

大语言模型背景下Python程序设计课程教学改革探索

李莉¹, 郭祥云², 吕金娜¹, 代晓明², 焦明江³, 郝保水¹

(1. 北京信息科技大学 计算机学院; 2. 北京信息科技大学 管理科学与工程学院;

3. 北京信息科技大学 马克思主义学院, 北京 102206)

摘要: 以ChatGPT为代表的生成式人工智能对教育领域正引发的一场认知革命, 针对生成式人工智能对Python程序设计等课程带来新的冲击等问题, 在全面分析生成式人工智能对Python程序设计课程带来的挑战和机遇的基础上, 深入剖析Python程序设计课程面临的新情况、新问题, 从课程内容体系重构、教学模式创新、教学资源与工具创新以及教学评估体系创新4个方面, 提出针对性强、可操作性高的应对策略建议。教学实践表明, 将大语言模型融入Python程序设计课程教学改革, 不仅取得了良好的教学效果, 也获得了学生的积极反馈。

关键词: ChatGPT; 大语言模型; Python程序设计; 教学改革

DOI: 10.11907/jjdk.251590

中图分类号: G642

文献标识码: A

文章编号: 1672-7800(XXXX)0XX-0001-05

扫描二维码阅读全文:



Exploration of Teaching Reform in Python Programming Course under the Background of Large Language Models

LI Li¹, GUO Xiangyun², LYU Jinna¹, DAI Xiaoming², JIAO Mingjiang³, HAO Baoshui¹

(1. College of Computer Science, Beijing Information Science and Technology University;

2. College of Management Science and Engineering, Beijing Information Science and Technology University;

3. College of Marxism, Beijing Information Science and Technology University, Beijing 102206, China)

Abstract: In response to the cognitive revolution triggered by generative artificial intelligence represented by ChatGPT in the field of education, especially the new challenges it brings to programming courses such as Python programming, this study comprehensively analyzes the challenges and opportunities generative AI poses to Python programming courses. It deeply examines the new situations and issues faced by these courses, and from four aspects—restructuring the course content system, innovating teaching models, innovating teaching resources and tools, and innovating teaching assessment systems—proposes targeted and highly practical strategies and recommendations. Teaching practice has shown that integrating large language models into the teaching reform of Python programming courses has not only achieved excellent teaching results but also received positive feedback from students.

Key Words: ChatGPT; large language models; Python program design; teaching reform

0 引言

自2022年底发布以来,以ChatGPT为代表的大语言模

型凭借其强大的信息整合和自然语言理解能力,在全球范围内引发了广泛关注。特别是2025年1月20日国内的深度求索公司开发DeepSeek-R1推理模型以其强大的中文理解与生产能力、开源免费和高性价比等特性,引发全球

收稿日期: 2025-10-23

基金项目: 北京信息科技大学教学改革项目(2023JGYB30, 2023JGZD04); 国家社会科学基金高校思政课研究专项(24VZ048)

作者简介: 李莉(1981-),女,北京信息科技大学计算机学院副教授,研究方向为人工智能、大数据分析、时空数据挖掘;郭祥云(1981-),女,北京信息科技大学管理科学与工程学院副教授,研究方向为机器学习、深度学习;吕金娜(1981-),女,北京信息科技大学计算机学院副教授,研究方向为计算机视觉、多模态数据处理、人工智能;代晓明(1978-),女,北京信息科技大学管理科学与工程学院讲师,研究方向为大数据分析、决策支持;焦明江(1987-),男,北京信息科技大学马克思主义学院讲师,研究方向为思想政治教育;郝保水(1976-),男,北京信息科技大学计算机学院讲师,研究方向为计算机应用技术、人工智能。

