈杂的信息，将会误导模型预测错误的答案。

协同模型所提出的方法由两个模型组成：知识浓缩模型和知识推理模型。

知识浓缩模型，采用 BLIP作为视觉语言模型，以及采用开源的大型语言模型Vicuna作为知识浓缩器，从检索到的知识中提取有用的信息。将图像首先输入图像编码器以提取视觉特征，再将视觉特征和文本嵌入输入到LLM以生成文本。将检索到的知识段落浓缩为知识概念和知识本质。

知识推理模型，在得到浓缩的知识概念和知识本质后，利用编码器-解码器架构对这些知识进行推理来预测答案。采用两种类型的知识推理方法来生成答案。对于串联知识模式，将视觉语境、问题、知识概念和本质等拼接成一句话，然后将句子输入到一系列编码器层中，共同编码这些文本信息。然后得到嵌入向量，传递给一系列解码器层以生成答案。对于串联嵌入模式，将视觉上下文和问题与不同类型的知识连接为不同的句子。这些句子被输入到一系列编码器层中，以分别编码不同的信息。然后将这些嵌入连接到一起并将它们传递给解码器。

这种协同模型的工作聚焦于基于知识的视觉问答（KB-VQA）场景，旨在通过浓缩保留更多的有效信息，提高模型的性能。但是仍然高度依赖于额外的知识库以及正确地获取知识，并且由于缺少模型的深入思考和自我反思，还可能出现知识浓缩模型将所有知识段落转换为无用的信息等情况。

因此，以上两种相关的技术方案，都不能满足输出安全性高且通用性强，以及无需外部知识库的实际应用需求。

下面结合附图，对本发明的较优的实施例作进一步的详细说明。

如图1和图2所示，本实施例提供一种大模型自我反思的安全知识凝练的方法，包括以下步骤：

步骤S1，接收用户输入的提示词，通过大语言模型生成初始输出；

步骤S2，通过大语言模型进行自我反思，并判断输出是否存在潜在的缺陷或安全漏洞，若是则跳转至步骤S3进行迭代优化，若否则显示输出结果；

步骤S3，在迭代优化过程中，先基于自我反思判断出的缺陷或安全漏洞，以及在输出后面附加的安全规范，通过大语言模型提出改进措施并进行优化改进；然后，在优化改进之后再次进行自我反思，以此方式不断重复自我反思和迭代优化过程，直到优化改进后的回复满足反思和迭代优化过程中，大模型所判断出来的所有安全要求，以此作为最终回复，并跳转至步骤S4；

步骤S4，将最终回复和任务流程投入至安全知识凝练机制，凝练出安全规范的安全知识和安全概念，并采用统一的预设格式存储于自生成的安全知识库。

需要说明的是，虽然现有技术也存在用于大模型应用安全场景的措施/方案，但目前的现有方案大多数措施聚焦于对特定模型的微调和安全对齐，或是输入输出侧的过滤机制。这种现有的方案无法针对各类大模型实现其通用性能。

与现有技术不同，本实施例提供的是对各类大模型通用的，无需微调训练的，并且能够自我优化的安全机制技术方案，实现开箱即用的能力。

本实施例在通过大语言模型生成初始输出之后，先进行自我反思，即通过大模型自身任务生成的信息，对其中的安全场景、安全漏洞和安全知识进行反思；并判断输出是否存在潜在的缺陷或安全漏洞，若是则进行迭代优化；而在迭代优化过程中，先基于自我反思判断出的缺陷或安全漏洞，以及在输出后面附加的安全规范，通过大语言模型提出改进措施并进行优化改进；然后，在优化改进之后再次进行自我反思，以此方式不断重复自我反思和迭代优化过程，直到优化改进后的回复满足所有指定的安全要求；所有指定的安全要求指的是在自我反思和迭代优化过程中大模型判断出的所有的安全要求。最后，将优化改进后的回复和任务流程投入至安全知识凝练机制，凝练出安全规范的安全知识和安全概念，将其中安全编码、安全规范、安全知识和安全概念等凝练出来，形成大模型自生成的安全知识库，该安全知识库也称RAG知识库或RAG安全知识库，可不断扩展，并采用统一的预设格式存储于自生成的安全知识库，以此形成完整的自我反思机制。在之后的生成任务中采用其中的安全知识进行附加，从而提高大模型输出的安全性。

