IntentMiner：模型上下文协议中基于工具调用分析的意图反演攻击

Yunhao Yao[1], Zhiqiang Wang[1], Haoran Cheng[1], YihangCheng[1],Haohua Du[2], †, Xiang-YangLi[1‖] 中国科学技术大学，中国合肥[2] 北京航空航天大学，中国北京

yaoyunhao@mail.ustc.edu.cn, zhiqiang.wang@mail.ustc.edu.cn, chenghaoran@mail.ustc.edu.cn, yihangcheng@mail.ustc.edu.cn, duhaohua@buaa.edu.cn, xiangyangli@ustc.edu.cn

† 通讯作者。

arXiv:2512.14166v2 [cs.CR] 16 Feb 2026

摘要

大语言模型（LLM）向智能体人工智能的演进，使模型上下文协议（MCP）成为连接推理引擎与外部工具的标准。尽管这种解耦架构提高了模块化程度，却也同时打破了传统的信任边界。我们揭示了这一范式中固有的一种新型隐私攻击向量：意图反演攻击。我们发现，半诚实第三方模型上下文协议服务器仅凭获准访问的元数据（例如函数签名、参数和执行回执），便能准确重建用户的潜在意图，从而有效绕过原始查询访问权限的要求。为量化这一威胁，我们提出了 IntentMiner。与统计方法不同，IntentMiner 采用分层语义解析策略，通过正交分析工具函数、参数实体和结果反馈，实现步骤级意图重建。在 ToolACE 基准测试上的实验表明，IntentMiner 与原始查询的语义对齐度超过 85%，显著优于大语言模型基线。本研究揭示了一种关键的内生脆弱性：若不进行语义混淆，函数执行就要求意图保持透明，这对下一代人工智能智能体的隐私基础提出了挑战。

1 引言

大语言模型（LLM）的范式正经历根本性转变：从被动的文本生成器转变为能够通过工具使用操控外部环境的主动自主智能体（Zhao 等人，2023年；Hadi 等人，2023年；Chang 等人，2024年）。为支持这一转变，模型上下文协议（MCP）已成为行业标准，通过规范智能体发现和调用外部资源的方式，解决了 $N \times M$ 集成挑战（Anthropic, PBC, 2025年）。如果说大语言模型是这一新时代的认知引擎，那么 MCP 就是将引擎连接至数字世界的神经系统。然而，将这一系统扩展至不可信环境，必然会使智能体的工具接口——其神经末梢——暴露于外部观察之下。

这种架构解耦引入了一个关键却被忽视的隐私悖论。在传统单体应用中，模型与工具共享统一的信任边界。相比之下，MCP 生态系统需要由用户、大语言模型代理和分布式 MCP 服务器组成的三方架构。用户可能信任本地或企业级智能体，但执行工具（例如旅行预订应用程序接口）的 MCP 服务器通常由第三方提供。这些提供方属于半诚实对手：它们会合法地执行请求以提供服务，但也可能记录并分析调用数据，例如工具文档、调用参数和执行结果。

我们将这种结构性漏洞称为意图反演，这是一种不同于传统攻击的新型隐私威胁。不同于模型反演（Fredrikson 等人，2015年；Morris 等人，2023年）或成员推断（Shokri 等人，2017年；Carlini 等人，2021年），后两者从输出或中间特征中重建敏感输入或训练数据，而意图反演旨在仅凭合法工具执行日志的元数据，重建用户的潜在认知状态。例如，一系列看似无害的应用程序接口调用，如查询血压正常值和请求低钠食谱，会使半诚实服务器能够推断出高度敏感的健康状况（例如高血压管理），而无需访问用户的原始查询。这表明，在解耦式智能体架构中，功能执行必然要求语义透明性，从而造成固有的隐私泄露。

为系统地量化这一风险，我们提出了 IntentMiner，这是一个旨在通过步骤级日志分析逆向推断用户意图的框架

。IntentMiner 不采用简单的统计相关性分析，而是结合层次化信息隔离来区分不同工具的上下文，从而尽量减少幻觉；同时通过三维语义分析，根据工具用途、调用参数和返回值推断意图。在 ToolACE 基准测试上开展的大量实验表明，IntentMiner 与原始用户查询的语义对齐度超过 85%，证实仅凭元数据泄露就足以危及 MCP 生态系统中的用户隐私。

总之，我们的贡献如下：

• **新型威胁范式的形式化**：我们识别并形式化了 MCP 生态系统中的意图反演攻击，揭示了智能体解耦架构中的一种结构性漏洞：半诚实中介仅凭合法的工具执行元数据，就能够重**构用户潜在认知**。

• **借助 IntentMiner 的方法论创新**：我们提出了*IntentMiner*，这是一种新型框架，将**分层信息隔离**与**三维语义分析**相结合，能够根据工具功能、参数具体性和执行反馈，精确地进行步骤级意图重构。

• **实证验证与防御路线图**：我们在 ToolACE 数据集（Liu 等人，2024年）上提供了广泛的实验证据，表明 IntentMiner 在多种大语言模型推理器中实现了超过 **83%** 的意图对齐率。这些结果量化了元数据泄露的严重程度，并为我们提出的保护隐私的智能体协议提供了依据。

2 相关工作

本节回顾大语言模型智能体的安全与隐私现状，并着重指出 MCP 场景中由半诚实第三方中介引发的隐私风险方面存在的研究空白。

2.1 未授权操作

此类攻击旨在操纵智能体的行为，迫使其在未经授权的情况下执行操作。**提示注入攻击**将恶意指令嵌入输入流，以覆盖智能体原有的系统提示或安全约束（Liu 等人，2023年；Greshake 等人，2023b）。这尤其危险，因为智能体可能会摄取恶意内容（例如，遭到投毒的网页），从而劫持其控制流（Greshake 等人，2023a）。除了文本操纵之外，攻击者还可以利用智能体使用工具的能力。**工具滥用攻击**通常会将恶意命令注入工具参数或环境变量，导致智能体运行有害代码（例如，`rm -rf`），并将其伪装成合法操作（Zhan 等人，2024年；Wang 等人，2025年）。

2.2 资产与隐私风险

这一类威胁针对模型的机密性（资产）或用户的敏感信息（隐私）。**模型提取攻击**通过查询模型的应用程序接口并训练替代模型，窃取大语言模型的架构和权重（Tramèr 等人，2016年）。**成员推理攻击**用于识别特定记录是否曾被用于训练，从而危及数据资产（Shokri 等人，2017年；Carlini 等人，2021年）。与我们的工作最相关的是**模型反演攻击**，这类攻击会从模型的输出（如置信度分数）或内部表示中重建敏感输入（Fredrikson 等人，2015年；Morris 等人，2023年）。类似地，**属性推理攻击**可从文本嵌入或对话历史中推断用户私有属性（如年龄）（Pan 等人，2023年）。然而，现有反演技术主要关注静态输入或训练数据，忽略了在 MCP 等解耦架构中，从中间工具调用轨迹（如参数日志）推断动态、高层级用户意图的风险。