AI行业关注,震撼了科技界,同时各大高校都纷纷接入了本地版本的DeepSeek-R1。Python语言是目前十分流行的高级编程语言之一,不仅高校计算机相关专业,而且管理类、文科专业都将Python程序设计语言开设为一门必修课程。ChatGPT等大语言模型不仅仅在对话领域展现出卓越的性能,在代码生成领域同样表现非凡。大语言模型通过强大的自然语言处理能力,能够理解和解析复杂的程序代码,解答研发人员的各种程序开发问题,甚至能根据用户注释和自然语言自动生成代码片段,使得编程过程变得更加智能、高效和便捷^[1]。目前,市面上出现了许多辅助编程的大模型工具,涵盖了代码补全、代码生成、代码解释、调试优化等多种功能。国外的主流AI编程助手有GitHub Copilot、Windsurf、Cursor等,国内的主流编程助手有字节跳动的Trae、阿里的Qwen2.5和智谱的GLM等。其中,Cursor由于集成了GPT-4、Claude 3.5/3.7等大模型,生成的代码质量高,且具备让AI自动运行命令、调试代码的功能,受到广大编程爱好者的追捧,利用AI编程助手生成代码已经成为程序员和科研工作者的首选。因此,国内很多社交媒体如知乎、微博和小红书上经常会出现“大模型要取代程序员”“有了大模型还需要学习编程吗?”等言论,无疑给Python程序设计等编程语言的教学工作带来了极大困惑和前所未有的挑战^[2]。

1 大语言模型给Python程序设计语言教学带来的挑战

1.1 学生容易产生过度依赖

大模型在编程领域呈现出爆发式增长趋势,尤其是结合了COT(Chain-of-Thought)和Self-Play RL技术之后,其在编程和数学领域的表现尤其突出。比如:由Anthropic公司推出的大语言模型Claude-3.5-Sonnet相对于上一代模型的代码能力有很大提高,被称之为代码神器。在工业界,不会编程的产品经理都能够上手构建自己的应用;在高校,学生借助大模型完成代码编写与报告撰写,已逐渐成为学习的新常态。哪怕是非常简单的编程问题,学生往往也会求助于大语言模型,由此造成的危害体现在两方面:一是不利于其编程能力的持续提升^[3],二是学生创新思维和独立思考能力相应降低。MIT的研究者于6月10日发布了一篇长达200多页的研究报告,揭示了使用大模型带来的思维能力退化风险^[4]。

1.2 学习效果评估难度加大

由于大语言模型的辅助,传统的学习效果评估方式不再适用。例如,学生通过AI大模型生成代码以完成编程作业,能够很容易通过测试用例,直接判定为满分。对于代码类的作业,即使教师增加人工修正的环节,还是无法评估学习效果。其原因在于,文本类的作业容易判断“AI”味,而代码类的作业则很难判断是否由AI生成。

1.3 大模型生成的代码质量不可靠

对于常见的简单代码,大模型辅助编程工具基本能够完成任务,但是对于一些稍复杂的任务,大模型则表现得力不从心。例如:笔者曾经让大模型实现过人工智能中的经典代码“A*搜索算法”,大模型生成的代码并不能正常运行,根据错误提示信息再反馈给大模型,“bug”仍然不能解决,且越改越错,耽误了不少时间。2025年2月份,OpenAI推出并开源了一个真实的、用于评估AI大模型编码性能的全新基准SWE-Lancer^[5]。SWE-Lancer包含了来自全球性自由职业平台Upwork的1400多个自由软件工程任务,在现实世界中总价值达到了100万美元,然而包括Claude 3.5 Sonnet、GPT-4o和o1等AI大模型没有一个能获得100万美元的全部任务价值。SWE-Lancer基准测试表明,AI大模型可以解决一些低级的编程问题,但还不能取代软件工程师。但是,对于Python程序设计语言的初学者而言,其面临的问题大多是基础性的编程问题,AI辅助编程工具可帮助学生顺利完成。然而,刚刚接触编程的学生往往意识不到这些问题,认为大模型能够取代编程,因而无需深入学习。人类学习的方式是首先理解简单任务背后的原理,然后迁移到复杂任务,如果学生没有通过独立完成简单任务加深理解、增强自信,那其将来必然缺乏将简单任务迁移到复杂任务上的能力。

