

生成式人工智能赋能师范院校程序设计课程教学模式构建研究

崔文升, 李宏博*, 李欣, 刘彰基

大庆师范学院计算机科学与信息技术学院, 黑龙江 大庆

收稿日期: 2026年6月15日; 录用日期: 2026年8月7日; 发布日期: 2026年8月17日

摘要

生成式人工智能正在影响程序设计学习和软件开发实践,也为师范院校程序设计课程改革带来新的可能。当前课程中,学生编程基础差异较大,教学案例与教育应用场景联系不够紧密,实践指导和过程评价仍显不足。围绕师范生数字素养与程序设计实践能力培养,本文构建了由目标重构、任务驱动、人机协同、过程评价和反思改进组成的课程教学模式。该模式将AI工具嵌入课前导学、课堂实践、课后拓展和课程评价过程,通过教育场景化任务、代码分析支持和学习过程证据,引导学生从语法模仿转向问题分析、代码审查、调试优化和教育应用迁移。以2023级软件工程专业1、2班90名学生为对象的课程试用表明,该模式能够提高学习反馈的及时性,促进学生关注代码理解、AI使用规范和程序设计的教育应用价值,可为师范院校程序设计课程智能化改革提供参考。

关键词

生成式人工智能, 师范院校, 程序设计课程, 教学模式, 人机协同

Research on the Teaching Model of Programming Courses in Normal Universities Empowered by Generative Artificial Intelligence

Wensheng Cui, Hongbo Li*, Xin Li, Zhangji Liu

School of Computer Science and Information Technology, Daqing Normal University, Daqing Heilongjiang

*通讯作者。

文章引用: 崔文升, 李宏博, 李欣, 刘彰基. 生成式人工智能赋能师范院校程序设计课程教学模式构建研究[J]. 创新教育研究, 2026, 14(8): 180-187. DOI: 10.12677/ces.2026.148593

Received: June 15, 2026; accepted: August 7, 2026; published: August 17, 2026

Abstract

Generative artificial intelligence is influencing programming learning and software development practice, bringing new possibilities for the reform of programming courses in normal universities. At present, such courses still face problems such as considerable differences in students' programming foundations, insufficient connection between teaching cases and educational application scenarios, inadequate practical guidance, and weak process evaluation. Focusing on the cultivation of normal university students' digital literacy and programming practice ability, this paper constructs a teaching model consisting of goal reconstruction, task-driven learning, human-AI collaboration, process evaluation, and reflective improvement. The model embeds AI tools into pre-class guidance, classroom practice, after-class extension, and course evaluation. Through educational scenario-based tasks, code analysis support, and learning process evidence, it guides students to shift from grammar imitation to problem analysis, code review, debugging and optimization, and educational application transfer. A course trial involving 90 students from Classes 1 and 2 of the 2023 Software Engineering major shows that the model can improve the timeliness of learning feedback and promote students' attention to code understanding, AI usage norms, and the educational application value of programming. This study provides a reference for the intelligent reform of programming courses in normal universities.

Keywords

Generative Artificial Intelligence, Normal Universities, Programming Courses, Teaching Model, Human-AI Collaboration

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

生成式人工智能(Generative Artificial Intelligence, GAI)的快速发展正在改变程序设计学习方式和软件开发实践,也推动高校程序设计课程不断向智能化、个性化方向发展。ChatGPT、DeepSeek、GitHub Copilot、Cursor 等智能工具已广泛应用于代码生成、程序调试、代码解释和测试样例生成等环节,为学生自主学习和教师教学提供了新的支持方式[1][2]。与此同时,生成式人工智能的广泛应用也带来了学习依赖、过程评价和学术诚信等新的挑战,如何科学发挥人工智能的辅助作用,促进学生程序设计能力持续发展,已成为当前程序设计教育研究的重要内容[3]。

程序设计课程是计算机类专业的核心基础课程,也是培养学生计算思维、算法设计能力和问题解决能力的重要载体。Wing 提出的计算思维理论强调利用计算机科学的基本思想分析和解决实际问题,为程序设计课程改革提供了重要理论基础[4]。对于师范院校而言,程序设计课程不仅承担专业知识教学任务,还需要服务于师范生数字素养、教育技术应用能力和创新实践能力培养。未来教师不仅需要具备基本程序设计能力,还需要能够利用人工智能工具开展教学资源建设、学习数据处理和教育应用开发。因此,程序设计课程教学应更加突出教育应用场景和数字化教学能力培养。

