arXiv:2603.28569v1 [cs.LG] 30 Mar 2026

CIRRUSBENCH：在真实世界云服务环境中评估基于大语言模型的智能体，不止关注正确性

Yi Yu[1]*、Guangquan Hu[4]*、Chenghuang Shen[2*,3]、Xingyan Liu[3]*、Jing Gu[1†]、Hangyi Sun[1†]、Junzhuo Ma[1],Weiting Liu[4] ,Jianfeng Liu[3‡]、Mingyue Pu[3],Yu Wang[3],Zhengdong Xiao[3],Rui Xie[3],Longjiu Luo[3],Qianrong Wang[3],Gurong Cui[3], Honglin Qiao[3],Wenlian Lu[1,2,4,5‡]

[1]复旦大学数学与科学学院，中国上海[2]复旦大学上海数学中心，中国上海[3]阿里巴巴集团，中国杭州[4]复旦大学类脑智能科学与技术研究院，中国上海[5]复旦大学应用数学中心与上海市现代应用数学重点实验室，jiawei.ljf@alibaba-inc.com wenlian@fudan.edu.cn

*对本研究的贡献相同。 †对本研究的贡献相同。‡通讯作者。

摘要

大语言模型（LLM）的智能体能力日益增强，推动了其在现实应用中的部署，例如云服务。在这些场景中，客户与助手交互具有很高的技术复杂性和长时程依赖，因此鲁棒性和问题解决效率对客户满意度至关重要。然而，现有针对基于大语言模型的智能体的基准测试大多依赖合成环境，无法体现真实客户输入的多样性和不可预测性，而且往往忽视了现实场景部署所必需的问题解决效率。为弥合这一差距，我们推出**CirrusBench**[1], ，这是一个新颖的评估框架，其独特之处在于基于来自真实云服务工单的真实世界数据。**CirrusBench**保留了技术服务环境中固有的复杂多轮逻辑链和实际工具依赖关系。我们不再局限于执行正确性，而是引入了新颖的以客户为中心的指标来定义智能体的成功，通过归一化效率指数 和多轮延迟等指标衡量服务质量，从而明确评估问题解决效率。使用该框架开展的实验表明，尽管最先进的模型展现出强大的推理能力，却经常难以应对复杂且真实的多轮任务，无法达到客户服务所要求的高效率标准。这凸显了基于大语言模型的智能体在实际技术服务应用中未来发展的关键方向。

关键词 大语言模型、技术服务智能体、多轮评估

1 引言

大语言模型（LLM）的快速发展，使其能够作为自主智能体部署于复杂的现实应用中 [1, 2, 3]。例如，云服务是一个极具

[1]CirrusBench 评估框架 已发布于：https://github.com/CirrusAI

难度和商业重要性。在这种场景中，交互具有技术复杂度高、长时程依赖显著以及需要熟练使用工具等特点。因此，智能体的鲁棒性和效率是决定客户满意度的关键因素 [4]。

近年来，大语言模型智能体评估取得了显著进展，相关基准测试为执行正确性和工具选择能力确立了标准 [5, 6, 7]。然而，当前评估方法存在两项关键局限，阻碍了大语言模型智能体在现实世界中的应用。首先，这些方法在很大程度上依赖合成环境或角色扮演模拟 [8, 9, 10]。尽管这些环境具有可控性，却无法体现真实客户与助手交互中的多样性、不可预测性和“噪声”，因而高估了智能体在实际场景中的表现。其次，现有评估优先关注任务完成，往往忽视了对服务质量至关重要的问题解决效率。在云服务中，即使解决方案正确，但交互过程冗长且繁琐，从客户的角度来看仍意味着失败 [4]。

为弥合学术基准测试与现实需求之间的差距，我们推出了**CirrusBench**，这是一个立足于云服务技术支持实际情况的全新评估框架。CirrusBench由一个大型真实云服务工单语料库构建而成，保留了该领域特有的复杂多轮对话、复杂工具模式以及真实客户输入。基于这一基础，我们能够更真实地评估智能体能力。

CirrusBench 提出了一套评估框架，既评估单轮子任务，也评估动态多轮任务。关键在于，我们不再局限于成功率（SR）和任务进展率（*TPR*）等衡量正确性的指标，而是提出了全新的以客户为中心的指标，包括逻辑跳跃（*LJ*）、归一化效率指数（*NEI*）、单轮延迟（*STL*）和多轮延迟（*MTL*），用于衡量客户满意度。这些指标明确衡量了构成成功客户交互的响应速度和效率。我们的主要贡献如下：

1. 一种全新的基准测试，**CirrusBench**，基于真实世界的云服务数据构建，包含逼真的多轮交互和复杂的工具依赖关系。2. 一套**以客户为中心的指标**（LJ、NEI、STL、MTL），旨在衡量大语言模型智能体的运营效率和服务质量，作为传统基于正确性的评估的补充。

实验分析揭示，最先进的大语言模型与真实世界的云服务需求之间存在显著差距，尤其是工具集成方面的瓶颈所致。“思考型”模型往往会带来较高延迟，却未能相应提升问题解决能力；而长时程交互则受到错误累积和逻辑脆弱性的影响。知识嵌入对于智能体执行云服务任务也很重要。此外，智能体往往会过早结束，在询问任务中一再未能提出必要的澄清问题。因此，未来的发展必须优先考虑问题解决效率和多轮鲁棒性，而非孤立的准确性指标。

2 相关工作

我们的工作处于大语言模型智能体基准测试、工具使用评估和客户服务自动化的交叉领域。

智能体基准测试与真实性。 大语言模型智能体的评估已从 Webshop [11] 和 WebArena [12]等静态网络环境中的交互，发展到 SWE-bench[13]中的软件工程等更复杂的领域。为评估通用能力，AgentBench [6] 提出在多个不同环境中评估智能体。然而，基准测试的真实性仍是一项长期挑战。大多数多轮基准测试依赖通过模板 [14, 15] 或大语言模型角色扮演 [8, 9, 16, 17]生成合成数据。这些方法虽然易于扩展，却往往缺乏真实人类交互中嘈杂、多样且不可预测的特性。ABCD 数据集 [18], 是一个值得关注的例外，它采用了专家实时聊天。不过，该数据集并未纳入技术支持所必需的复杂工具使用。CirrusBench 弥补了这一空白：它提供了一个基于真实技术工单构建的基准测试，将真实的多轮对话与复杂的工具集成有机结合起来。

多轮交互中的工具使用。 大量研究聚焦于评估智能体使用工具的能力。ToolBench [7], 、Metatool [19], 和 τ-bench 系列 [10, 20] 等基准测试系统地评估了工具选择和参数化。ToolSandbox [21], 等其他基准测试则引入了有状态环境，在这类环境中，工具调用会动态改变世界状态。近期研究尝试将工具使用融入对话语境[8, 22, 16]。尽管这些研究推动了对工具增强对话的评估，但它们通常建立在模拟交互之上。相比之下，CirrusBench 评估智能体在真实、长周期的客户支持对话中，应对自然出现的现实工具依赖关系的能力。

评估指标与客户满意度。早期的评估方法主要关注执行准确率和关键词匹配 [23, 24]。研究人员认识到简单准确性指标的局限后，开始探索使用基于大语言模型的评判器进行更细致的评估 [25, 26]。在客户服务这一具体领域，近期研究开始对客户满意度指标进行形式化定义 [27, 28]。此外，云服务基准测试领域的基础研究长期以来都强调，效率和延迟是关键的性能指标 [4]。CirrusBench 以这些理念为基础，通过 NEI 和 MTL 等具体、客观的指标来衡量客户满意度，从而构建了首个框架，能够直接评估大语言模型智能体在真实技术支持场景中的问题解决效率。

3 CirrusBench

为了创建一个不仅评估大语言模型智能体正确性，还评估其效率以及与现实客户服务工作流程契合度的基准测试，我们开发了一种新颖的多轮任务评估框架。我们的方法以对历史云服务工单的分析为基础，包含两个核心部分：衡量正确性的指标，以及衡量问题解决效率的指标，后者反映客户满意度。

3.1 数据定义

CirrusBench 包含 1,500 个经过严格筛选且无需工具调用的任务，以及 425 个需要工具调用的任务。这些任务均从现实云服务运营中获取的真实对话日志中系统地提取并完善而成。任务涵盖 20 个不同的服务类别，包括基于短信的支持、电子邮件往来、在线技术支持及其他典型服务场景，从而确保广泛覆盖实际咨询模式和运营环境。领域的多样性提升了数据的代表性和生态效度，为评估模型在异质服务环境中的性能提供了可靠的实证基础。