2 大语言模型给Python程序设计语言教学带来的机遇

利用AI赋能各个学科的教学改革已经在教育领域全面开展^[6-8]。大语言模型虽然给Python程序设计语言教学带来了一定的冲击和挑战,但是如果能够对其加以充分利用,也会给教学工作带来一些机遇^[9-10]。文献[11]结合生成式人工智能的教育潜能与程序设计类教学挑战,提出将生成式人工智能有效融入计算机程序设计教学的创新策略和方法中以形成人机共生模型。文献[12]提出利用国内著名的大语言模型辅助教学的具体实践案例,为程序设计类课程教学改革提供新思路。

2.1 丰富教学资源

DeepSeek等大语言模型可以根据用户的提示词,快速给出教学大纲、教案、PPT、习题、教学日历等文本文档的初稿,而且在格式、内容和排版方面也能够满足相应要求^[1]。教师可以此为基础,根据自己的实际教学安排作一定调整,就能够完成教学资料的准备工作,极大地缩短传统备课时间。对于Python程序设计课程的基础代码部分,也可以让大模型生成。因此,充分利用大语言模型,教师可以将精力主要用在教学设计以及对每位学生的掌握情况上,从而节省大量重复性工作。

2.2 个性化学习支持

大语言模型可以根据学生的学习进度、知识掌握程度和兴趣爱好,为每个学生制定个性化的Python学习路径。

例如,对于初学者,模型可以推荐从基础语法和简单项目开始学习;对于有一定基础的学生,则可以提供更具挑战性的项目和进阶知识。文献[13]也验证了使用基于大语言模型的编程学习环境(Large Language Model-based Programming Learning Environments, LPLE)能够提供自适应学习支持。该研究以 215 名 Python 初学者为对象开展实验以验证 LPLE 是否真正能够为学生提供个性化支持。实验组使用 LPLE 即提供代码审查、反馈、问答功能,而对照组使用传统环境。实验结果表明,LPLE 在编程能力提升、动机影响、基本心理需要满足和学习行为分析等 4 个方面,实验组学生均优于对照组。此外,该文献还通过访谈作了定性分析,初学者反映通过自然语言交互增强了归属感,即时反馈提升了错误调试信心和多样化学习选项支持自主性。

2.3 实时答疑解惑

大语言模型可以随时随地代替教师为学生答疑,通过大语言模型解决学生在学习过程中遇到的问题,提高了学生的学习效率。此外,教师还可以根据自己的形象制作属于自己的数字人分身,还可以根据检索增强技术训练适合自己授课内容的大模型,解决特定领域的大模型知识问答系统。

2.4 大模型辅助编程作业自动评估

传统的编程作业主要有两种评估方式:人工评估和自

动评估。前者耗时耗力,但是能够评估学生的思维过程。例如:在学生的编程作业中,对部分代码步骤正确或者最后结果的格式与实际测试格式有一定出入的情况,人工评估仍可以给一定分数,从而增强学生学习积极性。后者虽然节省时间,但是只根据程序最后运行结果与测试结果的字符串是否匹配打分,忽略中间步骤。例如:在笔者授课班级的期末考试测验中,学生直接将测试用例的结果“print”出来,在平台自动评估中直接通过。大语言模型则具备自然语言处理和代码理解能力,为智能化评估提供了新可能。文献[14]探讨了如何利用大语言模型自动评估编程作业,并通过实验比较不同模型的表现。实验结果表明,Gemini 1.5 Pro 和 GPT-4 等大语言模型可有效支持编程作业自动评分,显著提升评估效率与一致性。

