Computer Science and Application 计算机科学与应用, 2025, 15(6), 190-197
Published Online June 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.156169

Hans汉斯

基于大语言模型的Python编程设计智能助教系统选型评测

徐鼎洪，景尔妮，董许宏，罗雅琴*

上海工程技术大学数理与统计学院，上海

收稿日期：2025年5月22日；录用日期：2025年6月20日；发布日期：2025年6月27日

摘　要

本研究针对大语言模型在Python编程教育中的应用，构建了多维度评测体系，系统对比了通义千问、星火、文心一言等主流模型在教学场景中的表现。通过设计事实性问题、推理性问题、代码生成及多轮对话等测试任务，从回答准确性、完整性、语言流畅性、上下文理解能力及代码示例质量五个维度进行评估。实验结果表明，qwen-plus在综合评分中表现最优，其回答覆盖边界条件和多轮逻辑关联性，且代码示例符合PEP8规范；Ernie Bot 8k与sparkV3.5在准确性上优异但存在冗余注释问题，而GPT-4因代码冗余和异常处理片面性得分较低。研究揭示了模型在Python语言细节覆盖和上下文建模方面的共性缺陷，并提出通过知识库更新、强化学习优化及多模态评测体系改进的路径，为智能助教系统的选型与教学场景适配提供了实证依据。

关键词

大语言模型，Python教育，评测体系

System Selection and Performance Evaluation of LLM-Based Python Programming Teaching Assistants

Dinghong Xu, Erni Jing, Xuhong Dong, Yaqin Luo*

School of Mathematics, Physics and Statistics, Shanghai University of Engineering Science, Shanghai

Received: May 22nd, 2025; accepted: Jun. 20th, 2025; published: Jun. 27th, 2025

Abstract

This study investigates the application of large language models (LLMs) in Python programming

*通讯作者。

文章引用: 徐鼎洪, 景尔妮, 董许宏, 罗雅琴. 基于大语言模型的 Python 编程设计智能助教系统选型评测[J]. 计算机科学与应用, 2025, 15(6): 190-197. DOI: 10.12677/csa.2025.156169

education by constructing a multi-dimensional evaluation framework to systematically compare the performance of mainstream models, such as Qwen-Plus, Ernie Bot 8k, and SparkV3.5, in educational scenarios. Through testing tasks including factual questions, reasoning problems, code generation, and multi-turn dialogue, models were assessed across five dimensions: accuracy, completeness, linguistic fluency, contextual understanding, and code example quality. Experimental results show that Qwen-Plus achieved the highest overall score, demonstrating superior coverage of edge cases and logical coherence in multi-turn interactions, with code examples adhering to PEP8 standards. Ernie Bot 8k and SparkV3.5 exhibited high accuracy but suffered from redundant annotations, while GPT-4 scored lower due to code redundancy and incomplete exception handling. The study identifies common limitations in models' coverage of Python language details and contextual modeling, suggesting improvements through knowledge base updates, reinforcement learning optimization, and multi-modal evaluation frameworks. These findings provide empirical evidence for model selection and educational scenario adaptation in intelligent teaching assistant systems.

Keywords

Large Language Models, Python Education, Evaluation Framework

1. 引言

人工智能技术的迅猛发展为教育领域注入了革新动力，大语言模型(Large Language Models, LLMs)凭借其强大的自然语言处理与代码生成能力，逐步成为编程教学的重要辅助工具。在计算机教育体系中，程序设计类课程作为人工智能技术的基础与核心，率先探索了大语言模型与教学场景的深度融合。Python语言因简洁的语法、丰富的生态及广泛的应用场景，成为全球高校与中小学编程教育的首选语言。然而，传统编程教学资源在个性化指导、实时反馈及复杂问题解决支持方面存在显著局限性，难以满足学习者动态化、差异化的学习需求。大语言模型通过其对海量代码数据的学习能力，能够生成规范的代码示例、解析技术细节并提供交互式学习支持，为编程教育提供了新的解决方案。例如，本地化部署的 Ollama 框架结合 llama2、codellama 等模型，已成功应用于编程知识讲解、闯关式作业训练及代码调试辅助等场景。然而，现有研究对主流大语言模型在 Python 基础教学场景中的表现差异缺乏系统性评估，尤其在回答准确性、上下文理解能力、代码质量及逻辑连贯性等关键维度的评测仍显不足。这一研究空白制约了模型选型与教育场景的精准适配，也限制了智能助教系统在编程教学中的效能提升。