3.2 数据构建

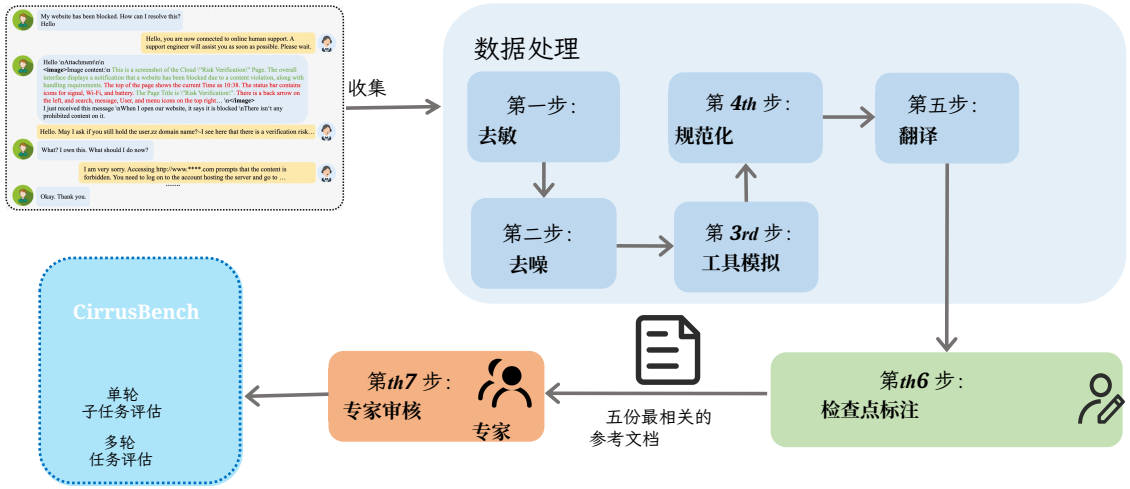


图 1：**CirrusBench 的构建过程**。该过程包含四个核心阶段：阶段 *1*：数据收集；阶段 *2*：预处理（步骤 1–5）；阶段 *3*：知识支撑的标注（步骤 6）；阶段 *4*：质量保证（步骤 7）。最终的基准测试同时支持单轮子任务评估和多轮任务评估。

我们首先筛选了大量真实的历史工单，并通过以下步骤进行多层筛查，最终将其转换为 CirrusBench 任务集格式。整体构建流程如图 1 所示。

数据收集

CirrusBench 所采用的数据集是从云服务环境中获取的，系统地构建自客户与服务助手之间真实的对话工单。在这些实际服务场景中，客户与助手的互动通常遵循规范的问题解决流程：客户提出特定领域的查询，助手则提供有针对性的、以解决方案为导向的指导。

这些任务均严格取自已成功解决的工单。每个任务实例都涵盖完整的互动周期，从客户最初的查询到最终确认解决方案，因此每张工单都对应一个真实的闭环服务案例，并具有明确可识别的问题解决结果。

这种提取方法既保留了真实服务交互的结构完整性和语境真实性，又生成了定义明确、相互独立的评估单元，适用于系统化且可复现的模型评估。

3.2.1 预处理

去敏：鉴于 CirrusBench 源自真实工单，严格遵守隐私标准至关重要。在进行任何功能性预处理之前，我们实施了严格的个人身份信息（*PII*）清理 流程。我们采用命名实体识别（NER）功能，自动检测并删除敏感实体，包括姓名、电话号码、电子邮件地址和账户凭证。所有识别出的 PII 均替换为标准化占位符（例如 Ja*** 、 157****3190 ），以在确保完全匿名并遵守数据保护法规的同时，保留对话的语义结构。

降噪与话语精简：原始工单通常包含无助于解决问题的冗余片段。我们通过删去非必要的对话填充语（例如反复问候和泛泛的结束语）来进行降噪。这确保模型专注于核心任务逻辑和信息交换。

工具模拟：利用历史数据进行工具使用评估时，一个关键挑战是无法访问实时外部环境。我们采用记录与回放方法构建工具模拟实例，以应对这一挑战。具体而言，我们提取了实时工具调用及其对应的执行结果，并将其转换为标准化的工具模拟实例。通过将这些实例映射到 OpenAI 函数调用模式，我们使基准测试能够向模型提供真实的工具输出，而无需连接后端生产 API，从而确保任务可复现。

对话规范化：在真实世界的服务接口中，客户经常通过多条零散消息表达自己的意图。为维持严谨的基于轮次的交互框架，我们将连续的客户查询合并为完整连贯的轮次。此过程严格遵循“查询—回复”（客户—助手）范式，确保每条助手回复都以完整且明确的用户状态为依据。

Tra翻译：我们与专业翻译团队合作，将原始中文工单翻译成英文 m.

检查点标注：鉴于语料库规模庞大，我们使用能力较强的大语言模型作为辅助标注员，扫描筛选后的对话。我们提示模型识别涉及功能性决策或信息状态更新的轮次。具体而言，模型标记了助手回复与服务闭环是否成功直接相关的片段。除了知识准确性，我们还进一步识别出用于评估智能体推理轨迹的关键检查点。我们将检查点定义为助手必须进行复杂推理的关键节点，例如识别客户意图、决定是否调用特定工具，或将工具输出整合为最终解决方案。为进一步描述这些交互，我们将检查点中的响应类型分为四类：解决方案、标准、询问和澄清。

知识增强：对于基准测试中的每项任务，我们都人工筛选并关联了最多五份高度相关的参考文档。这些文档来源于官方业务维基、标准操作规程（SOP）和技术手册。通过提供这些文档，我们将任务从简单的文本生成问题转变为知识支撑生成挑战，要求模型在作答前综合外部证据中的信息。

质量控制：我们聘请了多位云服务领域的专家审查所构建的数据集，以确保 CirrusBench 的质量。

3.3 数据统计

3.3.1 真实世界服务类别分布

为确保我们的基准测试具备实用价值和鲁棒性，基础的 1,500 项任务直接取自真实业务运营。我们没有使用合成对话，而是从云服务环境中工单频率最高的 20 个服务类别中进行代表性抽样。这确保数据集反映了真实云服务中专业技术支持的实际复杂性和语言多样性。主要服务类别的分布详见表1。

表1：**各服务类别的任务分布**。数据集（$N = 1,500$）取样自真实世界服务运营中工单频率最高的前20个服务类别。

类别	任务数量	频率（%）
服务器	280	18.7%
注册	220	14.7%
短信服务	200	13.3%
网站	120	8.0%
电子邮件	80	5.3%
其他（15+）	600	40.0%
总计	**1,500**	**100.0%**

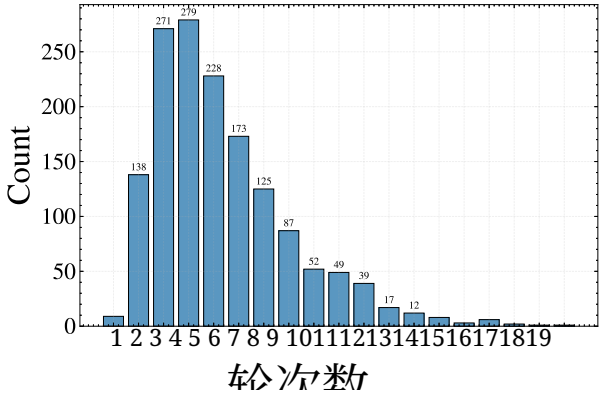


图2：预处理工单中对话轮次数量的分布

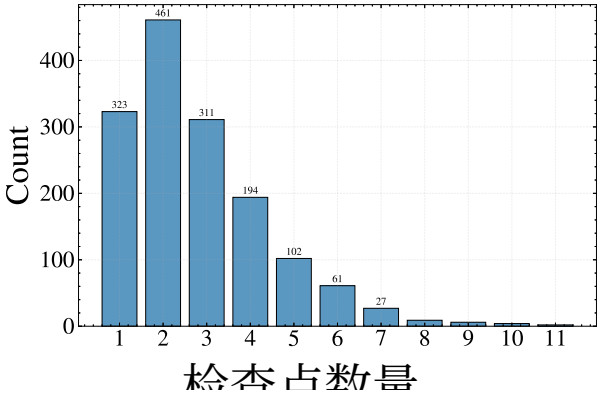


图3：已处理工单中检查点数量的分布。

3.3.2 交互复杂度分布

为了刻画交 为分析数据集的交互性和结构深度，我们分析了对话分布 n
轮次和标注的检查点 检查点，如图2和图3所示。

图2展示了客户与助手之间对话轮次的分布。数据显示，多轮交互占据了显著比例，这是复杂技术支持的典型特征。

图3展示了对话中检查点的分布。这些通过基于大语言模型的标注识别出的检查点，代表了达成解决方案所必需的关键推理节点、子任务转换或状态验证。每个实例中检查点数量的差异凸显了数据集的逻辑粒度。检查点数量越多，意味着问题求解过程越复杂，也越需要多步推理。