3 大语言模型赋能 Python 程序设计语言课程教学改革

针对目前在大语言模型背景下 Python 程序设计课程教学中存在的问题,结合 Python 程序设计课程的特点,从课程知识体系、教学模式、教学资源与工具和评估体系 4 个方面进行该课程教学改革。大语言模型背景下 Python 程序设计课程改革探索框架如图 1 所示。

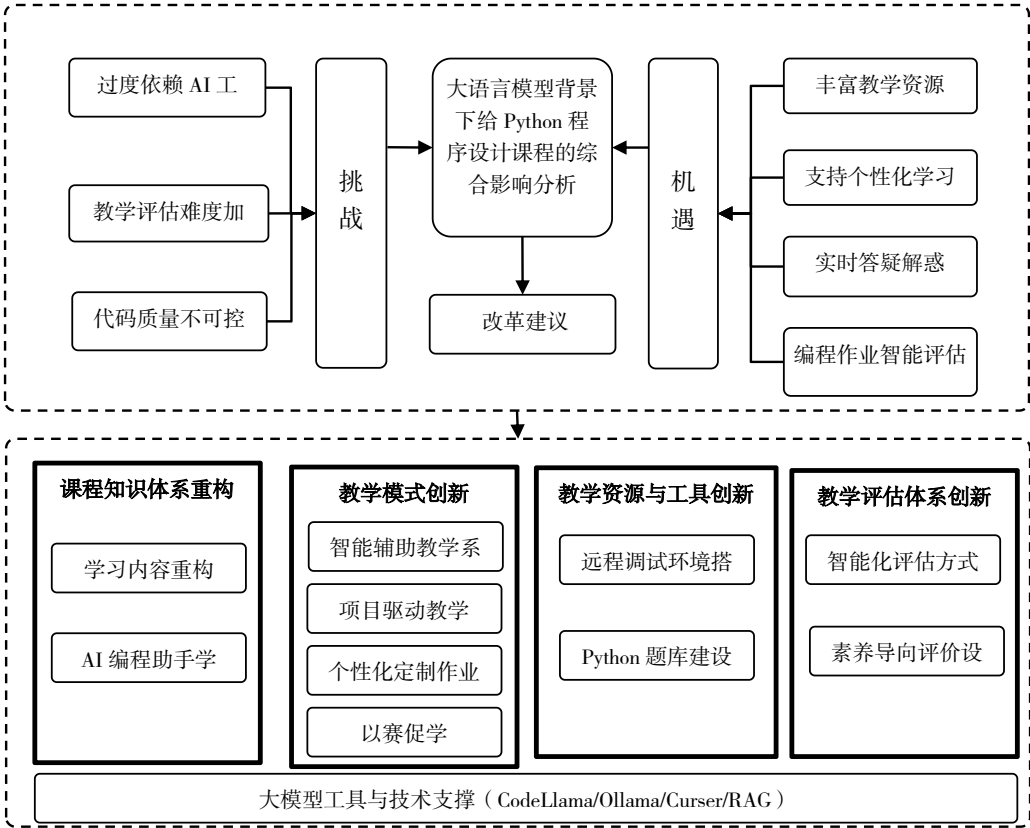


Fig. 1 The framework of teaching reform in Python programming course under the background of large language models
图 1 大语言模型背景下 Python 程序设计课程改革探索框架

3.1 课程知识体系重构

3.1.1 学习内容重构

将 Python 程序设计知识划分为基础语法、面向对象设计和数据分析等层次,明确各阶段的重点与难点。其中,重点部分讲解 Python 语言的特性、编程思维和代码规范性等,对于一些常见的知识点,如顺序语句、条件判断语句、循环语句等简单的控制语句则不再作为教学的重点内容。在 Python 数据可视化学习中,涉及到的琐碎知识点较多,如图形类型、线条类型与颜色、点的类型与颜色等,这些细节需要记忆的地方较多,但是原理性的东西几乎没有。因此,可将这部分学习的重点放在不同的图形能够表达数据的不同含义上,即学生更需要了解的是在什么情况下用什么类型的图形更能直观地表达数据背后的含义而无需对每个场景进行示例讲解。此外,也可以指导学生使用大模型辅助代码生成,教学生如何更好地使用提示词工程进行画图。