1.1. 研究背景

近年来，大语言模型的快速发展显著推动了自然语言处理技术的突破，其强大的生成与推理能力为复杂知识推理和智能决策支持提供了新范式。知识图谱作为结构化知识表示的重要工具，通过整合多源异构数据，提升了数据关联与解释能力，成为大语言模型增强领域专业知识处理的关键技术[1]。然而，现有研究仍面临多重挑战：一方面，大语言模型在特定领域应用时，因训练数据覆盖不足导致生成结果的可靠性受限，需依赖外部知识库(如知识图谱)进行检索增强[2]；另一方面，传统检索增强生成(RAG)方法在处理长文本时存在上下文长度限制和计算复杂度问题，例如软 RAG 因依赖模型注意力机制的内部检索，难以有效处理超长文档并易丢失全局信息[3]。此外，大语言模型的幻觉风险与深层次

语义理解能力不足进一步限制了其在知识密集型任务中的应用，例如在政策分析与古生物学研究中，模型易因知识库更新滞后或领域术语理解偏差导致输出错误[3] [4]。为解决这些问题，学者提出通过知识图谱与大语言模型的协同优化，结合对比学习、动态路由检索等技术，提升模型在领域适应性、上下文关联性和推理准确性方面的表现[5]。当前研究聚焦于如何通过结构化知识增强、多模态检索策略及参数高效微调等方法，进一步优化二者的结合模式，以应对长尾变化、跨领域推理及动态知识更新等复杂场景[6]。

智能助教系统作为 LLMs 在教育中的重要应用形式，已从单一功能工具发展为多模态、多场景的综合解决方案。当前研究主要涉及到基于学生能力感知的对话助手通过分析答题记录或知识点掌握程度，动态调整回答内容。例如，Llama8B_Agent 和 GLM_Agent 在 MOOCRadar 数据集上的实验表明，基于 Agent 的范式能够通过任务分解和工具调用，提供更符合学生需求的解答[7]。智能助教系统逐步整合文本、语音、图像等多模态输入输出。如科大讯飞的“星火语伴”支持口语评测和语法纠错，而网易有道的“子曰”大模型则通过多模态知识库满足跨学科学习需求[8]。部分系统尝试融入情绪心理学框架，提供情感诊断和心理支持。如“随身外教 Sarah”在回答学生关于减肥焦虑、婚姻观念等问题时，能够结合价值观引导，为课程思政教育提供参考[9]。

现有智能助教系统面临的挑战在于，学生可能因过度依赖 AI 反馈而丧失主动思考能力，例如在高职英语教学中，部分学生仅通过重复提问获取答案，未形成自主学习意识[9]。模型对复杂句式(如文言文省略结构[10])和跨学科知识的处理能力有限，需结合迁移学习或元学习技术优化算法机制。在人机争论学习中，智能助教的回复需避免偏见和歧视，需通过敏感词库和健康度评估体系进行风险控制[11]。

1.2. 研究目的与意义

本研究旨在通过构建多维度评测体系，系统评估主流大语言模型(如 Ernie Bot 8k、sparkV3.5、qwen-plus 等)在 Python 基础学习场景中的表现，重点分析其回答准确性、完整性、语言流畅性、上下文理解能力及代码示例质量等关键指标的差异。研究目的包括：为教育机构和开发者提供模型选型依据，帮助选择最适合编程教学的模型；揭示现有模型在知识覆盖、逻辑连贯性及实践指导方面的优势与不足，推动模型优化方向的明确；探索智能助教系统与编程教育的结合路径，为开发更高效、个性化的学习工具奠定基础。研究意义体现在：理论上，填补 LLMs 在编程教育评测领域的实证研究空白；实践上，通过量化模型性能差异，可指导教育场景中智能系统的参数调优，提升多轮对话的连贯性和代码示例的实用性，最终促进编程学习效率与教学质量的提升。