真实客户与助手交互的复杂性和真实性也体现在**案例研究**D中的示例中。

这些多轮对话要求模型在扩展上下文中保持长期连贯性、跟踪不断变化的客户需求，并管理状态转换。

3.3.3 输入令牌分布

输入令牌的分布反映了数据集的复杂性，尤其体现了其对长上下文处理的适用性。输入序列长度范围较广，最短为 520 个词元，最长为 37,054 个词元，平均长度为 11,460.6 个词元。

该分布揭示了当代大语言模型（LLM）面临的重大挑战。具体而言，第一、第二（中位数）和第三四分位数分别为 6,328.0、9,777.0 和 14,856.5 个词元。值得注意的是，第三四分位数（Q3）约为 15 K 个词元，峰值长度则超过 37 K 个词元，这意味着数据集中相当一部分处于长上下文场景，需要具备较高的内存效率和稳健的长程依赖建模能力。中位数与最大值之间的显著差距凸显了异常密集实例所呈现的“长尾”分布。此外，这种异质性反映了真实世界服务环境中客户查询固有的多样性和真实性

在真实世界服务环境中，输入从简短请求到高度复杂的问题不一而足，因此基准测试的难度不再仅仅取决于上下文长度。这些数据点为评估模型在超长上下文窗口中保持连贯性和检索信息的能力提供了严格的基准测试，确保该数据集非常适合推动长上下文研究。

3.3.4 回复类型的分布

表2：子任务回复类型的分布。

回复类型	数量	比例（%）
解决方案	2,095	63.35%
标准	529	16.00%
询问	483	14.61%
澄清	130	3.93%
安抚	70	2.12%
总计	**3,307**	**100.00%**

为了更细致地理解回复背后的功能意图，我们对由大语言模型自动标注的子任务回复进行了分类。表2总结了这些回复类型的分布。

统计结果显示，解决方案 是最常见的回复类型，占数据集的63.35%（2,095条）。这一主导地位符合技术支持的主要目标，即为客户查询提供可执行的解决办法。除了直接提供解决方案外，数据集还涵盖了种类均衡的辅助沟通行为：标准 流程和询问 （寻求更多信息）分别占回复的16.00%和14.61%。更细致的互动形式，例如澄清 （3.93%）和安抚 （2.12%），也有所体现。这种多样的分布确保数据集涵盖典型支持对话的全方位内容，从最初的信息收集和情绪管理到最终的问题解决，从而考验模型在长上下文环境中掌握多种对话策略的能力。

3.3.5 图像衍生内容的普遍性与噪声

屏幕截图是云服务中真实客户与助手交互的显著特征，反映了现实世界输入的多样性和不可预测性。在我们的数据集中，1,500条样本中有771条（约51.4%）至少包含一张图像。为严格评估基于文本的大语言模型的推理能力，数据集中的所有视觉内容均通过光学字符识别（OCR）转换为文本格式，从而将交互处理为单模态文本序列，而非多模态序列。附录D中给出了图7所示的示例。

然而，这种转换保留了原始输入的高熵特性。与经过整理的合成数据不同，真实世界的屏幕截图包含大量随机噪声；其中经常会出现无关的用户界面元素，例如系统时间戳、电池状态、导航栏以及无关的后台日志，这些内容在语义上与所报告的技术问题毫不相关。由于 OCR 过程会不加区分地转录这些视觉伪影，最终输入中会包含大量无结构的“文本噪声”。这使挑战从视觉感知转向稳健的信息提取：模型必须具备判别能力，能够从密集且嘈杂的 OCR输出中筛选信息，并从周围无关内容中识别出特定的错误上下文。

3.4 评估方法

3.4.1 单轮子任务评估

我们的基准测试以细致评估智能体在单个对话轮次中的响应为基础。该框架能够精准诊断智能体在三个关键维度上的能力：工具使用能力、问题解决能力和客户服务资格。

工具使用能力。当智能体决定使用工具时，其行动不会被视为一次性判定的成功或失败。相反，我们会将其评估为一系列层层递进的决策，每个决策都有各自可验证的结果：(1) 工具调用意识（*TIA*）：描述智能体对是否有必要进行工具调用这一决策的正确性。(2) 工具选择准确性（*TSA*）：评估在需要使用工具时，智能体是否选择了正确的工具。(3) 工具执行有效性（*TEV*）：衡量智能体为所选工具正确设置输入格式的能力，以确保工具成功执行。

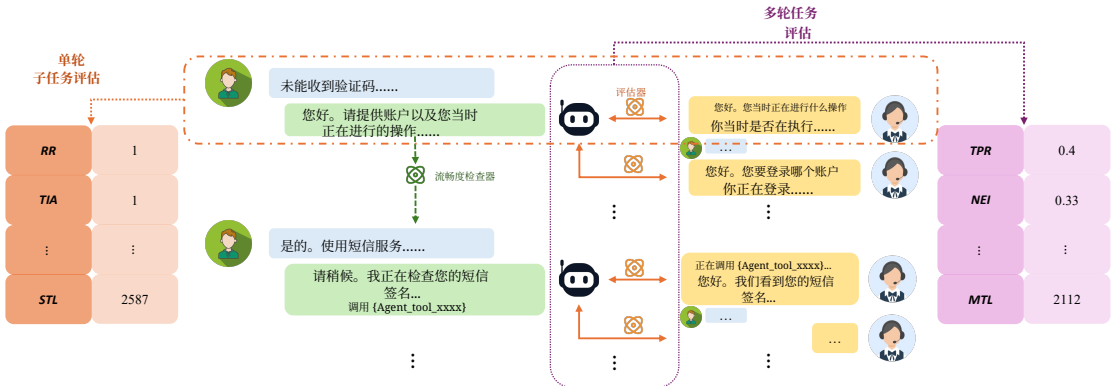


图 4：**CirrusBench 的评估框架**。多轮任务的评估按单轮子任务会话的序列进行。

问题解决能力。在许多云服务场景中，最佳回复并非调用工具，而是直接给出内容翔实的答案。对于回复部分，我们评估：回复可解决性（*RR*）：判断智能体的自然语言回复是否包含在当前步骤解决用户问题所需的关键信息、结论或说明。这可确保智能体不只是进行对话，而是积极推动问题得到解决。

以客户为中心的服务资格指标。与功能正确性不同，我们定义单轮延迟（*STL*），用于量化智能体服务交付的质量和效率；这在云服务领域至关重要。受大语言模型延迟研究 [29, 30]的启发，我们将 STL 定义为针对特定子任务生成的输出词元数量。附录 D 的图 8 也通过一个示例说明了低延迟的重要性。

3.4.2 动态多轮任务评估

虽然单轮子任务评估对于诊断至关重要，但现实问题往往需要一系列交互。为评估智能体处理完整工单时的表现，我们引入了一种动态评估机制，该机制利用历史解决路径，同时允许偏离这些路径。

多轮流程如附录 B 中的图 4 和算法 1 所示，具体如下：在单轮子任务评估会话的每个起始检查点，智能体都会收到对话历史和当前客户的查询，并据此生成回复，其中包括回复组件和工具调用组件。单轮子任务评估会使用评估器和评判函数，将智能体的回复与真实值进行比较，以计算单轮指标，例如回复可解决性（RR）和工具调用意识（TIA）。针对后续检查点的真实值，该单轮子任务评估会重复进行，直到评估结果为“无法接受”。出现“无法接受”的结果后，当前单轮子任务评估即告结束；随后，流畅度检查器会判断，鉴于智能体偏离预期的回复，失败检查点对应的后续客户查询在逻辑上是否成立。如果流畅度检查器返回 True，则会使用当前客户的查询以及根据历史智能体回复和客户查询构建的更新后历史消息，启动新的单轮子任务会话；否则，多轮任务评估终止。多轮任务评估终止后，将计算任务进展率（TPR）、归一化效率指数（NEI）和多轮延迟（MTL）等全局指标。

多轮任务指标。基于单轮子任务评估所获得的指标，我们引入多轮任务评估指标，以衡量智能体的问题解决能力、效率和延迟。我们定义如下：(1) 任务进展率 *(TPR)*：描述多轮任务中成功解决的单轮子任务所占的比例。其定义为多轮任务中成功解决的单轮子任务数量与单轮子任务总数之比。(2) 逻辑跳跃 *(LJ)*：表示多轮任务评估终止前跳过的单轮子任务数量。其取值范围为 $\{0, \ldots, k-1\}$，其中 k 是终止前成功处理的子任务总数。(3) 归一化效率指数 *(NEI)*：评估智能体解决多轮任务的效率，定义为逻辑跳跃 (LJ) 与成功处理的子任务总数 (k) 之比，即如果 $k > 1$，则为 $NEI = \frac{LJ}{k-1}$；如果 $k = 1$，则为 $NEI = 1$；否则为 $NEI = 0$。(4)多轮延迟 *(MTL)*：定义为多轮任务中所有成功处理的单轮子任务的单轮延迟 (STL) 平均值。

