

非计算机专业 Python 课程 AI 协同教学改革*

胡金萍 黄利广 蔡天星**

深圳技术大学人工智能学院, 深圳 518118

摘要 针对高校非计算机专业 Python 课程的内容偏离智能化编程生态、教学过程缺失人机协同、评价方式难以识别学生 AI 协同编程能力、课程学习成果向专业问题迁移不足等问题, 本文构建 AI 协同教学改革方案。方案围绕非计算机专业学生的专业应用需求, 构建“基础筑基—AI 辅助—专业应用—项目落地”四层课程内容体系, 形成与之衔接的一体化教学流程, 并提出“限制—引导—开放—审视”的阶段性 AI 介入机制和多元评价体系。教学实践中, 将大语言模型和智能编程工具嵌入代码理解与实践项目, 并结合学生专业背景分专业构建案例库。课程平台数据显示, 改革后学生课堂出勤率较以往有所提高, 实验作业完成率由 65% 提升至 73%, 综合项目完成率由 76% 提升至 89%。实践表明, 该模式有助于降低编程学习门槛, 提升学生 AI 协同编程能力和专业问题解决能力, 可为非计算机专业程序设计课程改革提供参考。

关键字 非计算机专业, Python 程序设计, AI 协同教学, 人机协同编程, 课程改革

AI-Collaborative Teaching Reform in Python Courses for Non-Computer Majors*

Jinping Hu Liguang Huang Tianxing Cai**

School of Artificial Intelligence Shenzhen Technology University,
Guangdong 518118, China;
caitianxing@sztu.edu.cn

Abstract—To address the problems in Python courses for non-computer majors, including a mismatch between course content and the intelligent programming ecosystem, the absence of human-AI collaboration in the teaching process, the difficulty of evaluating students' AI collaborative programming ability, and the insufficient transfer of learning outcomes to discipline-specific problems, this paper proposes an AI collaborative teaching reform model. Oriented toward the professional application needs of non-computer majors, the model constructs a four-level course content system of “foundation building—AI assistance—professional application—project implementation,” develops an integrated teaching process aligned with this content system, and proposes a staged AI intervention mechanism of “restriction—guidance—openness—reflection” together with a diversified evaluation system. In teaching practice, large language models and intelligent programming tools were embedded into code understanding and practical projects, while discipline-specific case libraries were developed according to students' professional backgrounds. Data from the course platform show that, after the reform, students' classroom attendance improved compared with previous semesters, the completion rate of experimental assignments increased from 65% to 73%, and the completion rate of comprehensive projects increased from 76% to 89%. The results indicate that this model helps lower the entry barrier to programming, enhances students' AI collaborative programming ability and professional problem-solving ability, and provides a reference for the reform of programming courses for non-computer majors.

Keywords—non-computer majors, Python programming, AI-assisted collaborative teaching, human-AI collaborative programming, curriculum reform

1 引言

人工智能 (Artificial Intelligence, AI) 的快速发展正重塑社会生产方式、行业发展路径和高等教育人才培养模式。大学生掌握人工智能基础知识和应用能力, 已由能力提升选项转变为适应智能社会和未来职业发展的基本素养, 也是高校落实“新工科”“新

文科”建设要求、培养复合型应用人才的关键任务^[1]。Python 作为 AI 技术生态中最具通用性和实践价值的编程语言之一, 是非计算机专业学生进入人工智能领域的重要入口。然而, 当前高校非计算机专业 Python 课程建设并未完成 AI 协同: 一是课程内容缺乏与专业应用场景融入, 偏离智能化编程生态; 二是教学过程以教师讲授为主, 缺少人机协同学习过程; 三是评价方式偏重结果正确性, 难以有效识别学生的 AI 协同编程能力; 四是课程学习成果向专业问题迁移不足, 学生难以将编程知识转化为解决专业问题的能力^[2-7]。

*基金资助: 本文得到深圳技术大学教改项目 (20261013) 资助

*通讯作者: 蔡天星 caitianxing@sztu.edu.cn

AI 时代 Python 课程的教学目标不能仅简单允许学生使用 AI 工具或借助 AI 生成代码, 而应面向真实专业场景, 围绕课程内容体系、教学流程、AI 介入边界和评价方式进行系统设计, 培养学生将专业问题转化为计算任务、利用程序处理数据、借助 AI 工具完成任务的综合能力^[8-10]。