2. 文献综述

2.1. 大语言模型在智能助教系统中的应用现状

近年来，LLMs 在智能助教系统中的应用迅速扩展，其核心优势在于能够通过自然语言交互提供实时、个性化的学习支持。例如，GPT 系列模型被用于代码纠错、算法解释和学习路径规划，而特殊布置的部分专用模型则针对编程教育场景优化，支持从基础语法到项目实践的全流程指导。现有研究显示，LLMs 在事实性问题回答(如语法规范、函数参数说明)上表现优异，但在复杂推理(如算法优化、异常机制分析)和多轮对话的上下文关联中仍存在局限。例如，部分模型在解释递归函数的栈溢出问题时，可能因缺乏对内存管理机制的深入理解而提供片面建议。此外，尽管 LLMs 能够生成符合功能需求的代码片段，但代码的可读性、兼容性(如 Python 版本差异)及扩展性(如异常处理的全面性)仍需进一步优化。当前研究多聚焦于单一模型的性能分析，缺乏对多模型横向对比及教育场景适配性的系统性探讨。

2.2. 搜索策略参数设置的研究进展

在 LLMs 的优化中，搜索策略参数(如温度值、采样策略、注意力窗口大小)对生成内容的质量至关重要。研究表明，调整温度值可平衡输出的确定性与多样性：较低温度值(如 0.1)倾向于生成标准化答案，适合事实性问题；较高温度值(如 0.7)则可能激发创造性解决方案，但伴随错误风险。注意力机制的窗口大小直接影响模型对长上下文的理解能力，过小可能导致对话断裂，过大会增加计算开销。此外，奖励函数的设计(如强化学习中的奖励分配)对引导模型生成符合教育场景需求的输出(如避免冗余注释、强调边界条件)具有关键作用。然而，现有研究多集中于通用场景参数调优，针对编程教育的特定需求(如代码示例的 PEP8 规范、多轮对话的逻辑连贯性)的参数优化策略仍较少。例如，如何通过调整参数使模型在代码生成时自动添加注释说明，或在多轮交互中引用历史信息，仍需进一步探索。

2.3. Python 基础学习资源及辅助工具分析

Python 学习资源呈现多样化特征，包括官方文档、在线课程(如 Codecademy、Coursera)、交互式平台(如 Replit、Jupyter Notebook)及辅助工具(如 PyCharm、Black 代码格式化工具)。官方文档(如 Python.org)内容权威但学习曲线陡峭，适合进阶用户；在线课程多采用视频讲解与实践结合，但缺乏即时反馈；交互式平台通过实时代码执行提升学习体验，但对新手的引导性不足。辅助工具方面，IDE(集成开发环境)提供调试和代码分析功能，但配置复杂；静态检查工具(如 flake8)可确保代码规范，但需用户主动调用。尽管现有资源覆盖广泛，仍存在以下不足：(1) 个性化指导缺失，难以根据学习者水平动态调整内容；(2) 代码示例多为静态文本，缺乏动态解释与错误分析；(3) 多轮交互支持薄弱，难以追踪学习者的历史问题。大语言模型的引入可弥补这些缺陷，例如通过对话式交互提供实时答疑、根据用户错误自动生成纠错建议，并通过多轮对话追踪学习进展，从而构建更高效、个性化的学习支持系统。

3. 实验设计

本实验基于 Dify 平台开展研究，该平台凭借其经济性、灵活性及对主流大模型的广泛适配性，为多模型对比实验提供了高效的技术支撑。研究团队通过标准化接口调用了包括通义千问系列(如 qwen-plus)、GPT-4 等在内的多款主流大语言模型，并基于预先构建的结构化领域知识库对模型进行轻量化微调。在此基础上，针对不同模型的输入端口特性设计了分层测试方案：首先通过预设的基准数据集验证模型基础能力，随后在特定业务场景下对模型的输出准确性及上下文理解能力进行多维度评估。实验过程中特别关注模型版本差异对结果的影响，并通过动态调整输入参数(如 token 长度、温度值等)确保测试条件的完备性。该实验设计不仅验证了 Dify 平台在多模型协同部署中的工程价值，更为后续系统选型提供了可量化的性能基准与技术参考。