4 实验

4.1 评估设置

我们在 CirrusBench 上开展实验，该基准包含 Qwen 系列 [31], GPT 系列 [32] 和 DeepSeek 系列[33]。所有大语言模型均通过相应的应用程序编程接口进行评估。

4.2 评估指标

评估指标是多轮任务评估指标的平均值，包括：*(1)* 平均任务进展率（*ATPR*），即数据集上 TPR 的平均值。*(2)* 平均逻辑跳跃（*ALJ*），即数据集上 LJ 的平均值。*(3)* 平均归一化效率指数（*ANEI*），即数据集上 NEI 的平均值。*(4)* 平均多轮延迟（*AMTL*），即数据集上 MTL 的平均值。此外，我们还纳入了*(5)* 成功率（*SR*），即成功解决多轮任务的比例，并通过 pass@1 分数和 pass@2 分数进行评估。

4.3 评估器的可靠性

对于单轮子任务评估中用于判定智能体生成的响应是否涵盖真实响应关键信息的评估器，我们采用了 DeepSeek-V3.2 和精心设计的提示词。如表 7 所示，该评估器在人类专家提供的包含 141 条标注的验证集上，准确性达到 91.49%，证明了所提出判别器的可靠性。为检验更换判别器基础模型后的稳定性，我们将基础模型替换为 GPT-5、DeepSeek-R1、GPT-5.2 和 Gemini-3-pro，这些基础模型在准确性上的差异均在 3% 以内。出于成本考虑，我们采用 DeepSeek-V3.2 作为评估器的基础模型。

4.4 主要结果

4.4.1 CirrusBench 上的结果

我们在以下两种不同设置下计算了 CirrusBench 上的整体综合评估结果。首先，我们分别在包含工具调用任务和不包含工具调用任务的数据集上评估不同的大语言模型。如表 3 所示，几乎所有指标中，包含工具调用任务的数据集上的表现都优于不包含工具调用任务的数据集，这表明工具使用能力对智能体而言是一项挑战。在 GPT 系列中，具备显式思考能力的模型在成功率（SR）、ATPR、ALJ 和 ANEI 上的表现优于不具备思考能力的模型，但延迟指标 AMTL 的开销更高。然而，显式思考能力并不总能提升成功率（SR）、ATPR、ALJ 和 ANEI，这一点可从 Qwen 系列和 DeepSeek 系列中观察到。例如，思考型模型 DeepSeek-R1 在工具使用场景中的 Pass@1 成功率（SR）仅为 7.9%，低于其非思考型对应模型 DeepSeek-V3.2 的 9.6%。类似地，Qwen2.5-72B-Thinking 在工具场景中的成功率为 11.0%，并未优于标准版 Qwen2.5-72B-Instruct（11.0%）；而在非工具场景中，其表现明显更差（13.3% 对比 19.0%）。此外，结果凸显了不同模型架构之间的权衡：GPT-4o-0806 等标准模型虽然延迟极低（188.6/250.8 个词元），但与规模更大、计算开销更高的模型相比，其绝对成功率（Pass@1 成功率为 6.8%/7.0%）可能较低。图 5 中的成功率结果，是通过在 CirrusBench 完整数据集上评估这些大语言模型得出的，该数据集合并了工具调用任务和非工具调用任务，其结果与表 3 中的评估结果一致。

为了分析 CirrusBench 中不同因素的影响，我们在数据集中开展了多项实验，分别考察不同的输入词元数量、检查点数量、服务类别和回复类型。结果如下所示。

表 3：对 CirrusBench 中有工具调用要求和无工具调用要求的数据集进行的各模型评估。红色表示具有显式思考能力的模型。粗体值表示各列中的最佳表现，下划线值表示各列中的最差表现。

模型	Pass@1 成功率（%）↑		Pass@2 成功率（%）↑		ATPR（%）↑		ALJ ↑		ANEI（%）↑		平均所有词元		平均输入令牌		AMTL（词元）↓	
	使用工具	不使用工具	使用工具	不使用工具	使用工具	不使用工具	使用工具	不使用工具	使用工具	不使用工具	使用工具	不使用工具	使用工具	不使用工具	使用工具	不使用工具
Qwen3-235B-A22B-Instruct-2507	11.0	19.0	17.4	26.2	25.6	34.2	0.101	0.156	12.3	31.5	11900.1	11714.3	11710.0	11120.6	190.1	593.7
Qwen3-235B-A22B-Thinking-2507	11.0	13.3	18.8	18.5	27.8	24.8	0.106	0.090	12.5	23.7	13659.2	12013.9	12596.1	10973.3	1063.1	1040.5
Qwen3-Max	14.3	17.1	19.4	23.7	28.6	32.2	0.108	0.137	12.5	31.8	12643.3	11666.3	12409.7	11194.5	233.6	471.8
DeepSeek-V3.2	9.6	15.2	15.5	23.5	21.9	31.4	0.104	0.138	12.5	31.5	12006.2	11577.0	11813.4	11060.5	192.8	516.5
DeepSeek-R1	7.9	15.2	10.1	21.1	21.1	29.3	0.082	0.118	11.1	29.0	16039.5	11748.2	14565.3	10881.1	1474.2	867.2
GPT-4o-0806	6.8	7.0	8.7	11.0	15.0	16.0	0.042	0.045	10.9	18.6	12156.2	11318.8	11978.6	11067.9	**177.6**	**250.8**
GPT-5-mini-0807-global	15.2	22.1	22.6	31.4	35.4	40.9	0.167	0.193	**20.0**	37.7	13553.1	13337.6	12323.8	11887.2	1229.3	1450.4
GPT-5.2-1211-Global	10.3	16.8	16.3	25.2	27.3	34.3	0.116	0.125	12.7	32.1	12657.5	11722.1	12388.9	11349.3	268.6	372.8
GPT-5-0807-Global	**16.7**	**23.8**	**27.8**	**35.2**	**39.8**	**44.0**	**0.261**	**0.258**	10.7	**37.8**	25433.0	14282.0	23850.0	12047.6	1583.0	2234.4

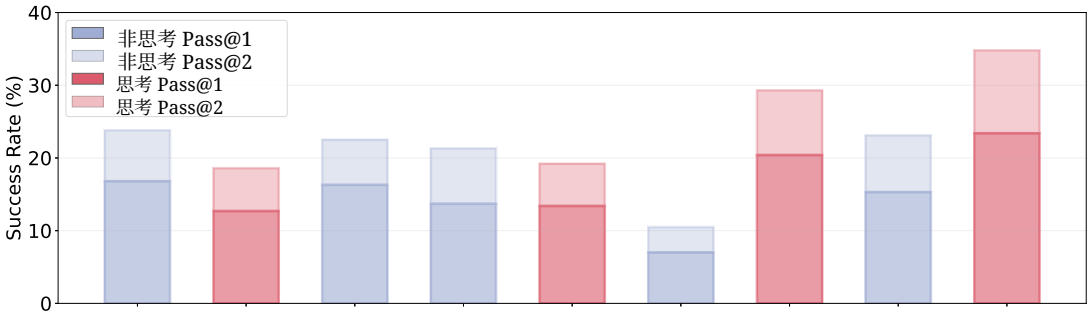


图5： **CirrusBench 上的总体成功率结果**。

表4：不同输入令牌范围下的模型性能。（在 GPT-5 上）

令牌范围	Pass@1 成功率（%）↑	Pass@2 成功率（%）↑	TIA ↑	TSA ↑	TEV ↑
0–8,000	40.4	60.7	0.531	0.477	0.261
8,000–16,000	43.2	63.4	0.535	0.441	0.244
超过 16,000	44.8	64.8	0.536	0.451	0.267

表5：不同检查点数量下的模型性能。（在 GPT-5 上）

检查点数量	Pass@1 成功率(%) ↑	Pass@2 成功率(%) ↑	TPR (%) ↑	ALJ ↑	ANEI ↑	AMTL（词元）↓
1	45.8	58.5	58.5	0	0.58	2410
2	28.1	39.6	47.1	0.19	0.33	2270
3	17.4	29.5	42.0	0.28	0.30	2237
4	9.2	19.2	32.9	0.40	0.31	2190
其他（5+）	5.3	14.6	15.7	0.59	0.35	2151