3.1.2 AI 辅助编程助手学习

在教学内容改革上,由被动接受学生使用 AI 编程助手变为主动教会学生使用该工具。并且,增加 AI 编程助手的使用、编程提示词设计等教学内容。教会学生在开发环境 Pycharm 或 VScode 中部署 Deepseek 等大模型插件以及 AI 编程助手如 Cursor 和 Coze 的使用,要求学生能够熟练掌握其主要功能,并实时跟踪 AI 编程助手的最新进展情况。

3.2 教学模式创新

3.2.1 智能辅助教学系统

受文献[1]的启发,利用 Ollama 等框架部署本地化大语言模型 Llama2、CodeLlama 等,为学生提供代码纠错、案例解析、个性化学习路径推荐功能。此外,设计针对不同任务的提示词模版,以词频统计教学为例,可以根据代码生成还是代码纠错设计不同的提示词模版,根据任务难度分为简单版本、进阶版本和高阶版本。简单版本只需要完成最简单的英文单词的词频分析,进阶版本需完成支持一段中文文本文件的输入,高阶版本则需要完成高性能的词频统计。根据学生完成情况,综合判定其是否掌握当前任务进而进入下一个任务推荐,从而真正实现“因材施教”的教育思想。

3.2.2 项目驱动教学

课程采用过程式的考核方法,分为大小作业和期末闭卷考核。大作业以完成一个数据分析项目为主,在课程刚开始时布置好,如每年最新的政府工作报告热词分析任务,该任务包含了中文分词、列表和字典使用、文件读取以及可视化分析等多个知识点。将这个任务的知识点贯穿到每个章节的学习中,让学生带着问题去学习,既激发了学生的学习兴趣,又将最新的思政元素融入到课堂中,起到了润物细无声的效果。

3.2.3 个性化定制作业

课程为学生提供个性化的作业,如根据每个学生真实的基本信息和期末考试成绩数据进行分析。这一方面有

助于提供针对性的练习和反馈,另一方面也避免了从网上找答案的抄袭行为,甚至是直接利用大模型辅助代码生成的现象。

3.2.4 以赛促学

通过鼓励学生积极参加计算机设计大赛之《大数据主题赛》赛道并将参加该项比赛作为课程过程性考核的一个方面以促进学生的 Python 程序设计科学的学习。该比赛要求学生基于给定的数据集,自选思考维度,利用 Python 程序语言解决实际的挑战性问题。该项比赛主要考察的是学生基于 Python 程序语言“提出问题、分析问题和解决问题”的能力。对于这种复杂且具有挑战性的任务,大语言模型辅助编程工具无能为力,通过将该项比赛成绩纳入课程考核之中无疑能够更进一步地锻炼学生的独立思考能力、提高学生的自学能力,激发其创新精神和创造力。

3.3 教学资源与工具优化

3.3.1 搭建远程调试环境

通过搭建 Jupyter Notebook 远程调试环境,为学生提供远程服务器,并且在服务器预先配置 Python 环境和必要的库,学生可以直接通过浏览器访问服务器。在该环境下,每个学生都可以创建自己的文件夹,远程使用 Python 的编程环境作为个人编程环境的补充,从而解决了繁琐的环境设置问题。对于教师而言,通过这种方式可以随时访问和运行学生的代码,方便代码审查和评估。

3.3.2 Python 题库建设

传统的 Python 课程作业主要来源于教材、教辅资料或网络公开的内容,对于此类题目,大模型等编程助手可以直接输出标准答案。因此,布置此类作业无法真实反映学生的掌握情况,从而需要个性化设计更具有挑战性的作业,例如在学习复合数据类型这一知识点时可布置当年四六级考试高频分析的作业。