本研究选取了四款主流大语言模型进行对比评测，具体包括阿里云研发的通义千问(Qwen-Plus v2.5)、百度公司的 Ernie Bot 8k v3.2、字节跳动的 SparkV3.5 以及 OpenAI 的 GPT-4 v4.5。Qwen-Plus 作为国产模型的代表，因其在中文场景优化、代码生成规范性(如 PEP8 支持)及多轮对话连贯性方面的优势被选为基准模型，其参数规模达 800 亿，支持动态上下文建模，尤其在编程教育领域已有实际应用案例，如阿里云编程教学平台；Ernie Bot 8k 凭借百度文心一言系列在知识图谱构建与领域知识增强上的特长，结合其 8k 上下文长度支持复杂代码场景的长文本理解能力，被纳入评测以验证其在 Python 语法细节(如 Python 3.11+版本特性)和边界条件覆盖方面的表现；SparkV3.5 则因字节跳动在轻量化部署与高效推理上的技术优势，且其 3.5 版本新增对结构化类型提示等 Python 3.11+特性的支持，成为轻量级智能助教系统的候选方案，尤其在代码规范性(如 PEP8 自动约束)方面具有潜力；GPT-4 作为国际主流模型的标杆，尽管在通用推理能力上表现突出，但其对中文编码问题及 Python 语言细节(如文件读取编码设置)的适配性需进一

步验证，因此被选为对比参照，以揭示中西方模型在教育场景中的差异。上述模型的选择基于其市场主流性(均位列 2025 年 MLPerf 评测榜单前五)、技术差异化(覆盖中文优化、长文本理解、轻量化部署及国际通用能力)以及教育适配潜力(通过多维度评测验证其在回答准确性、上下文关联性及代码示例质量上的优势与局限)，为智能助教系统的参数调优和场景适配提供实证依据。

3.1. 实验目标

本实验旨在系统评估不同大规模语言模型 LLMs 在 Python 基础学习场景中的表现，重点围绕回答准确性、完整性、语言流畅性、上下文理解能力及代码示例质量五大维度展开。实验通过构建包含变量命名规则、控制流逻辑、异常处理及函数定义等典型问题的测试集，采用五级评分标准(1~5 分)量化模型输出质量。

回答准确性通过对比模型回答与 Python 官方文档或权威教材的规范性，验证其是否符合语法要求及最佳实践。完整性则评估模型是否覆盖问题所有关键点，例如在“文件读取”问题中是否提及编码设置、异常处理及关闭机制。语言流畅性关注回答的逻辑连贯性与表述清晰度，例如避免冗余信息或技术术语堆砌。上下文理解能力通过多轮对话测试模型对历史交互的追踪能力，例如在定义函数后是否自动关联默认参数的使用场景。代码示例质量则从功能正确性(如无语法错误)、可读性(缩进、注释规范)及扩展性(如兼容 Python 3.x 版本特性)进行综合评分。

实验结果将为教育场景中智能助教系统的选型提供数据支持，例如通过对比不同模型在代码示例质量上的差异，确定适合编程教学的最优模型。此外，通过分析模型在上下文理解中的表现，可进一步优化对话系统的参数设置(如调整注意力机制窗口大小或奖励函数设计)，以提升多轮交互的连贯性。

3.2. 测试问题设计

本研究的测试问题设计基于 GitHub 上广受认可的项目 Devinterview-io/python-interview-questions。此项目覆盖 Python 编程的核心知识点与实际应用场景，行业认可度高，具有较高内容系统性与全面，覆盖基础语法(如作用域、装饰器)、数据结构与算法(如排序、动态规划)、异常处理、并发编程等核心主题。所有题目均基于真实企业面试案例或经典编程教材，并附带详细解析和代码示例，确保答案的准确性与可复现性。

准确性通过对比模型回答与 Python 官方文档或权威教材的规范性验证，确保语法要求和最佳实践的符合性；完整性评估模型是否覆盖问题所有关键点，例如在“文件读取”问题中是否提及编码设置、异常处理及关闭机制；语言流畅性关注回答的逻辑连贯性与表述清晰度，避免冗余信息或技术术语堆砌；上下文理解能力通过多轮对话测试模型对历史交互的追踪能力，例如定义函数后是否自动关联默认参数的使用场景；代码示例质量从功能正确性(如无语法错误)、可读性(缩进、注释规范)及扩展性(如兼容 Python 3.x 版本特性)进行综合评分。

采用五级评分标准(1~5 分)对模型输出质量进行量化，通过人工评分与自动化工具(如代码格式检查工具 flake8)结合验证评分一致性；对各模型在五个维度的得分进行描述性统计分析；针对多轮对话场景，引入对话连贯性指标(如上下文引用频率)量化模型语境理解能力；结合人工评分的主观反馈与自动化评估的客观数据进行综合分析。