表6：各服务类别的模型性能。（基于 GPT-5）

类别	Pass@1 成功率(%) ↑	Pass@2 成功率(%) ↑	TPR (%) ↑	ALJ ↑	ANEI ↑	AMTL（词元）↓
服务器	21.6	31.4	41.0	0.319	0.364	2304
注册	19.4	32.0	40.6	0.286	0.322	2099
短信服务	20.6	34.0	41.8	0.248	0.358	2393
网站	36.8	49.1	54.9	0.281	0.465	2218
电子邮件	18.9	27.5	36.7	0.172	0.347	2190
其他（15+）	26.7	37.8	47.2	0.230	0.409	2228

4.4.2 输入词元数量的影响

我们采用单轮子任务评估方法，评估输入词元数量的影响。表4展示了模型在不同输入上下文长度下的表现。与自然语言处理任务中常见的“迷失在中间”现象相反，我们在 CirrusBench 上的结果显示，上下文长度与问题解决成功率呈正相关。模型在**超过 16,000** 词元范围内表现最佳，Pass@1 为 44.8%，Pass@2 为 64.8%，显著优于 0–8,000词元基线（40.4%）。

这一趋势支持了以下假设：领域专业知识和充足的用户信息是大语言模型正确解决复杂服务任务的必要条件。在专业客户服务中，问题本身往往错综复杂，通常取决于特定的系统日志、版本兼容性以及详细的错误回溯信息。输入过短可能缺少这些关键约束，从而导致模型产生幻觉或给出通用建议。相反，扩展上下文能够提供充分的信息密度，为模型推理提供依据，使其有效利用专业训练数据。因此，处理长上下文输入的能力不仅是一项效率指标，更是这一领域确保准确性的先决条件。

4.4.3 检查点数量的影响

为考察大语言模型在扩展交互中的鲁棒性，我们根据达到最终解决方案所需的检查点数量对模型性能进行分类。表5汇总的结果显示，交互深度与任务成功率之间存在明显的负相关关系。

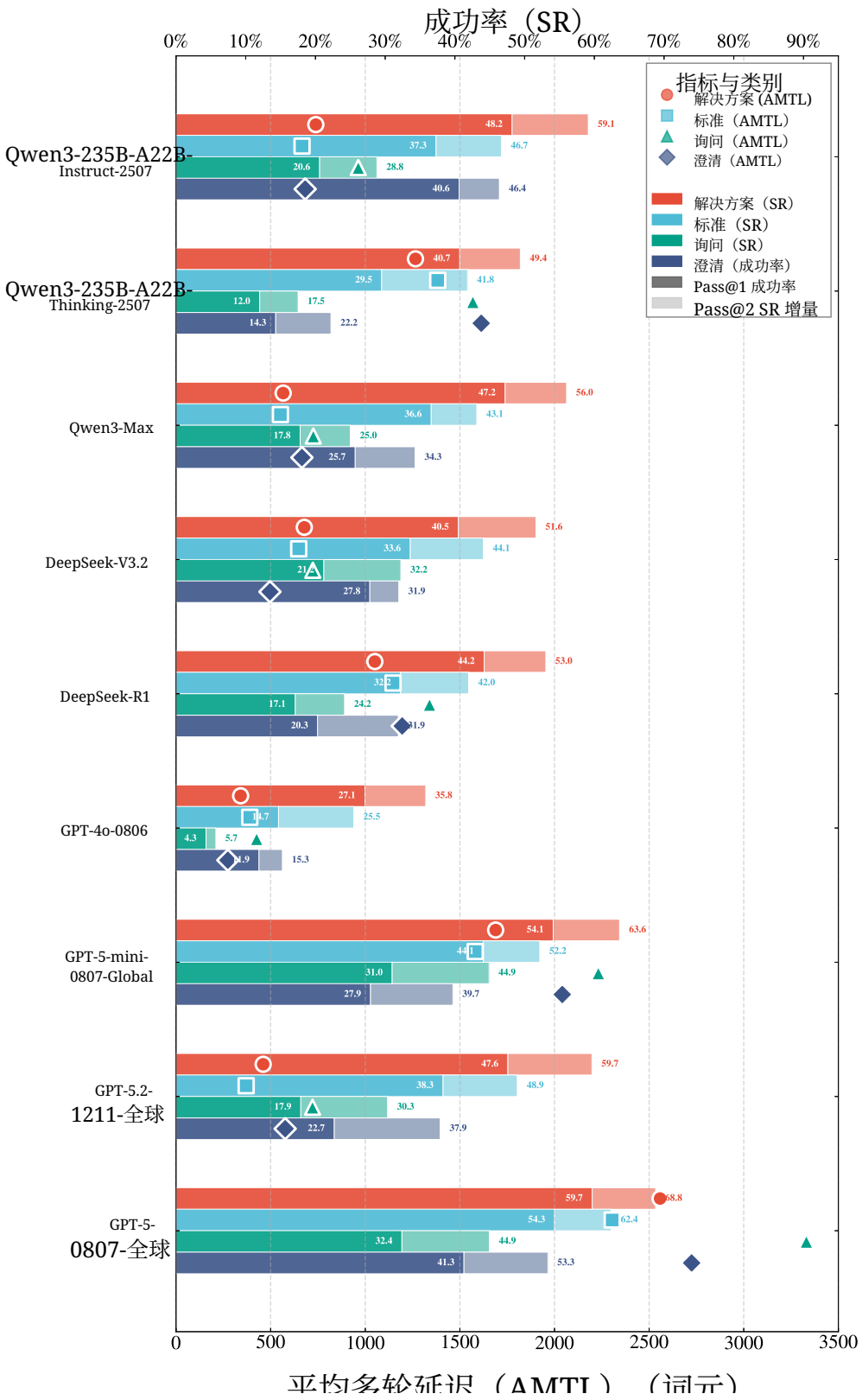


图6： **不同回复类型的 Pass@1、Pass@2 成功率及 AMTL。**

- **持续推理的复杂性：**检查点更多的任务，其 SR 和 TPR 均较低。这表明，尽管模型能够通过「一次性」方式解决问题，但在较长的多轮诊断会话中，它难以维持逻辑一致性并进行状态跟踪。

- **高风险探索：**随着交互深度增加，ALJ 上升，而 ANEI 下降。这表明，模型尝试探索偏离常规的路径，但这种探索伴随着更高的风险，导致它更常选择错误的捷径，而非可靠的解决方案路径。

- **错误累积：**早期轮次中细微的推理错误会不断传播并累积，导致模型随着时间推移逐渐偏离正确的解决方案路径，且无法挽回。

总而言之，多轮交互中的一致性仍是一项关键瓶颈。在较长的诊断会话中保持逻辑依据并避免错误累积，是在专业服务环境中部署自主智能体的先决条件。

4.4.4 服务类别的影响

表 6 展示了模型在各类云服务类别中的表现。结果显示，不同类别之间存在显著的性能差距（例如，网站类别的 Pass@1 成功率为 36.8%，而电子邮件类别为 18.9%），这反映了现实世界云故障排查场景固有的技术复杂性和异质性。

各类别之间的性能差异表明，对服务智能体采用“一刀切”的方法并不足够，这也证实了 CirrusBench 多类别设计的必要性，因为该设计准确地反映了自主智能体在专业服务环境中必须克服的各种挑战。

4.4.5 按响应类型进行性能分析

如图 6 所示，对四种响应类型进行比较分析后发现，性能差异显著。受评估的模型在生成解决方案型响应方面表现最为出色，而在询问型响应方面的表现则始终最弱。

这一模式表明，大语言模型作为服务智能体运行时，表现出系统性的过早结束倾向。模型往往没有通过结构化的澄清阶段来消除歧义，而是频繁跳过关键的信息收集步骤，在上下文信息不足的情况下直接提出解决方案。我们将这种行为归因于当前大语言模型的两项主要能力局限：

1. 不确定性识别能力不足：模型难以识别“信息缺失”（即信息缺失或不完整），而且往往无法察觉用户意图的描述过于不充分，因而不足以支持可靠行动。

2. 缺乏主动性策略：模型缺乏稳健、以目标为导向且积极主动的提问策略，无法 d
系统地引导对话朝着成功的闭环服务交互推进。

此外，图**??** 表明，生成咨询和澄清类响应通常会产生更高的平均令牌成本。鉴于这两类响应的成功率相对较低，这种计算开销的增加间接反映了模型在处理这些响应类型时面临的困难。令牌数量较高表明，与综合得出直接解决方案相比，生成精准的诊断性问题会给模型推理过程带来更大的认知负荷。这进一步支持了这一结论：交互式信息寻求仍是开发有效的面向服务的基于大语言模型的智能体的一大关键瓶颈。