近年来,生成式人工智能赋能教育教学已成为国内外研究热点。UNESCO 发布的《Guidance for Generative AI in Education and Research》指出,生成式人工智能能够为个性化学习、教学资源生成和学习支

持提供新的技术路径,但同时应关注学习主体性、伦理规范和教育公平等问题[5]。已有研究表明,大语言模型能够提高学习反馈效率,在程序设计课程中可用于代码辅助生成、错误分析、编程指导和学习评价等环节,有助于改善学生学习体验和课堂教学效果[6][7]。与此同时,也有研究指出,如果缺乏教师有效引导,学生容易形成对 AI 工具的依赖,影响算法设计、程序调试和自主思考能力的发展[6]。

总体来看,现有研究主要关注生成式人工智能在程序设计课程中的工具应用和教学辅助,对于师范院校程序设计课程如何兼顾教育应用场景、人机协同学习和过程评价改革,仍缺少较为系统的研究;针对师范生培养特点构建程序设计课程教学模式的研究也相对有限。此外,人机协同认知理论认为,学习应由教师、学习者与智能工具共同完成知识建构, AI 应作为学习支架而非学习主体,这为生成式人工智能融入程序设计课程提供了新的理论视角[8]。

基于此,本文立足师范院校程序设计课程教学实际,围绕师范生数字素养和程序设计实践能力培养需求,分析生成式人工智能赋能程序设计课程教学改革的现实依据,构建“目标重构、任务驱动、人机协同、过程评价、反思改进”的教学模式,并结合《面向对象程序设计》课程试用情况,对教学实施路径和应用效果进行分析,以期对师范院校程序设计课程智能化教学改革提供参考。

2. 生成式人工智能赋能师范院校程序设计课程的现实依据

生成式人工智能进入程序设计课程,不只是增加一种辅助工具,还会影响课程目标、学习过程和评价方式。师范院校程序设计课程既要帮助学生掌握编程基础,也要服务于师范生数字素养和教育技术应用能力培养。因此,课程改革需要从学生基础、任务场景和评价规范三个方面寻找切入点。

2.1. 学生基础差异要求课程提供个性化学习支持

程序设计课程对逻辑思维、抽象能力和实践操作要求较高,而师范院校学生的学习经历和编程基础差异较为明显。有的学生入学前接触过编程语言或信息技术课程,能够较快理解变量、循环、函数、类与对象等内容;也有学生缺乏编程经验,对程序运行机制、代码结构和调试过程理解较慢,容易在学习初期产生畏难情绪。传统课堂通常采用统一进度和统一实验任务,较难兼顾不同层次学生的需要。生成式人工智能可以为学生提供概念解释、错误提示、代码示例和思路引导,也可以辅助教师设计分层任务、整理常见问题和生成补充练习。合理引入 AI 工具,有助于缓解学生基础差异带来的教学压力,使课程支持方式更加灵活。

2.2. 师范生培养需求要求程序设计任务贴近教育场景

师范院校程序设计课程不同于单纯面向软件开发岗位的训练。师范生学习程序设计,除了掌握语言语法和程序结构,还需要理解数字技术如何服务教学资源建设、课堂互动、学习评价和教育数据处理。当前部分课程仍以孤立知识点和简单例题为主,学生容易把程序设计理解为机械编码训练,缺少面向教育问题的应用意识。生成式人工智能为任务设计提供了更多支持。教师可以围绕学生成绩统计、课堂互动小程序、学习数据可视化、教学资源管理等场景设计项目任务,并利用 AI 辅助生成任务说明、案例数据、代码框架和测试样例。学生在完成任务时,可以借助 AI 理解需求、分析代码和定位错误,但仍需自己完成问题分析、功能设计、代码修改和结果验证。通过这种方式,程序设计课程能够更好地连接编程知识与教育应用。

2.3. AI 参与学习要求重构过程评价与使用规范

生成式人工智能为程序设计学习提供了便利,也带来了新的管理问题。学生可以快速获得代码和解释,如果缺少教师引导,容易出现直接复制生成代码、忽视程序逻辑、跳过调试过程等现象。对于程序