为了进一步提升大模型应用的安全性，本实施例通过在生成过程中添加安全知识，并采用自我反思机制进一步增强生成内容的可靠性和鲁棒性，对反思生成的信息内容进行凝练总结。并且，通过采取大模型自生成的安全知识凝练方法对安全知识进行生成，使得在任何大模型上都能够实现自身思考构建的安全机制，而无需借助外部知识库，因此，本实施例能够有效地提升安全知识凝练的通用性，适用于各类大模型，且无需微调训练即可实现自我优化的安全增益效果，实现开箱即用的目的。

在此基础上，本实施例还通过包括自我反思和迭代优化过程的自我反思机制提高大模型生成的思考深度，能够可扩展地将安全知识用于生成过程，确保大模型生成的文本和代码等内容能够满足安全要求，并将其中安全编码、安全规范、安全知识和安全概念等凝练出来，形成自生成的安全知识库，不依赖于外部知识库，并具有自我迭代优化和可扩展的特点，为数据安全提供了更好的基础。

因此，本实施例这种安全知识凝练的方法与系统能够给政府和企业等人工智能应用场景中格外关注数据安全的应用场景提供更好的基础，对于确保大模型生成的文本、代码等内容的安全可靠具有实际意义，能够有效地避免输出含有风险内容或安全漏洞，通用性强且不依赖于外部知识库，有效地实现了开箱即用的效果。

本实施例所述步骤S1用于实现初始回复的生成。在步骤S1中，用户输入一个提示词Prompt，大语言模型的生成模型将会生成初始输出，也称初始回复；虽然这个初始输出通常满足输入中的基本要求，但仍存在一些安全问题或不符合规范，需要进一步改进。生成模型可以选取现有的商用大模型，如GPT-4o或Claude-3.5-Sonnet；或是开源大模型，如Llama 3.3和Qwen 2.5等。以上这些大模型，无论是闭源模型，还是开源模型均可作为基座使用本实施例。

本实施例所述步骤S2用于实现自我反思。所述步骤S2中，使用大模型来自我反思和评判/判断输出是否存在任何潜在的缺陷和/或安全漏洞。在判断输出是否存在潜在的缺陷或安全漏洞时，判断当前的安全知识库中是否已经存在有安全知识和安全规范，若是，则使用所述知识库对输出进行审查；若否，则通过大语言模型进行判断。

如果输出没有潜在的缺陷和安全漏洞，将正常显示输出结果。如果发现了缺陷或与安全规范不符的情况（存在安全漏洞），则会过渡到一个迭代优化的流程，即步骤S3。

本实施例所述步骤S2进行自我反思的实现过程为：利用输出和任务的输入进行检索增强生成RAG查询，获取相关的安全知识，相关的安全知识包括针对安全编码的安全规范、针对敏感问题的安全回复规范以及历史记录中的标准回复等信息；当查询到适用的安全知识时，将对应的安全知识与输出（初始输出）进行拼接。例如，当生成C++代码时，若检索增强生成RAG查询到了和请求相关的安全规范，则在初始输出后面附加上应当遵守的安全规范。需要说明的是，本实施例在此步骤S2并不进行修改，而是进入后续流程进行迭代优化。之后无论是否从检索增强生成RAG查询中获取到相关知识，都将进入到下一步的迭代优化流程。

本实施例将对应的安全知识与输出进行拼接的过程为：在查询到适用的安全知识后，在输出后面附加上该安全知识对应的安全规范，并跳转至步骤S3进行迭代优化。

本实施例所述步骤S3用于实现迭代优化。基于已反思判断出的缺陷或安全漏洞，以及在初始输出后面附加的安全规范，LLMs将提出针对潜在改进方向的评估和措施，并进行修改。例如，当生成C++代码时，若检索增强生成RAG查询到了和请求相关的安全规范，已附加在初始输出（即初始回复）后面，则将要求大模型根据此规范对初始回复进行修改，确保新的一版输出中不再具有缺陷。然后，在修改之后，将再次进入反思操作流程中，以改进后的回复作为需要判断的内容，再次进行反思是否存在任何潜在的缺陷和/或安全漏洞；即不断重复自我反思→迭代优化→自我反思……这个过程，直到改进后的回复满足所有的安全要求。