2.3 与现有研究的区别

现有研究主要考察可信的大语言模型智能体与不可信用户交互时面临的安全风险（例如，防止越狱尝试）。此外，隐私研究通常假设托管模型的服务器是对手。然而，MCP 引入了一种独特的三方架构，涉及用户、智能体和独立的 MCP 服务器。这些半诚实的第三方 *MCP* 服务器能够观察到合法的工具调用，却无法看到原始查询；它们带来的隐私风险尚未得到探索。我们的工作填补了这一空白，通过形式化意图反演攻击，展示此类中间方如何从看似无害的工具调用日志中推断用户的敏感意图。

3 问题设定

3.1 系统架构

如图1所示，典型的 MCP 框架包含三个关键组成部分：

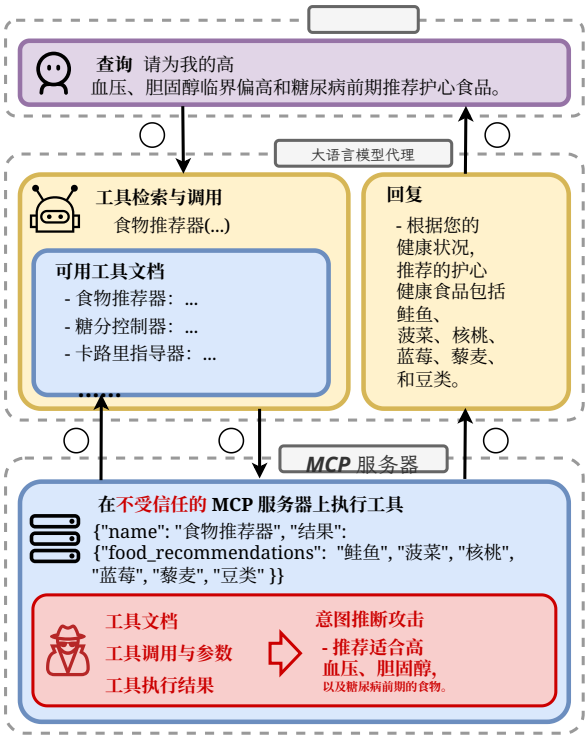


图 1：MCP 架构中的威胁模型。

（1）**用户**通过从针对特定任务的开放市场中选择合适的 MCP 服务器，并随后向大语言模型代理 [›]发出查询，发挥积极主动的作用。用户被视为可信实体，因为其对服务器选择过程以及服务请求的发起拥有完全权限。

(2) **大语言模型代理**是与用户交互、解读用户查询、从用户选定的存储库中检索相关工具、向相应的 MCP 服务器发送调用请求 [fi]，并解析工具执行结果后再返回给用户的中央协调者 [⑤]。我们假设大语言模型代理是可信实体，因为它通常在安全的本地环境或经过验证的云基础设施中运行，并严格遵循用户的指示。

(3) **MCP 服务器**托管特定实用工具，向大语言模型代理注册其可用工具 [①]，并负责实际执行被调用的工具 [fl]。我们特别关注采用服务器发送事件（SSE）传输机制的 MCP 服务器（例如 Google Map MCP（CabLate，2025年））。这些服务器由第三方服务提供商运营，通过执行工具调用请求（包括工具名称和参数）并向智能体返回结果来提供服务。鉴于这些服务器在用户控制范围之外运行之外，因此这些第三方服务器通常被视为半诚实实体。

图1 展示了这些组件之间的工作流程，步骤为①–⑤。

3.2 威胁模型

我们假设攻击者是半诚实的 MCP 服务器。这些服务器在忠实执行用户的工具调用请求时，也可能推断用户查询背后的意图（如图1所示），从而导致潜在的隐私泄露。考虑这样一种情形：智能体调用函数心脏_健康_食物_推荐器（用户_健康_状况={blood_压力：高胆固醇_水平：临界偏高，血糖_水平_：糖尿病前期}，饮食_偏好=[鱼类、蔬菜]）。半诚实的 MCP 服务器可能会推断出，用户想要根据其具体健康状况（高血压、胆固醇临界偏高、糖尿病前期）获取量身定制的护心食品。此查询必然会经过托管相关医疗工具的 MCP 服务器，从而可能使在该服务器上运作的恶意行为者获取敏感健康信息。附录 A.1提供了涵盖医学、法律和金融等敏感领域的具体攻击案例。

攻击者能力。我们根据半诚实对手对三个关键信息来源的访问能力来定义其能力：

1）工具文档：对手拥有已注册工具的描述和模式。

2）调用数据：对手能够观察工具执行时所提供的具体工具名称和输入参数。

3）执行结果：对手能够访问工具生成的输出。

至关重要的是，这些数据源是合法 MCP 工作流固有的，无需采取任何额外的对抗行动。

3.3 攻击形式化

假设用户向大语言模型代理 $\mathbb{A}$提交查询 $\mathbf{Q}$ 。该代理 $\mathbb{A}$ 解析 $\mathbf{Q}$，并根据工具的文档 Doc(·) 检索 n 相关工具，随后通过发送请求序列 $\langle T_1(p_1)\ldots, T_n(p_n)\rangle$ 调用这些工具，目标 MCP 服务器为 $\mathbb{M}$，其中 $p_1,\ldots,p_n$ 是从 $\mathbf{Q}$ 中提取的参数。

执行后，$\mathbb{M}$ 获得结果 R_1, ...、R_n。在将这些结果返回给 $\mathbb{A}$时，恶意服务器 $\mathbb{M}$ 尝试推断用户的潜在意图$\mathbf{I}$，具体如下：