设计课程而言，最终代码能否运行，并不能完全说明学生是否真正掌握了问题分析、算法设计和代码实现能力。因此，课程评价需要从结果评价转向过程评价。教师不仅要看程序运行结果，还应关注学生的需求分析、代码设计、调试记录、AI 使用说明和学习反思。学生在提交实验作业时，应说明 AI 使用场景、提示内容、采纳情况和个人修改过程。教师可通过课堂提问、代码讲解、过程记录和实验报告，综合判断学生是否真正理解代码逻辑和解决方案。

3. 生成式人工智能赋能程序设计课程教学模式构建

师范院校程序设计课程既要培养编程基础和计算思维，也要服务师范生数字素养、教育技术应用能力和教育场景问题解决能力的形成。生成式人工智能融入课程后，教学模式需要围绕课程目标、学习任务、主体关系和评价方式进行调整，形成“目标重构、任务驱动、人机协同、过程评价、反思改进”的基本框架，如表 1 所示。

3.1. 课程目标的师范取向

生成式人工智能背景下，程序设计课程目标应从“掌握语法、完成代码”转向“理解思想、解决问题、规范协同和迁移应用”。知识目标强调学生掌握程序设计的基本概念、程序结构和运行机制；能力目标强调学生能够进行问题分析、程序设计、代码实现、调试修改和结果验证；素养目标强调学生理解程序设计与教育数字化之间的关系，形成规范使用智能工具的意识。学生可以使用 DeepSeek、ChatGPT 等大语言模型辅助理解概念、解释代码和分析错误，也可以借助 Cursor、GitHub Copilot、通义灵码等工具获得代码提示，但必须判断生成内容是否合理，并对代码进行修改和验证。

3.2. 生成式人工智能赋能程序设计课程教学模式的构建

Table 1. Framework table of the teaching model for programming courses empowered by Generative Artificial Intelligence
表 1. 生成式人工智能赋能程序设计课程教学模式框架表

模式要素	目标指向	教师任务	学生活动	AI 支持方式	评价证据
目标重构	编程能力、AI 协同能力和教育应用能力	明确知识、能力和素养目标	理解学习要求，明确应用方向	生成目标说明和知识清单	学习任务单、目标达成说明
任务驱动	促进知识整合与问题解决	设计分层任务和教育场景案例	完成需求分析、代码实现和结果验证	生成任务情境、案例数据和测试样例	项目代码、运行结果、任务报告
人机协同	形成教师主导、学生主体、AI 辅助关系	指导 AI 使用，组织代码讲解	借助 AI 理解概念、分析错误和优化代码	提供代码解释、错误诊断和优化建议	AI 使用记录、代码修改说明
过程评价	关注真实学习过程和能力形成	建立评价标准，开展代码走查	提交调试记录、测试样例和学习反思	辅助整理问题和反馈建议	调试记录、测试报告、反思报告
反思改进	促进代码理解、审查和应用	根据反馈优化任务与评价	总结问题，完善功能和代码结构	生成复盘问题和拓展思路	版本记录、项目完善说明

师范院校程序设计课程既要培养编程基础和计算思维，也要服务师范生数字素养、教育技术应用能力和教育场景问题解决能力的形成。生成式人工智能融入课程后，教学模式需要围绕课程目标、学习任务、主体关系和评价方式进行调整，形成“目标重构、任务驱动、人机协同、过程评价、反思改进”的基本框架，如表 1 所示。

任务是连接知识学习与能力形成的重要载体。师范院校程序设计课程应保留必要的基础语法训练，同时增加教育场景化任务，使学生在真实或准真实问题中理解程序设计知识。课程任务可按照“基础训

练、综合应用、教育拓展”组织。基础训练用于巩固变量、选择结构、循环结构、函数、数组、类与对象等内容；综合应用用于培养模块划分、代码调试和功能整合能力；教育拓展则引导学生将程序设计知识用于教学管理、学习评价和资源建设。以《面向对象程序设计》中的“学生成绩管理系统”为例，教师可引导学生围绕学生信息、课程信息和成绩管理等对象设计类结构，实现信息录入、成绩统计、等级划分、数据查询和结果输出等功能。该任务既涉及类与对象、封装、文件操作和异常处理等知识点，也贴近师范生熟悉的教学管理情境。AI 工具可辅助需求梳理、代码提示和错误诊断，但任务完成的关键仍是学生对对象关系、功能逻辑和应用价值的理解。