3.4 教学评估体系创新

3.4.1 智能化评估方式构建

当前代码作业评估体系存在局限性,传统手工批改方式费时费力,而相关的作业和考试平台,如学习通或者希冀平台只根据代码是否通过测试用例打分,难以全面识别学生代码的逻辑结构、实现思路与潜在问题,可充分发挥大语言模型在代码理解优势,利用 Gemini 1.5 Pro 和 GPT-4 等大语言模型实现对学生编程作业的智能自动化自动评分^[10]。通过模型对代码进行功能判别、结构分析和规范性评价,不仅能够提高评分效率和准确性,还可为学生提供定制化的反馈意见,促进其对编程思维和风格的深层理解,助力实现“教—学—评”一体化的智能教学支持体系。

3.4.2 素养导向评价设计

为提升学生的综合实践能力和代码核心素养,课程改革中引入项目展示与复盘口头陈述相结合的评价机制。学生在完成 Python 项目开发后,以小组形式进行成果展示,阐释项目设计思路与技术实现,教师根据其阐述现场

口头询问技术细节,从而评估学生在表达、协作、问题分析与反思等方面的综合能力表现。通过形式多样的展示与互动,激发学生团队协作意识与沟通表达能力,实现从结果导向过程与素养导向转变,推动课程评价体系从知识掌握走向能力综合与素养融合,更加契合人工智能时代对复合型人才的培养需求。

4 学习效果分析

对比传统的程序设计类课程教学模式,融入大语言模型的Python 程序设计语言课程教学改革取得了良好效果。共2个平行教学班(分别为数管班、信立班)开设该门课程,课程考核包括平时成绩、实验成绩和期末考试卷面成绩,所占比例分别为20%、10%和70%。在教学过程中,信立班级采取改革后的教学模式,数管班级采取改革前传统的教学模式,各班级总评成绩如图2所示。可以看出,信立班优秀率和良好率都超过了数管班,并且不及格率低于数管班。信立班教学效果评教得分为99.81,且获得了学生的积极反馈,他们表示,在大语言模型的支持下,其学习动力更足,学习自信心也更强。

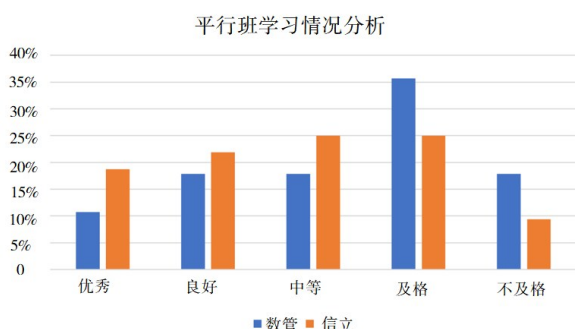


Fig. 2 Comparison of teaching effectiveness

图2 教学效果比较

5 结语

大语言模型的出现提升了Python 程序设计教学效果,但同时也带来了一系列挑战。本文在全面分析生成式人工智能对Python 程序设计课程所带来的挑战和机遇的基础上,结合Python 程序设计课程的特点,提出大语言模型下的Python 程序设计课程改革探索框架,并开展实践探索。结果表明,改革后的教学效果明显提升,同时获得了学生的积极反馈。未来,Python 教育需在基础知识巩固与AI 协作能力培养之间找到平衡,避免学生成为“只会调AI 的代码搬运工”,而是真正具备计算思维与创新能力的开发者。

参考文献:

[1] YIN L Z, XU J J, LI S S, et al. Exploration of teaching computer program-

ming courses in the era of artificial intelligence[J]. Computer Education, 2025, 23(2): 123-127.

尹良泽,徐建军,李姗姗,等. 人工智能时代下的计算机程序设计课程教学探索[J]. 计算机教育, 2025, 23(2): 123-127.