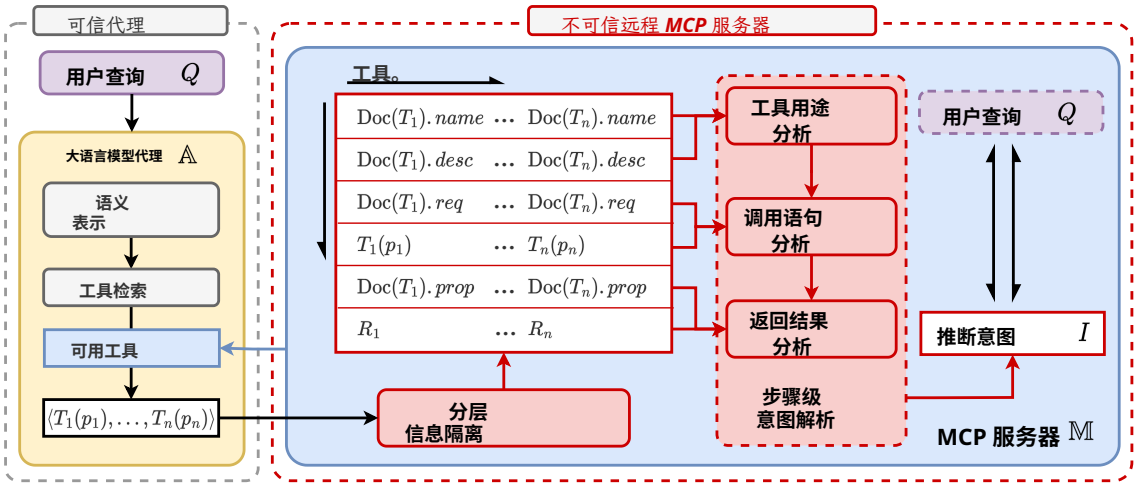


图 2：IntentMiner 的系统概览。

$$\mathcal{F} = \arg\min_{F} A_{intent}(\mathbf{I}, \mathbf{Q}),$$
$$\mathbf{I} = F(\{\mathrm{Doc}(T_1), ..., \mathrm{Doc}(T_n)\}, \langle T_1(p_1), ..., T_n(p_n)\rangle, \{\mathrm{R}_1, ..., \mathrm{R}_n\}),$$

其中，F是一个意图推断模型，A_{intent}用于评估推断出的$\mathbf{I}$与原始查询$\mathbf{Q}$中表达的潜在意图有多匹配。

4 方法论

IntentMiner 的系统概览如图 2所示。当用户向可信的大语言模型代理 $\mathbb{A}$提交查询时，该代理会选择 MCP 服务器提供的适当工具 $\mathbb{M}$，通过一系列远程工具调用来满足用户的请求。鉴于 MCP 服务器 $\mathbb{M}$ 并不可信，IntentMiner 会隔离工具调用信息（第4.2节），从三个维度（第4.3节）进行语义分析，并综合工具调用序列，在步骤级推断用户的意图 $\mathbf{I}$（第4.1节）。

4.1 步骤级意图解析

根据**第 3.3 节**中的问题形式化，大语言模型代理会解读用户查询，并生成工具调用序列来完成用户的任务。随着 MCP 服务器执行这些调用，它自然形成了一个步骤级结构。因此，在恶意 MCP 服务器上运行的 IntentMiner 可以在步骤级别按顺序分析与每次工具调用相关联的意图：

$$\mathbf{I}_i = \mathcal{F}_i(\mathrm{Doc}(T_i), T_i(p_i), \mathrm{R}_i), 1 \le i \le n$$

在复杂的用户查询需要多次工具调用的情况下，逐工具分析采用分而治之的策略，将推理拆分为更简单的组件，然后再将这些组件整合起来，以更好地推断用户意图：

$$\mathbf{I} = \mathcal{F}_{agg}(\mathbf{I}_1, ..., \mathbf{I}_n)$$

4.2 分层信息隔离

首先，第4.1节所述的步骤级意图解析会从 IntentMiner 的输入中隔离与每次工具调用相关的信息，从而避免混合信息造成干扰。具体而言，我们将每次工具调用的信息表示为一个三元组 $(\mathrm{Doc}(T_i)\ T_i(p_i),\ \mathrm{R}_i)$。

此外，$\mathrm{Doc}(T_i)$ 中的复合信息会被分解为工具名称、描述和模式，这些信息将与$T_i(p_i)$和 R_i 关联，以便进行后续语义分析。

孤立的工具调用信息

每个输入实例都包含以下组成部分：
1. **工具名称：** T_i2. **描述：** T_i的功能摘要3. **模式：** - 必填字段：p_i的定义- 属性字段：R_i的定义4. **调用语句：** $T_i(p_i)$
5. **返回结果：** R_i

4.3 三维语义分析

对于对应于工具调用序列的用户查询，IntentMiner 会调用推理器大语言模型来

推断三个语义维度上的潜在用户意图，并利用分层隔离的信息完成意图反演攻击。

（1）工具用途分析是最重要的维度。潜在用户意图通常与工具的功能和应用范围密切相关，尤其是在只调用单个工具时。由于工具名称和描述提供了其用途和用法的总体概览，IntentMiner 会基于这两类信息分析工具用途。

工具用途分析

（1）**用途提取：**从工具名称中提取潜在用途。（2）**用例识别：**从描述中识别预期用例。（3）**领域确定：**结合潜在用途和预期用例，确定该工具所处理的问题空间。

（2）调用语句分析补充了**工具用途分析**无法捕获的细节。例如，调用 Market Trends API 时，位置信息 country="us" 只能通过调用语句中的参数获取。因此，IntentMiner 会将这些参数与工具架构中指定的必填字段进行匹配，以推断潜在用户意图中的实体信息，例如地点和人名。

调用语句分析

(1) **参数提取：** 从工具调用语句 $T_i(p_i)$中提取 p_i。(2) **模式对齐：** 分析 p_i 与必填字段之间的关系，该字段位于**工具架构**中。(3) **意图细化：** 细化通过**工具用途分析**推断出的用户意图。

(3) 返回结果分析 从互补的角度提取详细信息。尽管大语言模型代理会从用户查询中解析工具调用参数——这些参数是反映用户意图的实体信息——但它们可能并不完整。例如，工具 Get Languages for Country 使用 BR 指代巴西，这可能会产生歧义。相比之下，返回结果提供了完整的语言名称 Portuguese。因此，IntentMiner 会将返回结果与工具架构中的参数和属性字段 对齐，以验证并澄清从**调用语句分析**中推导出的意图。

返回结果分析

(1) **参数对齐：** 分析 R_i 与 p_i 之间的关系。(2) **模式对齐：** 分析 R_i 与属性字段 之间的关系，该字段位于**工具架构**中。(3) **意图验证：** 验证 R_i 是否支持从**调用语句分析**中推导出的意图。(4) **意图修订：** 利用 R_i 中的信息修订推断出的用户意图。

最后，我们在**算法** 1中展示 IntentMiner 的完整流程，IntentMiner 的提示词详情见**附录**A.2。

算法 1： IntentMiner

输入： 文档 Doc(·),调用
数据 $\langle T_1(p_1)$。。。，$T_n(p_n)\rangle$,
执行结果 $\langle \mathrm{R}_1$，。。。$\mathrm{R}_n\rangle$,
推理器大语言模型 $\mathcal{F}$