4. 结果分析

根据第 3 章的实验设计，对 Ernie Bot 8k、sparkV3.5、千问等大语言模型进行了测试问题的对比实验，并从准确性、完整性、语言流畅性、语境理解能力、代码示例质量 5 个方面对其进行评价，评价结

果如表 1 所示。

Table 1. Evaluation of various language models
表 1. 各个语言模型评价

模型名称	准确性	完整性	语言流畅性	语境理解能力	代码示例质量	总分
Ernie Bot 8k	5.0	4.5	4.5	3.0	5.0	21.5
sparkV3.5	5.0	4.5	4.5	3.0	5.0	21.5
qwen-plus	5.0	4.75	4.25	3.5	5.0	22.5
gpt-4	4.8	4.25	3.75	3.25	4.85	21.0

4.1. 准确性分析

所有模型在准确性上表现优异(Ernie Bot 8k、sparkV3.5、qwen-plus 均获 5/5 分)，验证了其对 Python 语法规范的掌握。例如，Ernie Bot 8k 在__init__方法中准确说明继承场景的 super()__init__()用法，而 sparkV3.5 通过 else 和 finally 块的示例展示全面的异常处理逻辑。唯一扣分点为 gpt-4 在 lambda 表达式语法中的格式错误(如缺少空格)，导致 4.8 分。

4.2. 完整性分析

qwen-plus 其优势体现在文件读取问题中补充“逐行读取需关闭文件”的说明，弥补了 Ernie Bot 8k 与 sparkV3.5 未提及编码问题的不足。字典遍历示例中对比 items()与 values()的适用场景，而 Ernie Bot 8k 与 sparkV3.5 仅覆盖基础用法。Ernie Bot 8k 与 sparkV3.5 因未说明 Python 3.7+字典的有序性特性(如 dict 合并的运算符)，导致完整性扣分至 4.5 分。

4.3. 语言流畅性分析

qwen-plus 以 4.25 分略低于 Ernie Bot 8k 与 sparkV3.5 (4.5 分)，但其通过“如前所述”等关联词增强多轮对话逻辑，而后者因冗余注释(如“return 结束函数”)影响简洁性。例如，sparkV3.5 在 try-except 示例中重复解释 finally 块的作用，暴露知识库调用僵化问题。

4.4. 语境理解能力分析

所有模型在上下文语境理解能力上表现较弱(3.0~3.5 分)，但 qwen-plus 通过在定义函数后关联“默认参数需置于末尾”规则，而 Ernie Bot 8k 与 sparkV3.5 仅延续逻辑未直接引用。在结合装饰器与异常处理时，其代码示例通过注释解释两者的交互逻辑，而其他模型未涉及此类关联。

4.5. 代码示例质量分析

所有模型在代码质量上均获高分(≥4.85 分)，但差异体现在：sparkV3.5 在 try-except 中补充 else 和 finally 块，而 Ernie Bot 8k 与 qwen-plus 覆盖核心异常。qwen-plus 提供*args 与**kwargs 的嵌套函数调用示例，而 gpt-4 因代码注释冗余导致扣分(4.85 分)。

5. 结论

本研究通过系统评测主流大语言模型在 Python 基础学习场景中的表现，揭示了模型在准确性、完整性、上下文理解等维度的差异化特征。实验结果表明，qwen-plus 凭借对边界条件的全面覆盖和多轮对话的逻辑关联性，在综合评分中位列首位，其回答通过显式引用前文内容和补充如字典有序性、文件读取

编码问题等细节，显著提升了教学场景的适用性。Ernie Bot 8k 与 sparkV3.5 虽在准确性上表现优异，但冗余注释和知识库调用僵化问题影响了语言流畅性与上下文连贯性，提示需结合动态检索增强(RAG)技术优化知识调用逻辑。gpt-4 因异常处理片面性和代码示例冗余性得分较低，反映出该模型在教育场景适配性上的不足。