3.3. 人机协同的角色关系

生成式人工智能赋能程序设计课程，关键在于明确教师、学生和 AI 工具的角色。教师是教学目标的设计者和学习过程的引导者，负责选择任务情境、把握教学重点、规范工具使用和判断学习质量。学生是学习主体，需要完成需求分析、方案设计、代码修改和结果说明。AI 工具是学习支持者，主要承担概念解释、思路提示、错误诊断和代码优化建议等辅助功能。在这一模式中，人机协同不是让 AI 替代学生完成程序，而是借助 AI 降低学习初期的理解难度，提高反馈效率，帮助学生发现代码中的问题。教师应引导学生从“直接获得答案”转向“借助 AI 改进思路”，从“复制代码”转向“审查代码”。只有保持学生主体地位，生成式人工智能才能成为促进程序设计能力发展的学习支架。

4. 生成式人工智能赋能程序设计课程的实施路径

生成式人工智能赋能程序设计课程，需要落实到课前、课中、课后和课程反馈等环节。课前重在了解学情，课中重在完成任务，课后重在迁移应用，反馈重在检验教学模式的可行性。

4.1. 课前导学与学情诊断

课前环节主要用于知识预习和学情诊断。教师可借助 DeepSeek、ChatGPT 等工具生成导学问题、代码阅读材料和预习任务，引导学生提前了解基本概念和任务背景。以“类与对象”教学为例，教师可围绕“如何用程序表示学生信息”“如何统计班级成绩”等问题设计预习内容。学生可使用 AI 工具辅助理解概念、阅读示例代码并整理疑问，教师根据预习反馈判断学生知识基础和理解难点。

4.2. 课中实践与任务推进

课中教学应以任务实践为主线。教师可采用情境引入、案例讲解、任务实践、代码调试和交流总结的流程组织课堂。以《面向对象程序设计》中的“学生成绩管理系统”为例，学生围绕学生信息、课程信息和成绩管理等对象设计类结构，实现信息录入、成绩统计、等级划分和结果输出等功能。在需求分析阶段，学生可利用 DeepSeek 或 ChatGPT 梳理功能模块；在代码实现阶段，可借助 Cursor、GitHub Copilot 或通义灵码获得代码提示；在调试阶段，可利用 AI 分析报错信息并生成测试样例。教师重点引导学生说明类设计依据，比较不同方案，判断 AI 生成代码是否存在逻辑漏洞、结构冗余或规范性问题。

4.3. 课后拓展与应用迁移

课后任务应体现分层要求和应用迁移。基础任务侧重巩固语法和基本功能实现，提升任务可加入文件读写、异常处理、数据排序等内容，拓展任务则面向教育场景，引导学生将程序设计知识应用于班级管理、学习数据处理和教学资源建设。例如，在学生成绩管理任务基础上，可要求学生生成班级成绩分析报告、统计不同分数段人数，或输出学习建议。学生可以借助 AI 工具优化代码和检查测试样例，但需要说明 AI 使用场景、采纳情况和个人修改过程。

4.4. 课程试用与效果反馈

为检验教学模式的可行性,本文以大庆师范学院 2023 级软件工程专业 1、2 班共 90 名学生为对象,在《面向对象程序设计》课程相关内容中开展初步试用。教学中围绕类与对象、封装、继承、文件操作和异常处理等知识点,设计“学生成绩管理系统”等任务,引导学生在需求分析、类结构设计、代码实现、程序调试和结果说明等环节使用 DeepSeek、ChatGPT、Cursor、GitHub Copilot 和通义灵码等工具。相关数据主要来源于课程作业提交记录、实验报告和课堂观察记录。从作业和过程材料看,90 名学生中有 86 人按时提交项目代码和运行结果,占 95.6%;82 人提交 AI 使用说明,占 91.1%;76 人提交较完整的调试记录,占 84.4%;68 人能够说明 AI 提示内容、采纳情况和个人修改过程,占 75.6%;54 人能够结合成绩统计、学习数据分析等场景说明程序功能的教育应用价值,占 60.0%。这些结果说明,多数学生能够按要求记录 AI 使用过程,并开始关注代码理解、修改依据和教育应用价值。课堂观察显示,基础薄弱学生在遇到语法错误、运行异常和代码理解困难时,能够借助 AI 工具获得解释和调试提示,减少简单问题上的停滞时间;基础较好的学生则更多利用 AI 工具进行功能扩展、代码优化和测试样例设计。与此同时,少数学生仍存在依赖 AI 生成完整代码、代码逻辑解释不充分、调试记录简单等问题。总体来看,该模式能够提升学习支持的及时性,增强程序设计任务与教育应用场景的联系,但后续仍需结合更多课程数据持续分析。