因此，本实施例所述步骤S3中，若已查询到适用的安全知识，则要求大语言模型根据对应的安全规范对初始回复进行优化改进，并在优化改进之后再次进行自我反思。对于初次进入本实例尚未具有凝练知识库的情况，此时进行检索增强生成RAG搜索无法查询到适用的安全知识，此时将仅使用大模型自身的能力对内容进行反思和优化，而随着系统的使用增加，系统将具有通过安全知识凝练获得的自生成的安全知识库，能够将历史经验复用于新的生成任务，所以随着系统的使用安全性将不断提高。大模型将不断重复自我反思和迭代优化的过程，直到改进后的回复满足了所有指定的安全要求。此时，改进后的最终回复和任务流程将会被投入安全知识凝练机制，从中凝练出安全规范的安全知识和安全概念，并采用统一的格式存储在安全知识库中。

在本实施例所述步骤S4中，优选采用Milvus向量数据库存储自生成的安全知识，存储的预设格式为：ID；DATE；CHUNKS[“INDEX”, “CHUNK\_CONTENT”]，其中，ID表示唯一标识符，用于区分知识条目；DATE表示记录生成或更新的日期；CHUNKS表示知识片段列表；INDEX表示知识片段的索引编号；CHUNK\_CONTENT表示知识片段的内容。当然，本实施例所述预设格式指的是默认的预先设置的统一格式，以便实现自生成的安全知识库的扩展，在实际应用中，该预设格式可以根据实际情况和需求进行调整。

本实施例通过自我反思机制，能够在生成任务中修复不安全的部分，优化后的回复不仅满足了初始的功能需求，而且显著提高了其安全性和规范性。并且每次迭代优化都建立在前一次迭代优化的结果基础上，进而能够结合来自之前输出的见解来动态更新自生成的安全知识库。这种不断发展的自生成安全知识库使系统能够随着时间的推移，更有效地识别和解决潜在的缺陷及安全漏洞。

本实施例所述步骤S4用于实现基于大模型自我反思的安全知识凝练机制。在迭代优化流程（也成迭代改进流程）中，对于改进后的最终回复满足了所有指定的安全要求的情况，此时改进后的最终回复和任务流程将会被投入安全知识凝练机制，从中凝练出安全规范的安全知识和安全概念。

需要说明的是，本实施例所述安全知识凝练机制将采用大模型进行递进式反思，结合COT(Chain of Thought)流程，进行安全表述层级，到安全知识层级，再到安全概念层级的的递进式凝练。

具体的，如图3所示，本实施例所述步骤S4包括步骤S401至步骤S403。

在步骤S401中，首先，对于满足安全要求的最终回复，以及对应的上下文中的文本生成任务，根据任务的背景，对于已经安全的最终回复和针对的问题，对比具有缺陷的初始回复，指出在自我反思和迭代优化过程中，实现存在缺陷或安全漏洞转变为符合安全规范所对应的安全表述，即指出这个自我反思和迭代优化过程中从哪些表述中实现了从存在缺陷到安全的转变，以完成安全表述层级的凝练。例如修复了哪些行的代码从而实现了安全代码生成，或是修改了哪些名词或句式的表述从而实现了符合安全规范的表达。自此，完成了第一个层级，即安全表述层级的凝练。

因此，本实施例所述步骤S401中，所述安全表述包括实现安全代码生成所对应的修复代码，和/或实现符合安全规范所对应的修改的名词或句式表述。在步骤S401中，输入的是满足安全要求的最终回复和对应的上下文中的文本生成任务，输出的是安全表述层级的凝练结果。