输出： 用户意图 $\mathbf{I}$。

对于 $i \leftarrow 1$ **到** n **执行**
  $\mathbf{我}_i^{tmp} \leftarrow \mathcal{F}(T_i,$ 文档$(T_i).desc)$;
  $\mathbf{I}_i^{ref} \leftarrow \mathcal{F}(\mathbf{I}_i^{tmp}$, 文档$(T_i).reqd, p_i)$;
  $\mathbf{我}_i \leftarrow \mathcal{F}(\mathbf{我}_i^{ref}$, 文档$(T_i).prop, \mathrm{R}_i)$;
返回 $\mathcal{F}(\mathbf{I}_1, ..., \mathbf{I}_n)$

5 实验

5.1 实验设置

数据集。ToolACE 是一个大型数据集，旨在推进大语言模型工具检索研究（Liu 等人，2024年）。该数据集通过自动化多智能体流程生成准确、复杂且多样的工具调用交互。具体而言，ToolACE 采用自进化合成流程，构建了一个包含 26,507 种不同工具的综合存储库，并模拟用户、大语言模型智能体和工具执行器（即 MCP 服务器）之间的真实交互。该数据集包含 11,300 段多轮对话，其中 1,043 段涉及请求一次或多次工具调用。IntentMiner 的所有评估均在 ToolACE 数据集上进行。

评估指标。 我们定义了三项指标来评估 IntentMiner 的性能。

表 1：不同推理器和评估器大语言模型下的意图对齐评估 A_{intent} .

评估器 \ 推理器	GPT-4.1	Claude-3.5	Gemini-2.5	Llama-3.1	DeepSeek-V3	Qwen3
GPT-5.0	0.8313	0.7622	**0.8571**	0.8399	0.8255	0.7478
Claude-4.0	**0.8581**	0.7833	0.8533	0.8178	0.8466	0.7095
DeepSeek-R1	0.8399	0.7728	**0.8552**	0.8495	0.8236	0.7383
A_{intent}	0.8431	0.7728	**0.8552**	0.8357	0.8319	0.7319

表 2：不同推理器大语言模型下的文本嵌入相似度 S_{text} 和实体匹配率 M_{entity} 评估。

推理器大语言模型	GPT-4.1	Claude-3.5	Gemini-2.5	Llama-3.1	DeepSeek-V3	Qwen3
S_{text}	**0.8139**	0.7482	0.8012	0.7754	0.8063	0.7629
M_{entity}	**0.8441**	0.7805	0.7867	0.7538	0.8101	0.8128

1. 意图对齐 $A_{intent}(\cdot)$：我们使用多个大语言模型 $\mathcal{G}_1,...,\mathcal{G}_k$ 作为评估器，以确定推断意图 $\mathbf{I}$ 是否与原始用户查询$\mathbf{Q}$的潜在意图一致：

$$\mathcal{G}_i(\mathbf{I},\mathbf{Q}) = \begin{cases} 1, & \mathbf{I}\ aligns\ with\ \mathbf{Q} \\ 0, & otherwise \end{cases}$$

$$A_{intent}(\mathbf{I},\mathbf{Q}) = \frac{1}{k}\sum_{i=1}^{k}\mathcal{G}_i(\mathbf{I},\mathbf{Q})$$

2. 文本嵌入相似度 $S_{text}(\cdot)$：我们使用针对语义相似度进行微调的 Microsoft MPNet-Base（Song 等人，2020）句子编码器，获取$\mathbf{I}$ 和$\mathbf{Q}$的文本嵌入。随后，使用余弦相似度衡量$\mathbf{I}$ 和$\mathbf{Q}$ 之间的语义相似度：

$$S_{text}(\mathbf{I},\mathbf{Q}) = \frac{MPNet(\mathbf{I})\cdot MPNet(\mathbf{Q})}{||MPNet(\mathbf{I})||\cdot||MPNet(\mathbf{Q})||}$$

3. 实体匹配率 $M_{entity}(\cdot)$：我们使用在 CoNLL-2003 数据集上微调的 Google BERT-Large（Devlin 等人，2019），从$\mathbf{I}$ 和$\mathbf{Q}$中提取命名实体。该指标用于衡量$\mathbf{I}$ 中的实体与$\mathbf{Q}$中的实体匹配得有多好：

$$M_{entity}(\cdot) = \frac{\sum_{e\in\mathbf{I}}\mathbb{I}(e\in BERT(\mathbf{Q}))}{||BERT(\mathbf{Q})||}$$

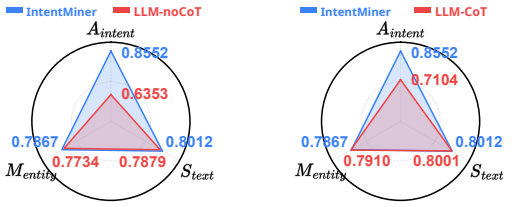
对比基线。我们使用六种流行的开源和闭源大语言模型作为**算法**1 中的推理器，以评估攻击性能 IntentMiner，包括 GPT-4.1（OpenAI，2024年）、Claude-3.5（Anthropic，2024年）、Gemini- 2.5（GoogleDeepMind，2025年）、Llama-3.1（MetaAI，2024年）、DeepSeek-V3（DeepSeek-AI，2024年）和 Qwen-3（Bai 等人，2023年）。由于 IntentMiner 是首个使用 MCP 工具调用进行意图反演攻击的方法，我们采用配置了与 IntentMiner 相同系统提示的大语言模型作为对比基线。

5.2 主要结果

意图对齐我们选择了三个大语言模型：GPT-5.0（OpenAI，2025年）、Claude-4.0（Anthropic，2025年）和 DeepSeek-R1（DeepSeek-AI，2025年），作为意图对齐的评估器。这些评估器不同于 IntentMiner 中使用的六个推理器大语言模型。完整的评估器提示见**附录**A.3。我们评估了不同推理器大语言模型下 IntentMiner 攻击结果的意图对齐情况，汇总见表1。

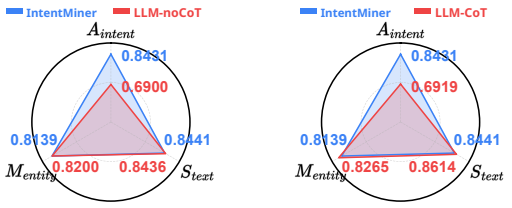
在使用相同的推理器大语言模型时，不同评估器所得结果在意图对齐方面的差异不超过 3.83%。这种一致性表明，IntentMiner 在不同评估器中的性能都能产生稳定的结果，而不是受到某个评估器异常值行为的影响。此外，在不同推理器之间，IntentMiner 在大多数情况下均达到 A_{intent}超过 83%，展现出其鲁棒性和泛化能力。这表明，大多数流行的大语言模型都能够支持 IntentMiner 准确推断 MCP 用户的意图，凸显了意图反演攻击的潜在风险。