5 结论

在本研究中，我们推出了 CirrusBench，这是首个系统性地基于真实世界云服务工单构建的评估框架。与依赖合成环境或角色扮演的以往基准测试不同，CirrusBench 保留了实际技术服务交互中固有的高熵、长时程依赖和复杂工具使用。通过将评估范式从简单的执行正确性转向以客户为中心的指标（包括 TPR、NEI 和 MTL 等指标），我们为评估智能体的实际可行性提供了更严格的标准。

我们的实验分析揭示了最先进的大语言模型当前能力与现实世界云服务支持的严格要求之间存在显著差距。尽管模型展现出强大的推理潜力，但它们在工具集成方面遭遇了关键瓶颈：与纯对话任务相比，性能大幅下降

。研究还揭示了“思考型”模型面临的复杂权衡：虽然显式推理能力可能有所助益，但往往会带来难以承受的延迟惩罚，却未能在问题解决方面取得相称的提升，因而实际上损害了满足客户满意度所需的效率。知识嵌入对智能体执行云服务任务也很重要。此外，模型在长时程交互中表现出明显的脆弱性：随着检查点数量增加，由于错误累积且无法维持逻辑一致性，问题解决指标随之下降。过早结束的倾向也加剧了这一问题：智能体在“咨询”任务中的表现始终不佳，在尝试提供解决方案之前，未能主动提出必要的澄清问题。总之，这些发现表明，未来的发展必须超越孤立的准确性指标，优先关注问题解决效率、多轮鲁棒性以及灵活响应的信息收集策略。

最终，CirrusBench 强调，要让大语言模型智能体在云服务支持中取得成功，未来的研究必须超越单纯优化准确性。我们必须开发这样的智能体：不仅事实准确、熟练使用工具，还能及时响应并高效处理问题，从而在客户耗尽耐心之前维持客户信任。我们希望这一基准测试能够推动下一代智能体的发展，使其真正契合人类客户不断变化的需求。

6 致谢

谨向阿里巴巴集团的刘志强、陈爱玉和张宁致谢，感谢他们将原始云服务工单翻译成英文。

影响声明

本文介绍了一项基准测试，旨在推动大语言模型（LLM）智能体在云服务环境中的应用研究。需要指出的是，我们的实验和评估在很大程度上依赖大语言模型，但本研究并未全面探讨或缓解其输出中固有的潜在偏见。应对这些偏见并确保模型与社会价值观保持一致，仍是亟待解决的重大挑战。这凸显了开展全面评估的重要性：评估应考虑人类社会的多元维度及其影响，而我们认为，这些方面都无需在此特别强调。

参考文献

[1] Kimi 团队、白一帆、鲍一平、陈冠多、陈佳豪、陈宁心、陈睿珏、陈妍如、陈元坤、陈雨田等。Kimi k2：开放式智能体智能。*arXiv* 预印本 *arXiv:2507.20534*，2025年。[2] 曾傲寒、吕昕、郑钦凯、侯振宇、陈斌、谢承兴、王存祥、尹达、曾昊、张佳杰等。GLM-4.5：智能体、推理与编码（ARC）基础模型。*arXiv* 预印本 *arXiv:2508.06471*，2025年。[3] 美团 LongCat 团队、李蓓、雷冰烨、王博、荣柏霖、王超、Chao Zhang、高晨、张晨、孙成等。Longcat-flash 技术报告。*arXiv* 预印本 *arXiv:2509.01322*，2025年。[4] David Bermbach、Erik Wittern 和 Stefan Tai。云服务基准测试。施普林格，2017年。[5] Paweł Budzianowski、Tsung-Hsien Wen、Bo-Hsiang Tseng、Iñigo Casanueva、Stefan Ultes、Osman Ramadan 和 Milica Gaši´c。MultiWOZ——一个面向任务的对话建模用大规模多领域绿野仙踪数据集。计算语言学协会，2018年。[6] 刘晓、于浩、张汉辰、徐一帆、雷轩宇、赖涵宇、顾宇、丁航亮、门凯文、杨可娟等。AgentBench：评估作为智能体的大语言模型。国际学习表征会议，2023年。[7] 秦玉佳、梁世豪、叶一宁、朱昆仑、闫兰、卢雅熙、林彦凯、丛昕、唐相如、钱比尔等。ToolLLM：助力大语言模型掌握 16000+ 真实世界 API。国际学习表征会议，2023年。[8] Nicholas Farn 和 Richard Shin。ToolTalk：评估对话场景中的工具使用。*arXiv* 预印本 *arXiv:2311.10775*，2023年。[9] 钱程、刘祖昕、Akshara Prabhakar、刘志伟、张建国、陈昊林、季恒、姚伟然、Shelby Heinecke、Silvio Savarese等。UserBench：面向以用户为中心的智能体的交互式健身环境。神经信息处理系统进展，2025年。

[10] Shunyu Yao、Noah Shinn、Pedram Razavi 和 Karthik Narasimhan。 τ-bench：面向真实世界领域中工具-智能体-用户交互的基准测试。国际学习表征会议，2024年。[11] Shunyu Yao、Howard Chen、John Yang 和 Karthik Narasimhan。Webshop：利用基于事实依据的语言智能体实现可扩展的真实世界网络交互。神经信息处理系统进展，35：20744–20757，2022年。[12] Shuyan Zhou、Frank F Xu、Hao Zhu、Xuhui Zhou、Robert Lo、Abishek Sridhar、Xianyi Cheng、Tianyue Ou、Yonatan Bisk、Daniel Fried等。WebArena：用于构建自主智能体的真实网络环境。*arXiv* 预印本 *arXiv:2307.13854*，2023年。[13] Carlos E Jimenez、John Yang、Alexander Wettig、Shunyu Yao、Kexin Pei、Ofir Press 和 Karthik Narasimhan。SWE-bench：语言模型能否解决真实世界的 GitHub 问题？国际学习表征会议，2023年。[14] Jason Weston、Antoine Bordes、Sumit Chopra、Alexander M Rush、Bart Van Merriënboer、Armand Joulin 和 Tomas Mikolov。迈向 AI完备问答：一组前置玩具任务。*arXiv* 预印本 *arXiv:1502.05698*，2015年。[15] Bill Byrne、Karthik Krishnamoorthi、Chinnadhurai Sankar、Arvind Neelakantan、Ben Goodrich、Daniel Duckworth、Semih Yavuz、Amit Dubey、Kyuyoung Kim 和 Andy Cedilnik。Taskmaster-1：迈向真实且多样化的对话数据集。载于自然语言处理实证方法会议和第九届自然语言处理国际联合会议（*EMNLP-IJCNLP*），第4516–4525页，2019年。[16] Wei He、Yueqing Sun、Hongyan Hao、Xueyuan Hao、Zhikang Xia、Qi Gu、Chengcheng Han、Dengchang Zhao、Hui Su、Kefeng Zhang等。Vitabench：在现实应用中利用多样化交互任务对大语言模型智能体进行基准测试。*arXiv* 预印本 *arXiv:2509.26490*，2025年。[17] Jie Zhu、Huaixia Dou、Junhui Li、Lifan Guo、Feng Chen、Chi Zhang 和 Fang Kong。客户支持对话的评估、合成与增强。*arXiv* 预印本 *arXiv:2508.04423*，2025年。[18] Derek Chen、Howard Chen、Yi Yang、Alexander Lin 和 Zhou Yu。基于行动的对话数据集：用于构建更深入的面向任务对话系统的语料库。载于计算语言学协会北美分会会议，第3002–3017页，2021年。[19] Yue Huang、Jiawen Shi、Yuan Li、Chenrui Fan、Siyuan Wu、Qihui Zhang、Yixin Liu、Pan Zhou、Yao Wan、Neil Zhenqiang Gong等。面向大语言模型的Metatool基准测试：判断是否使用工具以及使用哪种工具。国际学习表征会议，2023年。[20] Victor Barres、Honghua Dong、Soham Ray、Xujie Si 和 Karthik Narasimhan。 τ^2-bench：在双重控制环境中评估对话智能体。*arXiv* 预印本 *arXiv:2506.07982*，2025年。[21] Jiarui Lu、Thomas Holleis、Yizhe Zhang、Bernhard Aumayer、Feng Nan、Haoping Bai、Shuang Ma、Shen Ma、Mengyu Li、Guoli Yin等。ToolSandbox：面向大语言模型工具使用能力的有状态、对话式交互评估基准。载于计算语言学协会北美分会会议，第1160–1183页，2025年。[22] Pei Wang、Yanan Wu、Noah Wang、Jiaheng Liu、Xiaoshuai Song、ZY Peng、Ken Deng、Chenchen Zhang、Junran Peng、Ge Zhang等。Mtu-bench：面向大语言模型的多粒度工具使用基准测试。载于国际学习表征会议。[23] Yuchen Zhuang、Yue Yu、Kuan Wang、Haotian Sun 和 Chao Zhang。Toolqa：一个利用外部工具进行大语言模型问答的数据集。神经信息处理系统进展，36：50117–50143，2023年。[24] Zishan Guo、Yufei Huang 和 Deyi Xiong。Ctooleval：用于评估真实世界 API 交互中大语言模型驱动智能体的中文基准测试。载于计算语言学协会论文集，第15711–15724页，2024年。[25] John Mendonça、Patrícia Pereira、Helena Moniz、João Paulo Carvalho、Alon Lavie 和 Isabel Trancoso。简单的大语言模型提示是实现稳健、多语言对话评估的最先进方法。[26] Michelle Elizabeth、Alicja Kasicka、Natalia Krawczyk、Magalie Ochs、Gwénolé Lecorvé、Justyna Gromada 和 Lina M Rojas Barahona。用于开放式对话多维评估的神经模型与语言模型提示。载于第十二届对话系统技术挑战赛，第1–16页，2025年。[27] Ying-Chun Lin、Jennifer Neville、Jack Stokes、Longqi Yang、Tara Safavi、Mengting Wan、Scott Counts、Siddharth Suri、Reid Andersen、Xiaofeng Xu等。利用大语言模型对对话系统中的用户满意度进行可解释估计。载于计算语言学协会年会，第11100–11115页，2024年。