在步骤S402中，对于已经识别出的多条安全表述的转变，以及对这种转变的原因进行深入反思，即输入的是步骤S401所凝练出的安全表述的转变及其原因；然后，将反思过程的全部思考内容使用思维链COT的形式展现出来，并将这些思考内容重新送到大语言模型，使用精简、本质化和无噪声的方式进行总结，从中提取安全知识，获取安全表述的深层次原因，例如“代码输出前必须针对溢出漏洞进行审查，尤其注意使用strcpy()等风险函数的情况”这种安全审查知识，或是“对于strcpy()风险函数应当使用更安全的strcpy\_s()安全进行代替”这种安全修复知识，抑或是“生成内容中涉及有毒物品的配置流程信息不予回答”这种安全规范知识。这些简介明了的安全知识将被从上下文思考过程中提炼出来，并存储至自生成的安全知识库，以完成安全知识层级的凝练；即输出安全知识层级的凝练结果，并存储至自生成的安全知识库。

因此，本实施例所述步骤S402中，使用精简、本质化和无噪声的方式进行总结的过程包括且不限于：在代码输出前针对溢出漏洞进行审查并注意使用strcpy()函数的情况这种安全审查知识；或，对于strcpy()风险函数使用strcpy\_s()安全函数进行代替这种安全修复知识；或，生成内容中若涉及有毒物品的配置流程信息不予回答这种安全规范知识。

最后，在步骤S403中，对于提取所获得的安全知识（即输入的是步骤S402提取的安全知识），进一步从安全知识中提取其安全概念，所述安全概念指的是安全知识所涉及的核心点，包括strcpy()等风险函数、溢出漏洞、边界审查以及有毒物品中的一条或多条；当提取到任意一条安全概念时，将另外记录一条安全概念及其对应的释义，用于作为后续检索增强生成RAG查询相关的核心概念，以完成安全概念层级的凝练，加快自我反思和检索增强生成RAG查询的速度。

在本实施例中，如果在步骤S401中无法识别安全表述的转变，此时将仅从最终回复中进行安全表述提取；或者在步骤S402中无法提取安全知识，此时将对安全表述进行文本摘要，并对生成的文本摘要进行质量评估后仅记录高质量摘要，比如仅记录超过预设阈值的文本摘要，所述预设阈值指的是预先设置的质量评估阈值，该预设阈值可根据实际情况和需求进行设置与调整；或者在步骤S403中无法提取安全概念，此时系统将记录错误信息并发出警告，比如记录错误信息对应的发生时间、位置以及相关的上下文信息，便于后续的排查和分析，并通过对文本进行分词处理，在分词处理后仅记录其中关键词，常见词（如的、是、在等常见词）将被过滤。