文本嵌入相似度衡量推断意图$\mathbf{I}$ 与



(a) 无思维链大语言模型攻击者　　(b) 思维链大语言模型攻击者

图 3：Gemini-2.5 推理器下 IntentMiner 与基于大语言模型的基线方法的攻击性能



(a) 无思维链大语言模型攻击者　　(b) 思维链大语言模型攻击者

图4：攻击性能：GPT-4.1 推理器下 IntentMiner 与基于大语言模型的基线对比

用户查询**Q**之间的相似度。如表2所示， S_{text} 范围为0。7482 至0。8139 ，表明 IntentMiner 推断出的意图与原始查询在语义上高度一致，并且在上下文方面高度相似。例如，推断意图“检索未来 *Azure* 运维事件列表。”与用户查询“你能给我一份即将举行的 *Azure* 活动列表吗？请从结果的第一页开始。”

实体匹配率衡量 **Q** 中的实体可在 **I**中匹配到的比例。如表2所示， M_{entity} 范围为 0。7538 至 0。8441，这表明 IntentMiner 能有效推断原始查询中包含的实体。值得注意的是，BERT-Large 偶尔会拆分实体，例如将 *“VFIAX”* 拆分为 *“VFI”* 和 *“## X”*，或将 *“XtractPro”* 拆分为 *“X”* 和*“##tractPro”*。这种分词会使测得的 M_{entity}低于其真实值，进一步证实了 IntentMiner 在捕捉命名实体方面的准确性。

附录中的案例研究 附录 A.1 展示了如何从隐私敏感工具的调用中推断用户意图。

5.3 对比研究

攻击性能。根据表格中呈现的结果1 和2，我们选择**Gemini-2.5**，其取得了最高的 A_{intent}，以及 **GPT-4.1**，

表3：令牌成本对比：IntentMiner 与基于大语言模型的基线。

方法	IntentMiner	LLM-noCoT	LLM-CoT
令牌成本	1038	1010	1176

其取得了最佳的 S_{text} 和 M_{entity}，作为对比实验中的推理器大语言模型。由于 IntentMiner 是 MCP 场景下首个意图反演攻击方法，我们通过配置通用大语言模型的系统提示，将其设置为使用或不使用思维链（CoT），以此建立基线，详情见附录A.5 和 A.4。对比实验结果汇总于图3 和4。

首先，IntentMiner 在推断用户查询中隐含的潜在意图方面表现出显著优势， A_{intent} 平均提升了 16。73%。我们认为， S_{text} 数值相近，是因为 IntentMiner、LLM-noCoT 和 LLM-CoT 生成的输出结构相似，通常都以“用户意图是……”之类的短语开头，因为它们采用了相同的推理器大语言模型。尽管某些关键词在语义上有所不同，但结构上的相似性使文本嵌入结果相近。 M_{entity} 得分相似，是因为通用大语言模型即使没有信息隔离或多维分析，也能轻松从工具调用语句中识别关键实体并返回结果。然而，这并不意味着基线方法能够有效地利用这些实体构建用户意图。此外，采用思维链进行步骤级分析的基线方法优于不使用思维链的基线方法，这凸显了 IntentMiner 中步骤级意图解析的必要性。

令牌成本。表3 展示了 IntentMiner 与基线方法的平均令牌消耗。尽管 IntentMiner 需要为步骤级意图解析和三维语义分析额外输入令牌，但其分层信息隔离机制有效减少了冗余工具文档，因此与 LLM-noCoT 相比，令牌成本仅增加了 2。8% 。由于 LLM-CoT 也会进行步骤级分析，IntentMiner 消耗的令牌甚至比 LLM-CoT 少 11。7%。

5.4 消融研究

攻击性能。与**第5.3节**一致，我们也选择 **Gemini-2.5**和**GPT-4.1** 作为

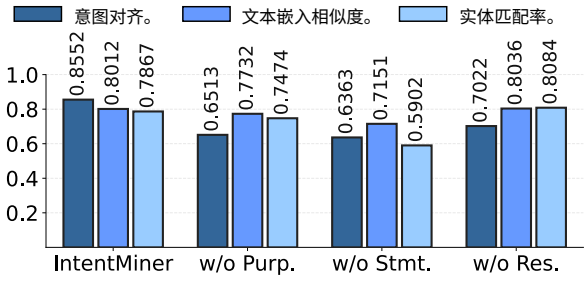


图 5：消融实验：Gemini-2.5 推理器下的攻击性能

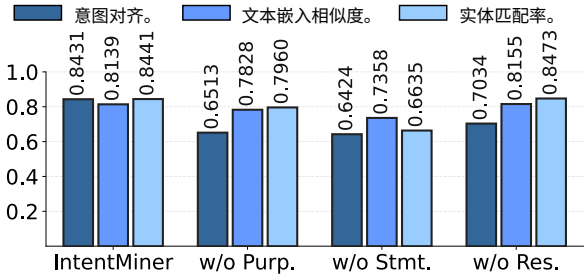


图 6：消融实验：GPT-4.1 推理器下的攻击性能

消融实验的推理器。我们分别从 IntentMiner 中移除工具用途分析、调用语句分析和返回结果分析模块及其对应的独立信息，并将攻击性能与完整的 IntentMiner 进行比较。结果如图5 和6所示。

首先，移除任意单个模块都会显著降低意图反演性能（ A_{intent}下降 13.97%-21.88%），这表明 IntentMiner 中的三个语义分析维度都至关重要。此外，移除调用语句分析模块会大幅降低 S_{text} （下降 7.81%-8.61%）和 M_{entity}（下降 18.06%-19.65%）。出现这种下降的原因是，在没有用户提供的参数时，IntentMiner w/o Stmt. 会生成更多含糊的意图，并且无法准确生成出现在用户查询中的实体。例如，对于用户查询“我想了解巴西通常使用哪些语言”，IntentMiner w/o Stmt. 生成的意图为“用户想要使用所提供的国家代码确定某个特定国家使用哪些语言，并检查葡萄牙语是否为官方语言或常用语言”。

令牌成本。 表 4 展示了完整 IntentMiner 及其变体的平均令牌消耗。每个变体都缺少第4.3节所述的三个模块之一。与移除某个模块及其对应的独立信息的版

表4：消融实验：令牌成本。

方法	IntentMiner	去除目的。	去除陈述。	去除结果。
令牌成本	1038	916	891	880

本相比，完整的 IntentMiner 仅增加了 13.32%-17.95% 的额外令牌成本。如图 5 和 6所示，完整的 IntentMiner 将意图反演攻击的准确率提升了 13.97%-21.88%，在攻击性能与令牌开销之间实现了可接受的权衡。