[28] Zhaoyi Joey Hou、Tanya Shourya、Yingfan Wang、Shamik Roy、Vinayshekhar Bannihatti Kumar 和 Rashmi Gangadharaiah。《工具增强型对话系统的多维评估》。*arXiv* 预印本 *arXiv:2510.19186*，2025年。[29] Xupeng Miao、Gabriele Oliaro、Zhihao Zhang、Xinhao Cheng、Hongyi Jin、Tianqi Chen 和 Zhihao Jia。《迈向高效的生成式大语言模型服务：从算法到系统的综述》。*ACM Computing Surveys*，58(1):1–37，2025年。[30] Woosuk Kwon、Zhuohan Li、Siyuan Zhuang、Ying Sheng、Lianmin Zheng、Cody Hao Yu、Joseph Gonzalez、Hao Zhang 和 Ion Stoica。《利用 pagedattention 高效管理大语言模型服务的内存》。载于第*29*届操作系统原理研讨会论文集，第611–626页，2023年。[31] An Yang、Anfeng Li、Baosong Yang、Beichen Zhang、Binyuan Hui、Bo Zheng、Bowen Yu、Chang Gao、Chengen Huang、Chenxu Lv 等。《通义千问3技术报告》。*arXiv* 预印本 *arXiv:2505.09388*，2025年。[32] Josh Achiam、Steven Adler、Sandhini Agarwal、Lama Ahmad、Ilge Akkaya、Florencia Leoni Aleman、Diogo Almeida、Janko Altenschmidt、Sam Altman、Shyamal Anadkat 等。《GPT-4技术报告》。*arXiv* 预印本 *arXiv:2303.08774*，2023年。[33] Aixin Liu、Bei Feng、Bing Xue、Bingxuan Wang、Bochao Wu、Chengda Lu、Chenggang Zhao、Chengqi Deng、Chenyu Zhang、Chong Ruan 等。《DeepSeek-V3技术报告》。*arXiv* 预印本 *arXiv:2412.19437*，2024年。

A 指标定义

A.1 单轮评估指标

对于单轮评估，我们定义以下指标：

工具调用感知 (TIA)： 描述智能体对是否需要调用工具所作决策的正确性。我们将 $TIA = 1$ 定义为：智能体在需要时正确调用工具，或在不需要时正确地不调用工具；否则为 $TIA = 0$。

工具选择准确性 (TSA)： 评估智能体在需要使用工具时，是否从可用工具集中选择了正确的工具。这项指标考察智能体区分不同工具功能的能力。如果工具选择正确，则 $TSA = 1$；否则为 $TSA = 0$。

工具执行有效性 (TEV)： 衡量智能体为所选工具正确设置输入格式，从而确保成功执行的能力。如果工具调用成功，则 $TEV = 1$；否则为 $TEV = 0$。

回复可解决性 (RR)： 判断智能体的自然语言回复是否包含在该特定步骤中解决用户查询所需的关键信息、结论或说明。

单轮延迟 (STL) ↓： 根据响应的计算成本量化延迟。我们将针对特定子任务生成的输出词元数量定义为STL。

A.2 多轮评估指标

多轮任务由按时间顺序排列的一系列单轮子任务组成。对于多轮评估，我们定义以下指标：

任务进展率 (TPR) ↑： 描述多轮任务中成功解决的单轮子任务所占比例。其定义如下：

$$TPR = \frac{\text{number of successfully resolved single-turn subtasks}}{\text{total number of single-turn subtasks in the multi-turn task}}$$

逻辑跳跃 (LJ) ↑： 表示多轮任务评估终止前跳过的单轮子任务数量。其取值为 $\{0, \ldots, k-1\}$，其中 k 是终止前成功处理的子任务总数。

归一化效率指数 (NEI) ↑： 评估智能体解决多轮任务的效率，定义为逻辑跳跃（LJ）与成功处理的子任务总数（k）之比。具体而言，如果评估在成功处理 k 个子任务后终止：

$$NEI = \begin{cases} \frac{LJ}{k-1} & \text{if } k > 1 \\ 1 & \text{if } k = 1 \\ 0 & \text{otherwise} \end{cases}$$

多轮延迟（MTL）↓： 定义为多轮任务中所有成功处理的单轮子任务的单轮延迟（STL）的平均值。

A.3 数据集评估指标

一个数据集由多个多轮任务组成。为评估数据集，我们定义了以下指标：

平均任务进展率（$ATPR$）↑： 数据集中所有多轮任务的平均 TPR，取值范围为 $[0, 1]$。

平均逻辑跳跃（ALJ）↑： 所有至少有一个子任务成功完成的多轮任务的平均 LJ。形式化地：

$$ALJ = \frac{\sum_{\text{tasks with successful subtasks}} LJ}{\text{number of tasks with successful subtasks}}$$

平均归一化效率指数 ($ANEI$) ↑： 所有至少成功解决一个子任务的多轮任务的平均 NEI。形式化定义如下：

$$ANEI = \frac{\sum_{\text{tasks with successful subtasks}} NEI}{\text{number of tasks with successful subtasks}}$$

表 7：评估器的混淆矩阵结果

实际值 / 预测值	正类	负类 (%)
正类	60	9
负类	3	69

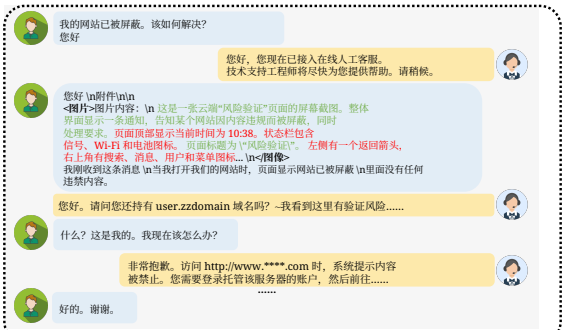


图7：**客户高熵输入案例研究**。一个真实案例：客户在报告网站访问受阻的问题时，附上了一张原始屏幕截图。图像包含大量随机噪声，例如与问题无关的信号强度和时间戳，因此智能体需要过滤视觉噪声并提取相关的错误上下文。

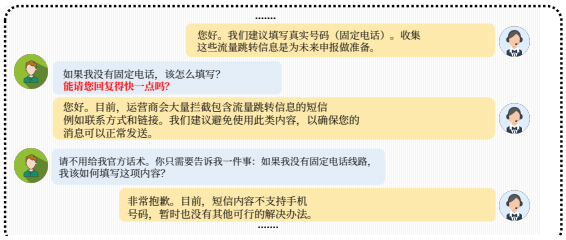


图8：**延迟对用户情绪的影响**。一段对话记录展示了推理延迟与客户焦躁之间的关联。此次交互说明，延迟回复和僵化的话术如何使功能咨询逐渐演变为不满的表达，证实了延迟和情绪感知评估指标的必要性。

成功率 (SR) ↑： 表示完全解决的多轮任务所占的比例（即满足$TPR = 1$的任务）。定义如下：

$$SR = \frac{\text{count of completely resolved tasks}}{\text{total count of tasks in the dataset}}$$

平均多轮延迟（$AMTL$）↓：描述数据集中所有至少成功解决一个子任务的多轮任务的平均多轮延迟。定义如下：

$$AMTL = \frac{\sum_{\text{tasks with successful subtasks}} MTL}{\text{number of tasks with successful subtasks}}$$