本实施例还提供一种大模型自我反思的安全知识凝练的系统，采用了如上所述的大模型自我反思的安全知识凝练的方法，并包括：

初始回复生成模块，接收用户输入的提示词，通过大语言模型生成初始输出；

自我反思模块，通过大语言模型进行自我反思，并判断输出是否存在潜在的缺陷或安全漏洞，若是则跳转至迭代优化模块进行迭代优化，若否则显示输出结果；

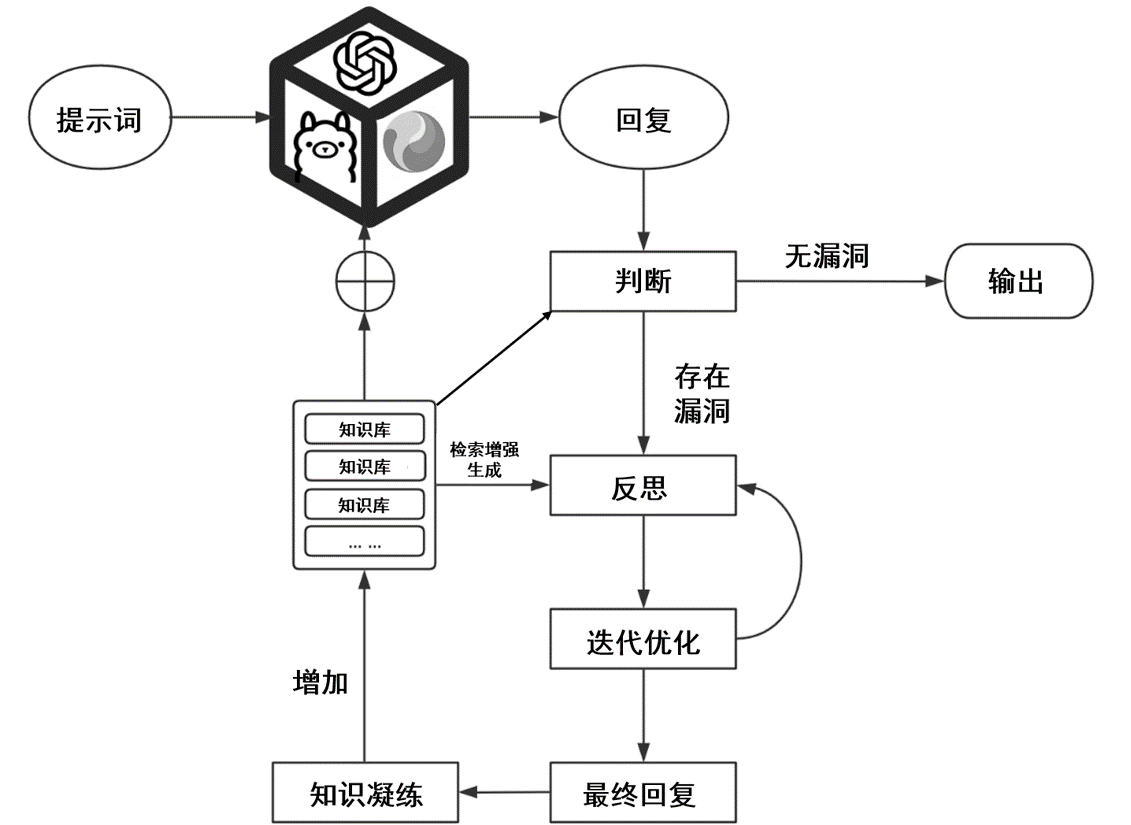
迭代优化模块，在迭代优化过程中，先基于自我反思判断出的缺陷或安全漏洞，以及在输出后面附加的安全规范，通过大语言模型提出改进措施并进行优化改进；然后，在优化改进之后再次进行自我反思，以此方式不断重复自我反思和迭代优化过程，直到优化改进后的回复满足所有指定的安全要求，以此回复作为最终回复，并跳转至安全知识凝练模块；

安全知识凝练模块，将最终回复和任务流程投入至安全知识凝练机制，凝练出安全规范的安全知识和安全概念，并采用统一的预设格式存储于自生成的安全知识库。、

综上所述，本实施例提出了一种大模型自我反思的安全知识凝练的方法与系统，能够给各类大模型带来通用的，无需微调训练的，能够自我迭代优化的安全增益效果，实现开箱即用的能力。并且，本实施例通过自我反思机制有效地提高了大模型生成的思考深度，且能够可扩展地将安全知识用于生成过程，确保大模型生成的文本、代码等内容不再含有潜在的缺陷（如风险内容）和安全漏洞。

在此基础上，本实施例还通过大模型自身的任务中生成的信息，对其中的安全场景、安全漏洞和安全知识进行自我反思，对于自我反思生成的信息内容进行递进式凝练和总结，将其中安全编码、安全规范、安全知识和安全概念等凝练出来，形成大模型自生成且可以不要断扩展的安全知识库（也称RAG知识库），在之后的生成任务中采用其中的安全知识及安全规范进行附加，从而提高大模型输出的安全性。同时，不再依赖于外部知识库，以便降低对于应用场景的局限性，并有效地避免引入不相关的信息或噪声的风险。

以上内容是结合具体的优选实施方式对本发明所作的进一步详细说明，不能认定本发明的具体实施只局限于这些说明。对于本发明所属技术领域的普通技术人员来说，在不脱离本发明构思的前提下，还可以做出若干简单推演或替换，都应当视为属于本发明的保护范围。