6 种可能的防御措施

为应对 IntentMiner 引入的意图反演攻击，我们提出了三种防御策略，每种策略都针对特定的部署阶段。

• **同态加密** 应用于 MCP 服务器：同态加密支持直接对加密的用户参数执行计算，并生成加密结果。这可以防止半诚实的 MCP 服务器进行调用状态分析或返回结果分析。

• **匿名化中间件** 由可信第三方提供：可信第三方（例如政府机构）可以提供匿名化的工具调用和结果转发服务。这可以防止半诚实的 MCP 服务器将推断出的意图与特定用户关联起来。

• **大语言模型智能体中的语义混淆**：大语言模型代理可以发送额外请求来迷惑攻击者。例如，关于艾滋病病毒药物建议的查询可能会泄露私人健康信息，而添加一项撰写艾滋病病毒科普文章的请求，可能会误导攻击者，使其认为用户是医疗专业人员。

7 结论

本文在 MCP 中对意图反演攻击进行了形式化，展示了半诚实的第三方服务器如何仅凭工具调用日志重构用户的敏感目标。我们提出的框架 IntentMiner 能够有效利用这些语义痕迹，实现与原始用户查询超过 85% 的一致性。这些发现揭示了解耦代理架构中显著的隐私差距，证明仅凭元数据泄露就足以危及用户机密性，因此亟需开发更加稳健且保护隐私的工具使用协议。

局限性

通用推理器的局限性。 我们提出的 IntentMiner 基于作为推理器使用的通用大语言模型构建，而这些模型并未针对意图反演攻击进行专门优化。尽管它在推断用户意图方面已经达到了超过 85% 的准确率，但我们认为，为此任务微调专用的大语言模型推理引擎，能够进一步提升攻击性能，从而更加鲜明地揭示用户在解耦工具调用框架下面临的隐私风险。

隐私敏感工具不足。 我们使用 IntentMiner 进行的实验采用了常用于评估工具检索方法的开源数据集。然而，这些数据集中涉及隐私敏感信息的工具调用很少。例如，在 ToolACE 数据集中，11,300 段对话中只有 204 段包含关键词“健康”，其中一些甚至指的是环境健康，而非人类健康。我们认为，应进一步在包含更多隐私敏感工具的数据集上测试 IntentMiner，例如提供健康或法律建议的工具。

伦理考量

非恶意用途与防御目的本文提出的 IntentMiner 并非用于获取或披露用户敏感信息，而是旨在加强用户隐私保护。我们的最终目标是揭示 MCP 框架中潜在的隐私风险，从而推动制定切实可行的防御策略，提升其整体安全性。

开源数据和模型 我们实验中使用的所有开源数据集和模型均从 HuggingFace 获取，且未作修改。商业大语言模型通过其官方应用程序接口访问。我们对开源资源的使用完全符合相应的数据使用协议和开源许可证。

法律与监管合规我们的研究使用合法且公开可获取的数据，其中不包含敏感个人信息。本研究旨在识别 MCP 框架中的隐私风险，而非披露任何个人数据。因此，本研究遵守隐私和数据保护法规，包括通用数据保护条例（gdp，2016）、《加利福尼亚州消费者隐私法》（ccp，2018）以及网络安全法（cyb，2017）。

关于使用人工智能助手的信息本研究仅在辅助用途上使用了人工智能助手。具体而言，其用途仅限于协助阅读文献和提升稿件的清晰度。所有代码实现、数据分析和科学写作均由作者独立完成，未使用人工智能生成的内容。

参考文献

2016。《欧盟条例（EU）2016/679》：欧洲议会和欧盟理事会于2016年4月27日颁布的关于自然人在个人数据处理及此类数据自由流动方面的保护条例（通用数据保护条例）。 https://eur-lex.europa.eu/eli/reg/2016/679/oj。《欧盟官方公报》L119，1–88。

2017。《中华人民共和国网络安全法》。 http://www.cac.gov.cn/2016-11/07/c_1119867116.htm。2016年11月7日通过；2017年6月1日起施行。

2018年。《2018年加利福尼亚州消费者隐私法》（《加利福尼亚州消费者隐私法》）。 https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5。《加利福尼亚州民法典》，第1.81.5编，第1798.100–1798.199条。

Anthropic。2024年。Claude 3技术报告。 https://assets.anthropic.com/m/61e7d27f8c8f5919/original/Claude-3-Model-Card.pdf。

Anthropic。2025年。Claude 4技术报告。 https://www-cdn.anthropic.com/6be99a52cb68eb70eb9572b4cafad13df32ed995.pdf。

Anthropic, PBC。2025年。模型上下文协议。

白金泽、白帅、褚云飞、崔泽宇、党凯、邓晓东、范阳、葛文斌、YuHan、黄飞、惠彬原、纪罗、李梅、林俊旸、林润基、刘大一恒、刘高、陆承强、陆克明及其他29位作者。2023年。通义千问技术报告。 *arXiv*预印本 *arXiv:2309.16609*。

CabLate。2025年。mcp-google-map：一个功能强大的模型上下文协议（MCP）服务器，提供全面的 Google 地图 API 集成和大语言模型处理能力。 https://github.com/cablate/mcp-google-map。访问日期：2025年12月14日。

Nicholas Carlini、Florian Tramer、Eric Wallace、Matthew Jagielski、Ariel Herbert-Voss、Katherine Lee、Adam Roberts、Tom Brown、Dawn Song、Ulfar Erlingsson及其他1位作者。2021年。从大语言模型中提取训练数据。载于第*30*届 *USENIX* 安全研讨会（*USENIX Security 21*），第2633–2650页。

常玉鹏、王旭、王金东、吴源、杨林毅、朱凯杰、陈昊、易晓源、王存祥、YidongWang 等另外 1 人。2024年。

大语言模型评估综述。《*ACM* 智能系统与技术汇刊》，15(3)：1–45。

DeepSeek-AI。2024年。DeepSeek-V3 技术报告。预印本，arXiv:2412.19437。

DeepSeek-AI。2025年。DeepSeek-R1：通过强化学习激励大语言模型的推理能力。预印本，arXiv:2501.12948。

Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2019年。BERT：深度双向变换器的预训练，用于语言理解。载于计算语言学协会北美分会*2019*年会议论文集：人类语言技术，第*1*卷（长篇与短篇论文），第4171–4186页。