因此, 本文以非计算机专业 Python 课程为研究对象, 围绕 AI 协同教学开展课程改革探索。主要贡献如下: 一、面向非计算机专业学生提出 AI 协同编程能力结构, 将问题转化、需求表达、代码理解与验证、修改与迁移纳入课程能力目标; 二、构建四层递进式内容体系, 实现编程知识、AI 工具使用与专业场景应用的融合; 三、提出阶段性 AI 介入机制和多元评价体系, 使 AI 使用过程可引导、可追踪、可评价。

2 非计算机专业 Python 课程教学现状与问题分析

高校面向非计算机专业开设的 Python 课程已成为培养学生计算思维和 AI 应用能力的重要载体。但从实际教学情况看, 课程在内容体系、教学过程、评价方式和学习迁移等方面仍存在一定不足, 尚未充分适应 AI 时代复合型应用人才培养的需求^[11-13]。

2.1 课程内容缺乏与专业应用场景融入, 偏离智能化编程生态

现有课程通常以基础语法和程序逻辑讲解为主此类内容有助于学生建立基本编程基础, 但课程整体缺失专业场景化案例, 且偏离智能化编程生态, 如表 1 所示。

表 1 传统课程内容

章节	课程内容	课程实验
一	Python 简介	环境安装与基础运行体验
二	基础语法与简单数据类型	变量、数字与字符串基础练习
三	列表与元组	列表与元组的创建、访问和基本操作
四	控制结构	条件判断与循环结构练习
五	字典与集合	字典、集合及简单数据管理练习
六	函数	函数定义、参数与函数调用练习
七	面向对象的程序设计	类与对象的基础设计练习
八	综合复习	基础语法综合练习

对于非计算机专业学生而言, Python 学习的价值不仅在于掌握编程规则, 更在于能够利用编程技术处理本专业中智能应用问题。但实际教学中, 课程案例多为通用性练习, 与学生专业背景结合不紧密, 学生难以理解编程知识与专业学习之间的联系, 学习动力不足。

同时, 课程内容与智能化编程生态之间的衔接不足。大语言模型和智能编程工具已经能够在程序设计和开发的全流程为学习者提供支持。但现有课程未将 AI 工具的使用方法、使用边界和评价标准系统纳入教学内容。学生容易出现不会有效使用 AI 或过度依赖 AI 生成代码的问题, 难以形成真正的人机协同编程能力。

2.2 教学过程以教师讲授为主, 缺少人机协同学习过程

现有课程在教学实施过程中, 仍较多沿用语法教学、上机练习、期末考核的传统教学路径。这种教学方式虽有助于保证基础内容讲授的系统性, 但真实问题情境不足, 难激发学生的探究欲望且学生参与度不高。

在 AI 融入编程实践的背景下, 课堂教学尚未形成稳定的人机协同学习链条。AI 工具往往停留在学生课后自发使用或临时求助层面, 尚未被有效嵌入课堂。学生缺少在教师指导下学习准确表达需求、理解 AI 生成代码、判断 AI 输出质量及修正 AI 错误的机会。项目化和任务驱动教学仍不充分^[14-16], 课程中缺少贯穿式、递进式的综合任务, 学生难以在真实项目中经历需求分析、方案设计、调试优化和成果表达等过程。

2.3 评价方式偏重结果考核, 难以识别 AI 协同过程能力

当前课程评价多以实验作业、上机编程和课程设计为主, 重点考查语法是否正确、规范, 如表 2 所示。这种考核方式虽可反映学生对基本语法的掌握情况, 但在 AI 协同教学背景下, 仅凭代码和运行结果, 难以判断学生是否真正理解代码逻辑, 具备合理使用 AI、评价 AI 输出和修改 AI 代码的能力。

表 2 评价方式概况

课程总评	满分 100 分			
	出勤	课堂参与	实验作业	课程设计
成绩构成	5%	5%	70%	20%

2.4 学习成果迁移不足, 专业问题解决能力培养不充分

由于课程内容与专业应用结合不够紧密, 学生在完成基础语法学习后仍难以将编程知识迁移到本专业实际问题中。学生需额外通过后续课程补学专业编程应用知识, 既增加了学习成本, 也使课程成果难以转化为持续的专业能力。