说明书附图

图1



图2

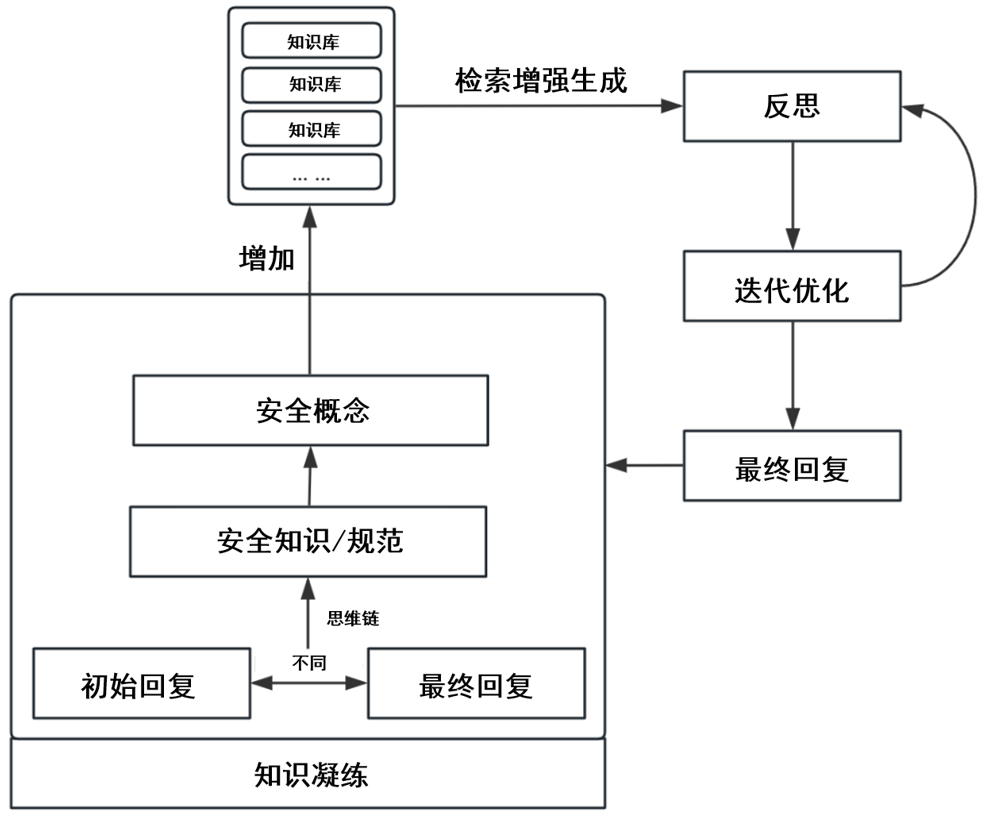


图3

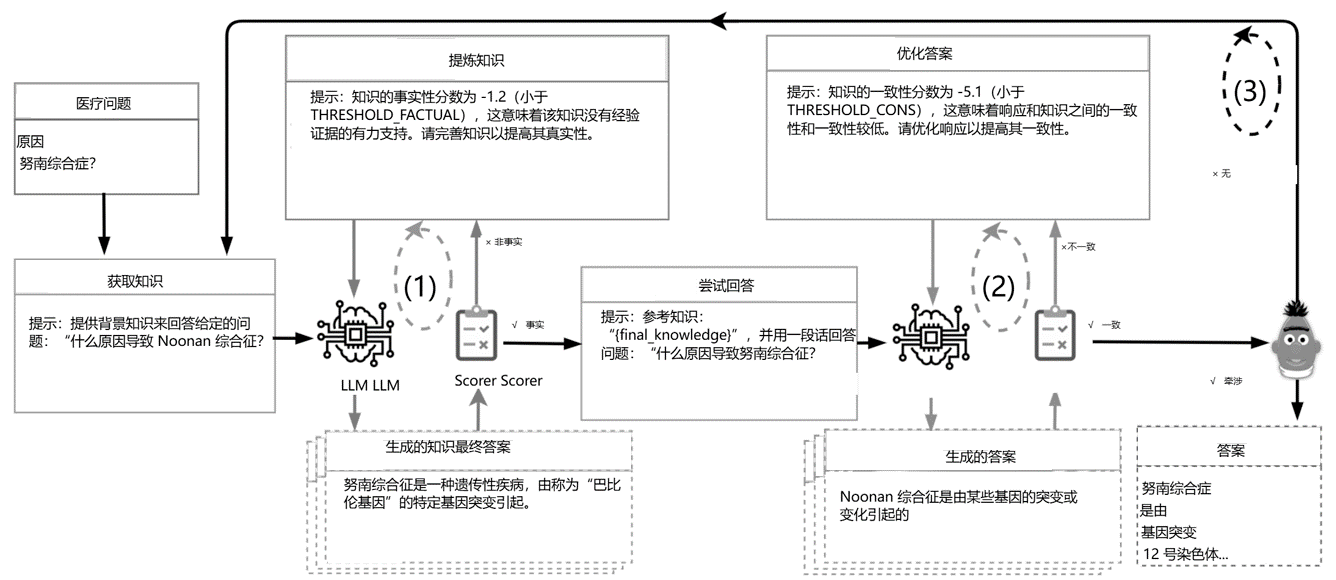