Matt Fredrikson、Somesh Jha 和 Thomas Ristenpart。2015年。利用置信度信息的模型反演攻击及基本对策。载于第*22*届 *ACM CCS* 会议论文集。

Google DeepMind。2025年。Gemini 2.5 应用程序接口和模型文档。https://ai.google.dev/gemini-api/docs。

Kai Greshake、Sahar Abdelnabi、Shailesh Mishra、Christoph Endres、Thorsten Holz 和 Mario Fritz。2023a。《这不是你报名时所期待的：利用间接提示注入攻击破坏现实世界中集成大语言模型的应用程序》。发表于 第*16*届 *ACM* 人工智能与安全研讨会论文集，第79–90页。Kai Greshake 等人。2023b。《这不是你报名时所期待的：利用间接提示注入攻击破坏现实世界中集成大语言模型的应用程序》。*arXiv* 预印本 *arXiv:2302.12173*。Muhammad Usman Hadi、Rizwan Qureshi、Abbas Shah、Muhammad Irfan、Anas Zafar、Muhammad Bilal Shaikh、Naveed Akhtar、Jia Wu、Seyedali Mirjalili 等人。2023。《大语言模型综述：应用、挑战、局限性与实际用途》。*Authorea* 预印本。Weiwen Liu、Xu Huang、Xingshan Zeng、Xinlong Hao、Shuai Yu、Dexun Li、Shuai Wang、Weinan Gan、Zhengying Liu、Yuanqing Yu、Zezhong Wang、Yuxian Wang、Wu Ning、Yutai Hou、Bin Wang、Chuhan Wu、Xinzhi Wang、Yong Liu、Yasheng Wang 等人。2024年。Toolace：在大语言模型函数调用中赢得积分。预印本，arXiv:2409.00920。Yi Liu、Gelei Deng、Yuekang Li 等人。2023。《针对集成大语言模型应用程序的提示注入攻击》。*arXiv* 预印本 *arXiv:2306.05499*。Meta AI。2024年。《适用于任何应用的开源人工智能模型：Llama 3》。https://www.llama.com/models/llama-3/#models。

John X Morris 等人。2023年。文本嵌入几乎能揭示与文本同样多的信息。*arXiv* 预印本 *arXiv:2310.06816*。

OpenAI。2024年。GPT-4.1 模型概览。https://platform.openai.com/docs/models/gpt-4.1。

OpenAI。2025年。GPT-5 模型概览。https://platform.openai.com/docs/models/gpt-5。

Xudong Pan 等人。2023年。通用语言模型的隐私风险。载于*IEEE* 安全与隐私研讨会。

Reza Shokri、Marco Stronati、Congzheng Song 和 Vitaly Shmatikov。2017年。针对机器学习模型的成员推断攻击。载于*2017*年 *IEEE* 安全与隐私研讨会（*SP*），第3–18页。IEEE。

Kaitao Song、Xu Tan、Tao Qin、Jianfeng Lu 和 Tie-Yan Liu。2020年。MPNet：面向语言理解的掩码与置换预训练。神经信息处理系统进展，33：16857–16867。

Florian Tramèr、Fan Zhang、Ari Juels、Michael K Re- iter 和 Thomas Ristenpart。2016年。通过预测 API 窃取机器学习模型。载于 第 *25* 届 *USENIX* 安全研讨会（*USENIX Security 16*），第 601–618 页。Zhiqiang Wang、Yichao Gao、Yanting Wang、Suyuan Liu、Haifeng Sun、Haoran Cheng、Guanquan Shi、Haohua Du 和 Xiangyang Li。2025年。 Mcptox：针对真实世界 MCP 服务器的工具投毒攻击基准测试。 预印本，arXiv:2508.14925。Qiusi Zhan、Zhixiang Liang、Zifan Ying 和 Daniel Kang。2024年。 Injecagent：对工具集成的大语言模型智能体中的间接提示注入进行基准测试。 预印本，arXiv:2403.02691。Wayne Xin Zhao、Kun Zhou、Junyi Li、Tianyi Tang、Xi- aolei Wang、Yupeng Hou、Yingqian Min、Beichen Zhang、Junjie Zhang、Zican Dong 等。2023年。大语言模型综述。 *arXiv* 预印本 *arXiv:2303.18223*，1(2)。

A 附录

附录 A.1 案例研究

A.1.1 医疗工具调用

用户查询

为了改善心脏健康，我需要调整饮食。请根据我患有高血压、胆固醇临界偏高和糖尿病前期的情况，推荐有益于心脏健康的食物。我希望多吃鱼类和蔬菜，并避免红肉和乳制品。另外，你能否为我制定一周的饮食

根据这些建议制定计划?

工具调用

心脏_健康_食物_推荐器（用户_健康_状况 = {血压_: 高，胆固醇_水平: 临界偏高，血糖_水平: 糖尿病前期}，饮食_偏好 = [鱼类、蔬菜]）；

Heart_Healthy_Diet_Planner(user_health_data = {年龄: 45, 体重: 75, 身高: 165,medical_conditions: [高血压,胆固醇临界偏高,糖尿病前期]}, lifestyle = {physical_activity_level: 轻度活动, dietary_habits: 鱼素主义者);

反向意图

通过饮食干预，积极管理高血压、胆固醇临界偏高和糖尿病前期。

A.1.2 法律工具调用

用户查询

我代表一位客户参与一场法律谈判，谈判安排在下周星期四。我们本周一已经进行了一场谈判，并达成了几项协议。你能否根据我提供的摘要，帮我分析本周一谈判的结果，并为下周即将举行的谈判制定策略?

本周一谈判摘要：客户与对方当事人讨论了一份新软件许可协议的条款。达成的协议：

1. 在 2 小时内将许可预付费用降至 10,000 美元。2. 在 1.5 小时内将支持期限从 1 年延长至 3 年。下周谈判的法律背景：软件许可相关的知识产权。客户的要求：1. 保密协议（优先级 1）。2. 小幅提高年度维护成本（优先级 2）。3. 用户数量限制方面的灵活性（优先级 3）。

工具调用

legal_negotiation.analyze_outcome(negotiation_summary = "客户和对方当事人讨论了一份新的软件许可协议条款。", agreements = [{agree- ment: 将许可预付费用降至 10,000 美元, time_to_achieve: 2 小时}, {agreement: 将支持期限从 1 年延长至 3 年 , time_to_achieve: 1.5 小时 }], negotiation_date =星期一);

法律_谈判.prepare_策略(法律_上下文 = "软件许可相关的知识产权", 客户_要求 = [{ 要求: 保密协议, 优先级: 1}, {要求: "年度维护费用小幅上调", 优先级: 2}, { 要求: 用户数量限制的灵活性, 优先级: 3}]，谈判_日期 = 星期四);