总体来看,该模式能够提升学习支持的及时性,增强程序设计任务与教育应用场景之间的联系。从学生实验报告和学习反思内容来看,多数学生能够说明 AI 在需求分析、代码调试和错误定位中的具体作用,部分学生能够主动比较 AI 生成代码与自主编写代码之间的差异,并提出改进意见。对学生提交的程序进行抽样检查发现,大部分程序能够按照课程要求完成模块划分和功能实现,代码注释和命名规范较以往有所改善,但在算法优化和程序结构设计方面仍存在一定提升空间。这表明,生成式人工智能能够提高学习支持效率,但学生的问题分析能力和自主设计能力仍需要教师持续引导。

5. 教学模式实施保障与优化建议

生成式人工智能赋能程序设计课程,不仅需要教学模式创新,还需要课程资源、教师能力、学生规范和评价机制等方面协同支撑。对于师范院校而言,程序设计课程既要满足专业知识教学要求,又要服务于师范生数字素养和教育技术应用能力培养。因此,应不断完善教学实施保障机制,提高教学模式的可持续应用能力。

5.1. 完善课程资源与教师支持体系

课程资源是教学模式顺利实施的重要基础。教师应围绕程序设计课程核心知识,建设覆盖课前导学、课堂实践、课后拓展和课程评价的数字化教学资源,包括典型案例、项目任务、测试样例、常见错误库和学习反思模板等。资源建设应充分体现师范院校特色,适当增加教学管理、学习评价、课堂互动、学习数据处理等教育应用案例,引导学生理解程序设计在教育数字化中的实际价值。

与此同时,教师应不断提升人工智能工具应用能力,熟悉 DeepSeek、ChatGPT、GitHub Copilot、Cursor、通义灵码等工具的基本功能及适用场景,能够根据课程目标合理选择 AI 工具,并对 AI 生成内容进行甄别、修改和优化,确保教学内容的科学性和准确性。

5.2. 建立规范的人机协同学习机制

生成式人工智能能够提供即时反馈和编程支持,但不能替代学生自主思考和程序设计过程。课程实施过程中,应明确教师、学生与 AI 工具之间的职责边界,坚持教师主导、学生主体、AI 辅助的人机

协同原则。

课程开始前,应向学生明确 AI 工具使用规范,允许学生利用 AI 完成概念理解、代码调试、错误分析和测试样例设计等学习活动,但不得直接提交未经分析和修改的 AI 生成代码。实验报告和课程项目应增加“AI 使用说明”栏目,要求学生记录 AI 工具名称、使用目的、提示内容、采纳情况以及个人修改过程,使 AI 辅助学习过程具有可追溯性。同时,可通过课堂提问、代码讲解、项目展示和课程答辩等方式,进一步检验学生对程序设计思想和代码逻辑的真实理解。

5.3. 完善课程评价与质量改进机制

生成式人工智能背景下,程序设计课程评价应更加关注学生学习过程和能力形成,不宜仅依据程序运行结果进行评价。课程评价可综合项目完成情况、代码规范、程序设计思路、调试过程、AI 使用规范、实验报告和学习反思等多项指标,构建结果评价与过程评价相结合的评价体系。

在评价实施过程中,可利用课堂观察、实验报告、AI 使用说明、代码版本记录等多源数据,对学生学习过程进行综合分析。对于教育场景项目,还应关注学生能否结合教学实际说明程序设计的应用价值,考查其利用程序设计解决教育问题的能力。课程结束后,教师可依据课程目标达成情况、学生学习反馈和项目完成质量持续优化教学内容、实践任务和评价指标,形成“实施-反馈-改进”的持续优化机制。