[2] XU H, JU X L, WANG H C. Challenges and countermeasures of programming teaching under large models[J]. Computer Education, 2023, 21(11): 60-64.

徐慧,鞠小林,王皓晨. 大模型下编程教学面临的挑战与应对[J]. 计算机教育, 2023, 21(11): 60-64.

[3] LIU L, WU Y F, CHEN W W, et al. Challenges and opportunities of large language models for university programming courses[J]. Computer Education, 2025, 23(2): 118-122.

刘凌,吴永芬,陈卫卫,等. 大语言模型对高校程序设计类课程的挑战与机遇[J]. 计算机教育, 2025, 23(2): 118-122.

[4] KOSMYNA N, HAUPTMANN E, YUAN Y T, et al. Your brain on ChatGPT: accumulation of cognitive debt when using an AI assistant for essay writing task[DB/OL]. <http://arxiv.org/abs/2506.08872>.

[5] MISERENDINO S, WANG M, PATWARDHAN T, et al. SWE-Lancer: can frontier LLMs earn \$1 million from real-world freelance software engineering?[DB/OL]. <https://arxiv.org/abs/2502.12115>.

[6] QIU F, WU Y D. Core elements analysis of educational innovation driven by generative artificial intelligence[J]. Education Development Research, 2024, 44(Z1): 9-16.

邱峰,吴跃东. 生成式人工智能驱动教育创新的核心要素分析[J]. 教育发展研究, 2024, 44(Z1): 9-16.

[7] GAO H H, CHEN Z J. Artificial intelligence empowering the teaching reform of programming courses[J]. Computer Education, 2024, 22(7): 41-43, 48.

高洪皓,陈章进. 人工智能赋能程序设计课程教学改革[J]. 计算机教育, 2024, 22(7): 41-43, 48.

[8] LIU B Q, YIN H H. Empowering teachers' digital literacy through artificial intelligence: strategies, scenarios and evaluation-feedback mechanisms[J]. Modern Educational Technology, 2024, 34(7): 23-31.

刘邦奇,尹欢欢. 人工智能赋能教师数字素养提升:策略、场景与评价反馈机制[J]. 现代教育技术, 2024, 34(7): 23-31.

[9] JACOBS S, JASCHKE S. Evaluating the application of large language models to generate feedback in programming education[C]//Kos: 2024 IEEE Global Engineering Education Conference (EDUCON), 2024.

[10] JU X L, ZHANG Y M, WANG H C, et al. Exploration of Python programming course teaching assisted by large language models[J]. Computer Education, 2024, 22(9): 33-37.

鞠小林,张艳梅,王皓晨,等. 基于大语言模型辅助教学的Python 编程课程教学探索[J]. 计算机教育, 2024, 22(9): 33-37.

[11] ZHANG H Z, ZHOU X B, XU Y H, et al. Generative artificial intelligence empowering innovative teaching in computer programming courses[J]. Computer Education, 2024, 22(7): 44-48.

张红卓,周小宝,许玉煊,等. 生成式人工智能赋能计算机程序设计类课程教学创新[J]. 计算机教育, 2024, 22(7): 44-48.

[12] XIE H B, LIU F, QIN H X. Exploring the application of large language models in the reform of foundational programming education[J]. Computer Knowledge and Technology, 2024, 20(8): 39-42.

谢红标,刘芳,覃浩轩. 大语言模型在程序设计基础教学改革中的应用探索[J]. 电脑知识与技术, 2024, 20(8): 39-42.

[13] CHOI S, KIM H. The impact of a large language model-based programming learning environment on students' motivation and programming ability[J]. Education and Information Technologies, 2024, 30: 8109-8138.

[14] MOHAMED K, YOUSEF M, WALAA M W, et al. Hands-on analysis of using large language models for the auto evaluation of programming assignments[J]. Information Systems, 2025, 128: 102473.