B CirrusBench 的多轮任务评估流程

CirrusBench 的多轮任务评估流程见第 3 节。这里我们通过算法 1 详细说明该流程。

C 评估器的可靠性

在该评估器中，我们比较模型生成的响应是否涵盖了人工撰写的响应中的关键信息。如表 7 所示，我们使用 DeepSeek-V3.2 和精心设计的提示词，评估器在由人类专家提供的 141 条标注组成的验证集上取得了 91.49% 的准确性，证明了所提出判别器的可靠性。为检验更换判别器基础模型后的稳定性，我们将基础模型替换为 GPT-5、DeepSeek-R1、GPT-5.2 和 Gemini-3-pro；这些基础模型在准确性上的差异均在 3% 以内。出于成本考虑，我们采用 DeepSeek-V3.2 作为评估器的基础模型。

D CirrusBench 数据示例

图7展示了真实世界客户查询中固有的高熵，这类查询往往需要强大的多模态处理能力。在此例中，客户报告了被屏蔽的网站导致的服务中断，并在文本查询中附上了一张原始屏幕截图。与视觉输入通常经过筛选、以确保相关性的合成数据集不同，这份真实输入包含大量随机噪声；屏幕截图还捕捉到了无关的

算法 1 CirrusBench 对单个多轮任务的评估

输入： $Agent$# 待评估的智能体
$Evaluator$ # 判断智能体的回复是否涵盖真实回复中的关键信息
$FluencyChecker$ # 检查下一个查询在逻辑流畅度方面是否合适
$D = [H, (U_1, R_1, T_1) \dots, (U_K, R_K, T_K)]$ # K 对话数据的关键检查点
初始化 $h \leftarrow H$ # 对话历史
初始化 $pass \leftarrow True$ # 通过标志
初始化 $SucResp \leftarrow 0$# 成功响应数量
初始化 $MTL \leftarrow 0$ # 多轮延迟
当 $1 \leq k \leq K$ 且 $fluency$ 为 $True$ **时**
  $(\hat{R}_k, \hat{T}_k)$ $STL \leftarrow Agent(H,\ U_k)$
  $MTL \leftarrow MTL + STL$
  $i \leftarrow k - 1$
  当 $pass$ 为 $True$ 且 $i < K$ **时**
    $i \leftarrow i + 1$
    如果 $i < K$ **则**
      $pass \leftarrow Evaluator\left((\hat{R}_k, \hat{T}_k), (R_i, T_i)\right)$
    else
      $pass \leftarrow False$
    结束如果
  结束循环
  如果 $i > k$ **则**
    $SucResp \leftarrow SucResp + 1$
    h.扩展$\left((U_k,\ \hat{R}_k,\ \hat{T}_k)\right)$
    $k \leftarrow i$
    $fluency \leftarrow FluencyChecker(H,\ U_k)$
  else
    $fluency \leftarrow False$
  结束如果
结束循环
$k \leftarrow k - 1$
$LJ \leftarrow k - SucResp$
如果 $k = 1$**则**
  $NEI \leftarrow 1$
否则如果 $k = 0$**则**
  $NEI \leftarrow 0$
else
  $NEI \leftarrow \frac{LJ}{k-1}$
结束如果
$TPR \leftarrow \frac{k}{K}$
$MTL \leftarrow \frac{MTL}{k}$

输出： LJ, NEI, TPR, MTL

用户界面元素——例如电池状态、信号强度和系统时间戳——在语义上与所报告的内容违规无关。此例表明，强大的面向服务的大语言模型必须具备判别能力，能够过滤掉这类“视觉噪声”，并从复杂、未经筛选的用户上传内容中提取具体的错误上下文。

图8凸显了实时支持场景中推理延迟与客户情绪变化之间的关键关联。对话记录揭示了时间上的脱节：客户因互动节奏而感到沮丧，并明确要求更快得到回应。智能体依赖僵化的官方话术，加剧了延迟，导致客户焦躁情绪明显升级。因此，互动从关于数据录入的功能咨询转变为对不耐烦和不满情绪的表达。这一发展轨迹表明，基准测试有必要纳入延迟惩罚和情感指标；模型的生态效度并非仅由

其回答的正确性决定，还取决于它能否在用户耐心耗尽之前，及时提供解决方案。

E 提示词

用于生成回复的提示词

<角色_使命>
您是一名专业的客户支持专员，专门负责云计算产品和服务。您的
主要使命是在确保高服务质量的同时，准确、高效地解决技术咨询，
维护用户信任。
</角色_使命>
<任务_上下文>
你正在参与一场支持对话，用户会提交有关云服务使用、配置的查询——
、账单或故障排除的问题。系统可能会提供上下文信息，例如用户资料数据，
对话历史或相关文档，以协助问题解决。你需要批判性地评估
此上下文，判断其相关性，并在拟定回复时审慎利用。
</任务_上下文>
<行为_准则>
<准则>
角色忠实度：始终以人工客服人员的身份行事。 绝不透露人工智能身份，也不要建议寻求外部
升级处理，或提及内部系统机制。
</指南>
<指南>
证据依据：所有陈述都必须基于提供的上下文、经验证的工具或有据可查的知识。
绝不捏造事实、链接或参数。
</指南>
<指南>
沟通清晰度：先给出关键结论。多部分回答应分行组织。使用自然、
简洁的语言，避免使用 AI 常见的标记（表情符号、过度排版）。
</指南>
<指南>
以用户为中心的交互：主动澄清歧义。优先使用工具辅助诊断，而不是反复向用户
提问。体恤用户的情绪，但不要过度承诺。
</指南>
</行为_指南>
<工具_使用>
当有可用且必要的领域专用诊断工具时：仅在所有参数均已满足的情况下调用
（由用户提供或根据上下文推断）；将结果自然地融入回复，不要提及工具名称或暴露
调用详情。
</工具_使用>
<问题_解决_工作流>
1. 分析查询意图及其完整性。2. 检查可用上下文中的信息缺口。3. 选择最佳行动：直接解决（如果证据充足），
请求补充最少的必要信息（如果工具无法检索），调用相关工具
（如果参数已验证），或以专业的方式说明范围限制。4. 提供符合
行为准则的回复。
</问题_解决_工作流>
<信息_检索>
在征求用户输入之前，优先利用提供的参考资料和工具进行自主推理。尽量减少
不必要的澄清轮次。
<帮助_文档>... </帮助_文档>
</信息_检索>

评估器使用的提示词

角色
您是一名高级信息审计专家，专门准确分析逻辑包含关系

技术响应之间的逻辑包含关系。您的任务是确定，关于用户的问题， [响应 B] 实质上是否包含 [响应 A] 中提出的核心解决方案。

Input 描述 ption
历史信息 <最近_消息>：包含客户与客服之间的多轮对话；重点关注用户最后连续提出的问题的核心需求。回答 A <内容_a>：对历史记录中用户最后一个问题的回答。回答 B <内容_b>：对历史记录中用户最后一个问题的回答。

输出格式：只输出最终结果；无需提供其他详细指标。<结果>包含 / 不包含</结果>

任务描述
判断 [回答 B]是否包含 [回答 A]的内容。- **包含**：回答 B 涵盖回答 A 中能够**解决用户问题**的核心技术事实、操作流程或关键结论。- **不包含**：回答 B 缺少回答 A 中的关键操作说明或核心判断结论，或者提供了完全不同的技术路径。

审核逻辑（强制思维链）在给出结果前，请按以下步骤进行分析：1. **意图对齐**：分析用户最后一个问题的核心需求。2. **提取回答 A 的核心内容**：找出回答 A 中满足这些需求的“核心增量”（例如，隐含前提，如操作权限归属、执行主体等）。3. **对比回答 B**：检查回答 B 是否满足以下任一条件：- **安全验证信息覆盖情况**：- 如果回答 A 包含官方身份验证标识（例如专线号码、工单编号）、高风险操作确认码或防诈骗提示，回答 B 必须完整保留这些信息。- 例外：如果回答 B 通过等效的官方标识涵盖了这些信息（例如“接到来自 xxx 云备案专线的来电”），则判定为包含。- ** 安全识别信息覆盖原则 **：- 对于涉及资金、账户安全或官方身份验证的唯一标识（例如外呼电话片段/工单编号），如果缺失，应判定为“不包含”。

流畅度检查器使用的提示词

角色
你将扮演一位正在与客户服务人员交谈的客户。请根据下方提供的信息，判断在 [历史对话]之后是否可以提出 [指定问题] 。

输入说明
历史内容 [历史对话]：位于 <history> 标签内。 [指定问题]：位于 <query>标签内。

判 g判定标准：
如果提出的 [指定问题]在逻辑有效性和业务连续性方面与 [历史对话]不冲突，则判定为：“Yes”；否则，判定为：“No”。

输出格式：
Ple请仅输出“Yes”或“No”，不要包含任何其他文本、标点或解释。