5.4. 关注 AI 应用风险与持续推广

生成式人工智能在提高学习效率的同时,也可能带来代码依赖、学习过程弱化、成果归属不清以及学术诚信等问题。如果学生过度依赖 AI 生成代码,容易忽视需求分析、算法设计和程序调试等关键学习环节,影响程序设计能力的发展。因此,教师应加强 AI 应用伦理教育,引导学生正确认识 AI 工具的辅助属性,强化自主分析、独立验证和规范引用意识。

本研究提出的教学模式目前已在《面向对象程序设计》课程中进行了初步试用,能够为师范院校程序设计课程改革提供参考,但仍存在一定局限。课程试用对象主要为一个年级两个教学班,研究周期较短,尚未开展平行班对照研究。后续可采用准实验研究设计,引入课程成绩、代码质量、学习投入、学习反思和课堂观察等多维数据,对教学模式进行持续验证。同时,可进一步完善教师实施指南、AI 工具推荐方案和课程评价量规,形成可复制、可推广的程序设计课程教学改革实践模式。

6. 结语

生成式人工智能的发展为程序设计课程教学改革提供了新的技术支撑,也推动教学目标、教学方式和课程评价不断调整。本文立足师范院校程序设计课程教学实际,围绕师范生数字素养和程序设计实践能力培养需求,分析了生成式人工智能赋能课程改革的现实依据,构建了“目标重构、任务驱动、人机协同、过程评价、反思改进”的教学模式,并提出了覆盖课前导学、课堂实践、课后拓展和课程评价的实施路径。课程试用结果表明,该模式能够提升学习支持的及时性,促进学生规范使用 AI 工具,增强程序设计学习与教育应用场景之间的联系,对提升学生代码理解、问题分析和实践能力具有积极作用。

需要指出的是,本研究属于教学模式构建与初步实践探索,课程试用对象仅为 2023 级软件工程专业两个教学班,研究周期较短,研究结论仍有进一步验证的空间。未来可采用平行班对照或准实验研究设计,引入课程成绩、代码质量、学习投入、学习反思和课堂观察等多维数据,对教学模式的实施效果进行持续跟踪和量化分析。同时,应进一步完善 AI 使用规范、课程评价量规和教师实施指南,积极探索生成式人工智能在程序设计课程中的科学应用,推动师范院校程序设计课程教学改革向更加智能、规范和高质量的方向发展。

基金项目

黑龙江省教育学会 2026 年度教育科学研究“十五五”规划一般课题(项目编号 LQY20260078 青年专项);

黑龙江省教育科学规划重点课题(项目编号 GJB1423022);

全国高等院校计算机基础教育研究会计算机基础教育教学研究项目(项目编号 2025-AFCEC-302)。

参考文献

- [1] Miao, F. and Holmes, W. (2023) Guidance for Generative AI in Education and Research. UNESCO.
- [2] 中共中央 国务院印发《教育强国建设规划纲要(2024—2035 年)》[EB/OL].
https://www.gov.cn/gongbao/2025/issue_11846/202502/content_7002799.html, 2025-01-19.
- [3] Wing, J.M. (2006) Computational Thinking. *Communications of the ACM*, **49**, 33-35.
<https://doi.org/10.1145/1118178.1118215>
- [4] Zawacki-Richter, O., Marín, V.I., Bond, M. and Gouverneur, F. (2019) Systematic Review of Research on Artificial Intelligence Applications in Higher Education—Where Are the Educators? *International Journal of Educational Technology in Higher Education*, **16**, Article No. 39. <https://doi.org/10.1186/s41239-019-0171-0>
- [5] Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., *et al.* (2023) ChatGPT for Good? On Opportunities and Challenges of Large Language Models for Education. *Learning and Individual Differences*, **103**, Article 102274. <https://doi.org/10.1016/j.lindif.2023.102274>
- [6] Becker, B.A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J. and Santos, E.A. (2023) Programming Is Hard—Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, New York, 15-18 March 2023, 500-506.
- [7] Denny, P., Prather, J., Becker, B.A., Finnie-Ansley, J., Hellas, A., Leinonen, J., *et al.* (2024) Computing Education in the Era of Generative AI. *Communications of the ACM*, **67**, 56-67. <https://doi.org/10.1145/3624720>
- [8] 姜毅, 陈建海, 巫英才, 邓水光, 纪守领. 大语言模型驱动的程序设计教学变革[J]. 计算机教育, 2025(8): 177-182.