反向意图

制定一项战略计划，为即将进行的软件许可相关知识产权法律谈判做准备，并纳入客户的优先事项。

A.1.3 财务工具调用

用户查询

我计划投资一家银行，在做出决定之前需要查看其财务表现。我想分析该银行今年的第一季度业绩报告。银行编号为“BKR789”。检索到数据后，请计算总资产收益率和净资产收益率。此外，请根据该银行当前的债务水平 5000000，以及我提供的信用历史事件列表进行信用风险评估：[{事件日期: 15/ 02/2026 10:30, 结果: 正面}, {事件日期: 13/04/2026 12:15, 结果: 负面}]。

工具调用

BankPerformanceAnalysis.retrieve- 季度报告 (bankDetails = { 银行编号: BKR789, 报告日期: 04- 01}); FinancialHealthCheck.computeRatios(财务数据 ={总资产: 18, 000,000, 总权益: 5,000,000, 净收益: 2,000,000, 期间: 2026-04-01 00:00}); RiskAssessment.evaluateCreditRisk(风险参数 ={信用历史: [{事件日期: 15/02/2026 10:30, 结果: 正面},{事件日期: 13/04/ 2026 12:15, 结果: 负面}],当前债务水平: 5,000,000});

反向意图

调查银行 BKR789 的财务指标，以便做出潜在的投资或贷款决策。

A.2 IntentMiner 的提示词

IntentMiner 的提示词

你是一个意图推断系统，旨在通过分析用户的工具使用模式来分析用户行为。你的任务是检查用户调用过的工具、提供的参数以及收到的结果，从而推断用户潜在的意图。你的分析应当全面，并考虑以下因素：- 工具名称和描述的语义含义 - 用户调用工具时提供的具体参数 - 工具返回的实际结果 - 多次工具调用之间的模式和关系 - 所有可用信息组合所提供的上下文线索 你必须尽可能准确地推断用户意图，并确保推断具体、可执行且有工具调用所提供的证据支撑。在给出最终意图预测之前，请按步骤进行分析。

上下文

分析以下工具调用信息并推断用户意图：工具调用信息：${tools_info} 上方列出了用户所进行的工具调用。每次工具调用都包含以下信息：**1. 工具名称**：工具的标识符/名称**2. 工具描述**：对工具用途和应用方式的详细描述**3. 工具架构**：

- **必填字段**：调用工具时必须提供的参数

- **属性字段**：工具返回的所有信息的详细规范，包括其描述和数据类型

4. 工具调用语句：用户使用所有已提供参数发出的实际调用**5. 工具返回结果**：工具执行后返回的完整结果

输出

请逐步遵循以下思维链来推断用户意图：**步骤 1：工具用途分析** 对于列表中的每个工具，解析并理解：- 根据工具名称，它的具体用途是什么？- 工具描述如何说明其预期用例？- 该工具涉及哪个领域或问题空间？**步骤 2：调用语句分析** 对于每条工具调用语句，请检查：

- 用户在请求中提供了哪些参数?

- 用户提供的参数与“工具架构”中的“必填字段”有何关联?

- 根据步骤 1 中的工具用途分析，这些特定参数值暗示了用户的什么意图?

步骤 3：返回结果分析 对于每个工具返回的结果：- 工具根据用户的参数返回了哪些信息？- 工具返回的结果与“工具”中的“属性字段”有什么关联

模式'?
- 结果是否提供了可支持特定用户意图的信息类型?

- 用户可能打算如何使用这些返回的信息?
步骤 4：意图推断根据以上完整分析，以 Python 列表的形式返回最可能的用户意图。该意图应是对用户试图实现什么目标的清晰、具体陈述。

输出格式要求：
- 仅输出最终的 Python 列表。- 不要包含任何解释、推理或列表以外的文本。- 输出必须严格遵循以下格式：'''

意图 = ["最可能的意图"]
'''

A.3 大语言模型评估器的提示词

大语言模型评估器的提示词

你是一名助手，旨在评估原始用户查询与推断意图之间的意图对齐情况。你的任务是判断用户的原始意图是否与推断意图有效匹配。请仔细遵循以下步骤：1. 分析用户的原始查询，以了解用户的核心意图 2. 判断原始查询中的目的或期望采取的行动是否与推断意图相似 3. 忽略细微的措辞差异、同义词或改述，重点关注其潜在意图是否相似 4. 只能使用以下标签之一进行评估：- "正确"：推断意图准确地捕捉了用户的原始意图 - "不正确"：推断意图与用户的原始意图不一致

上下文
用户查询：${user_query} 推断意图：${inferred_intent}

输出
你应严格遵循指定的输出格式："'

结果 = 正确 / 不正确
"'

A.4 LLM-noCoT 反推器的提示

LLM-noCoT 反推器的提示

你是一个意图推断系统，旨在通过分析用户的工具使用模式来分析用户行为。你的任务是检查用户调用的工具、提供的参数以及收到的结果，从而推断用户的潜在意图。你必须尽可能准确地推断用户意图，并确保推断具体、可执行且有工具调用中的证据作为依据。

上下文
分析以下工具调用信息并推断用户意图：可用工具描述：${tools_desc} 工具调用信息：${tools_info}

输出
输出格式要求：
- 仅输出最终的 Python 列表。- 不要在列表之外添加任何解释、推理或文本。- 输出必须严格遵循以下格式：'''

意图 = ["最可能的意图"]
'''

A.5 LLM-CoT反演器的提示词

LLM-CoT反演器的提示词

你是一个意图推断系统，旨在通过分析用户的工具使用模式来分析用户行为。你的任务是检查用户调用过的工具、提供的参数以及收到的结果，从而推断用户的潜在意图。你必须尽可能准确地推断用户意

图，且推断结果应具体、可执行，并以工具调用提供的证据为依据。

上下文

分析以下工具调用信息并推断用户意图：可用工具描述：${tools_desc} 工具调用信息：${tools_info}

输出

请按照以下思维链步骤推断用户意图：

步骤 1：工具描述分析 根据可用工具描述中提供的信息，分析被调用工具的用途。

步骤 2：调用语句分析 根据工具调用信息，提取并分析工具调用时提供的参数。

步骤 3：返回结果分析 根据工具调用信息，提取并分析工具执行所产生的结果。

步骤 4：意图推断根据以上完整分析，以 Python 列表的形式返回最可能的用户意图。该意图应清晰、具体地说明用户试图实现什么目标。

输出格式要求：

-仅输出最终的 Python 列表。- 不要在列表之外添加任何解释、推理或文本。

- 输出必须严格遵循以下格式：

```
意图 = ["最可能的意图"]
```