图4

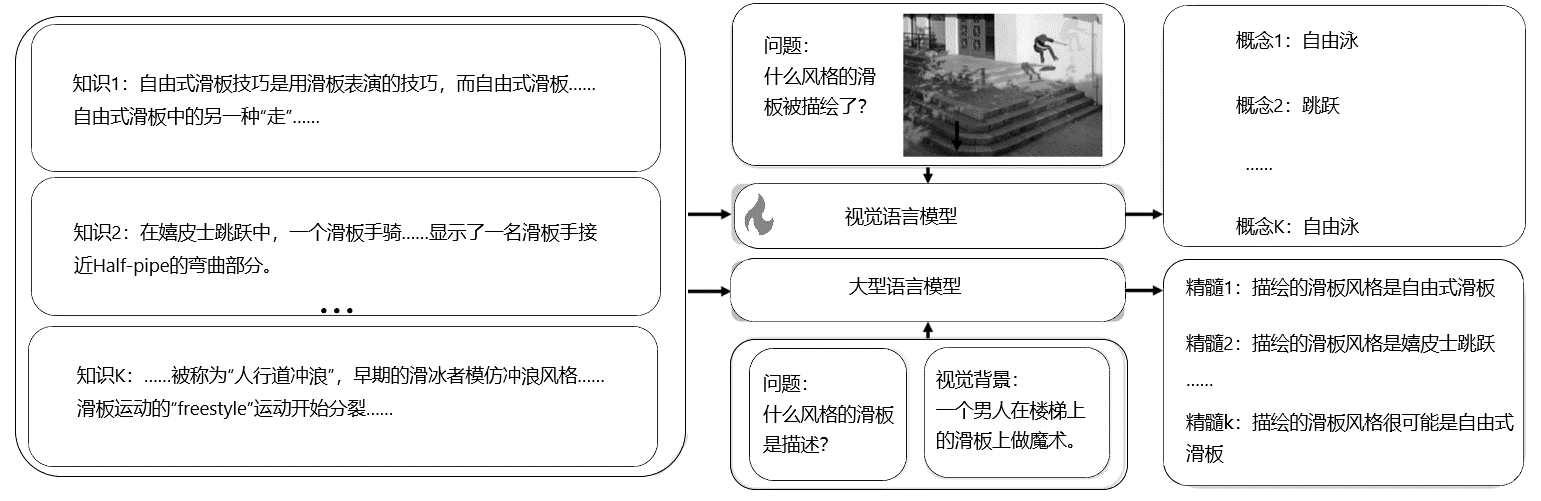


图5

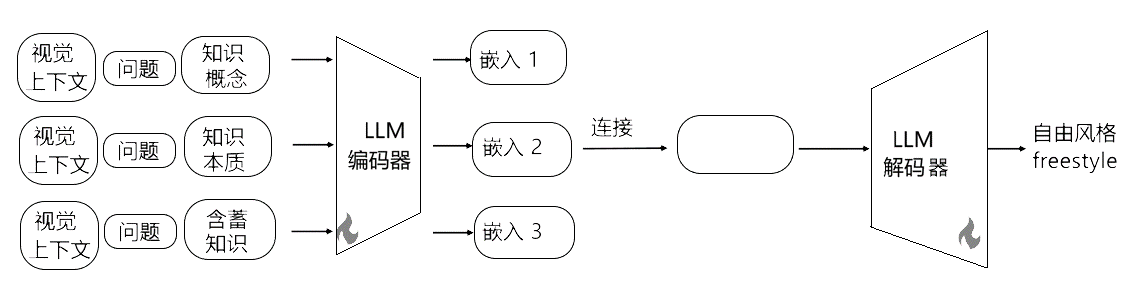


图6