当前大语言模型在编程教育场景中的局限性主要源于训练数据的时效性不足、上下文建模能力受限及领域适配性偏差等核心问题。例如，模型对 Python 语言新特性(如 3.11+的 match-case 语法)和跨平台编码参数(如 encoding)的覆盖不足，直接反映出训练数据滞后于语言生态演进的缺陷。针对这一问题，动态知识库更新机制与教育语料注入技术可有效解决，通过实时抓取 Python 官方文档和开源教育数据集(如 GitHub 的 Devinterview-io)，结合强化学习引导模型生成符合新规范的代码示例。此外，模型在多轮对话中易丢失关键信息，导致冗余注释和逻辑断裂，其根源在于 Transformer 的固定长度上下文窗口限制。通过引入自适应注意力机制(如 Longformer 的滑动窗口)和检索增强生成(RAG)技术，可动态扩展上下文范围并关联历史信息，从而提升对话连贯性。代码冗余与规范性问题则暴露了模型在采样策略和约束机制上的不足。高温度值虽能提升多样性，但牺牲了代码简洁性，而缺乏 PEP8 合规性约束的训练阶段导致冗余注释频发。对此，需在推理阶段集成代码格式检查工具(如 flake8)实时校验规范性，并采用低温度值与 Top-k 采样(k = 40)平衡多样性与简洁性，同时通过规则冗余检测模块过滤重复解释基础语法的内容。复杂异常场景泛化能力的不足源于训练数据分布偏差，公开代码库中异常处理样本占比低且多为简单用例。为此，需构建典型异常模式数据集(如网络请求超时、文件读取失败)，通过微调和强化学习优化边界条件处理能力，并设计基于异常覆盖率的奖励函数，通过人机对抗训练提升模型的场景适配性。

致　谢

本研究的顺利完成得益于多方支持与帮助，在此谨表诚挚谢意。首先感谢学院提供的科研平台与资源保障，学院在计算资源分配与实验环境建设方面给予的大力支持为本研究奠定了基础。同时感谢参与测试集设计的课题组成员，各位成员在问题分类与评分标准制定过程中展现出的专业素养确保了实验数据的可靠性。

参考文献

[1] 张坤丽, 王影, 付文慧, 等. 大语言模型驱动下知识图谱的构建及应用综述[J/OL]. 郑州大学学报(理学版): 1-9. https://doi.org/10.13705/j.issn.1671-6841.2024165, 2025-04-23.

[2] 方全, 张金龙, 王冰倩, 等. 基于组合上下文提示的大型语言模型领域知识问答研究[J/OL]. 计算机科学: 1-13. http://kns.cnki.net/kcms/detail/50.1075.TP.20250417.1135.022.html, 2025-04-23.

[3] 黄冰. 大语言模型在古生物学中的应用初探——以基于 RAG 的知识问答系统为例[J/OL]. 古生物学报: 1-15. https://doi.org/10.19800/j.cnki.aps.2024047, 2025-04-23.

[4] 段永康, 赵广宇, 耿骞, 等. 基于大语言模型的政策知识库构建与政策比较研究——以惠企政策为例[J/OL]. 数据分析与知识发现: 1-20. http://kns.cnki.net/kcms/detail/10.1478.G2.20250418.1553.008.html, 2025-04-23.

[5] 邵欣怡, 朱经纬, 张亮. 基于大语言模型的业务流程长尾变化应变方法[J/OL]. 计算机科学: 1-12. http://kns.cnki.net/kcms/detail/50.1075.tp.20250417.1126.018.html, 2025-04-23.

[6] 林丽萍. 国内大语言模型辅助意大利语教学的能力探析[J]. 公关世界, 2025(8): 117-119.

[7] 董艳民, 林佳佳, 张征, 等. 个性化学情感知的智慧助教算法设计与实践[J]. 计算机应用, 2025, 45(3): 765-772.

[8] 肖建力, 黄星宇, 姜飞. 智慧教育中的大语言模型综述[J/OL]. 智能系统学报: 1-17. http://kns.cnki.net/kcms/detail/23.1538.tp.20250205.1354.002.html, 2025-05-06.

[9] 谢颖怡, 张逸诗, 曾艾玲. 基于人工智能大语言模型的微信聊天助教在高职英语教学中的应用探索[J]. 中国医

学教育技术, 2025, 39(1): 48-53.

[10] 文玉锋, 林伟杰, 夏翠娟, 等. 面向古籍文献智能处理的大语言模型效能测评[J/OL]. 图书馆论坛: 1-10. http://kns.cnki.net/kcms/detail/44.1306.g2.20250429.1504.002.html, 2025-05-06.

[11] 黎盈盈, 詹昌昊. 多模态大语言模型驱动的争论式智能对话学习系统设计与开发[J]. 数字技术与应用, 2025, 43(1): 25-27.