3 AI 协同教学改革设计

针对上述问题, 本文从课程内容、教学流程、AI 介入机制和多元评价体系四个方面进行回应, 构建 AI 协同教学改革方案, 如图 1 所示。

3.1 AI 协同教学的目标框架与能力结构

AI 协同教学是指在 Python 课程中将大语言模型和智能编程工具嵌入教学的全过程,引导学生利用 AI 完成问题分析、代码理解、错误调试和结果反思等学习活动。核心目标不是以 AI 替代学生编程,而是通过人机协同过程培养学生 AI 协同编程能力,具体为四个方面:

问题转化能力。能够从专业实践场景中识别问题、明确任务和处理流程,将实际需求转化为可计算、可编程的任务。

需求表达能力。能够向 AI 准确描述编程任务,撰写有效提示词,并通过多轮交互细化和修正需求。

代码理解与验证能力。能够读懂 AI 生成代码的核心逻辑,理解变量关系和执行流程,并结合运行结果判断代码是否合理。

修改与迁移能力。能够在理解代码的基础上,将已有代码或方法迁移到专业任务中,完成专业相关的实践应用。

3.2 构建四层递进式课程内容体系

为满足非计算机专业学生在 AI 时代的学习需求,本文围绕学生的认知基础、专业背景和未来职业需求,系统重构 Python 课程内容,形成“基础筑基—AI 辅助—专业应用—项目落地”四层递进式内容体系。

基础筑基层。该阶段聚焦 Python 核心语法、基本程序结构和常用程序逻辑。与传统课程追求知识点完整覆盖不同,基础筑基强调掌握核心语法,引导学生在代码编写、阅读、修改和验证过程中理解程序运行机制,为后续 AI 辅助编程和专业实践奠定基础。

AI 辅助层。该阶段引入大语言模型和智能编程工具,将 AI 作为编程学习的辅助支持。学生通过任务描述、提示词设计、代码解释、错误诊断和功能优化等训练,学习如何与 AI 进行有效交互,并在理解 AI 输出的基础上完成代码修改与结果验证。重点培养学生合理使用 AI、判断 AI 输出质量和修正 AI 生成代码的能力,避免简单复制和过度依赖。

专业应用层。该阶段面向不同专业设计场景化案例,引导学生将 Python 知识和 AI 工具应用于专业问题,提升学生问题建模能力和知识迁移能力。例如,商科可开展市场数据分析,物流可开展路径优化与调度模拟,艺术设计可进行 AI 辅助创意生成。

项目落地层。该阶段以个人或小组项目形式,引导学生围绕专业真实问题,完成从问题定义、需求分析、方案设计、代码实现、AI 辅助优化到成果展示的完整实践过程。培养学生跨学科协作、创新思维与专

业问题解决能力,最终形成可运行程序、分析报告、可视化作品等成果。

3.3 构建与课程内容体系相衔接的一体化教学流程

课程内容体系需通过具体教学流程加以落实。本文将任务导入、问题分析、知识学习、AI 协同练习、阶段测验、综合应用和成果展示等环节融入四个递进式教学阶段,形成内容体系与教学过程相统一的一体化教学流程。

核心知识学习与阶段测验。基础筑基层围绕核心语法和基本程序逻辑展开,通过代码阅读、程序改错和编程练习,夯实学生基本编程能力。同时,设置限制 AI 使用的阶段测验,考查学生对基础语法的掌握情况,防止学生过度依赖 AI 生成答案。

AI 协同案例分析与训练。AI 辅助层重点训练学生正确应用 AI 完成具体编程任务。通过 AI 协同编程案例讲解与分析、课堂限时 AI 协同训练任务,考查学生 AI 协同编程能力。

问题分析与场景化任务实践。专业应用层围绕不同专业设计场景化任务,引导将专业问题转化为可编程任务。学生在 AI 辅助下完成数据处理、可视化分析、自动化操作或创意设计等实践,提升问题建模、任务拆解和专业应用迁移能力。

综合应用与成果展示。项目落地层以个人或小组项目形式开展。学生围绕本专业真实问题,完成问题定义、方案设计、代码实现、AI 辅助优化和成果展示。项目完成后,学生展示成果、核心代码、AI 交互记录和反思过程,教师进行综合评价。

3.4 设计阶段性 AI 介入机制

AI 协同教学的关键并不在于简单地允许学生使用 AI 工具,而在于根据教学目标、学习阶段和任务难度,合理设计 AI 介入的时机、方式和边界。基础学习阶段若过早依赖 AI,可能削弱学生对语法规则和程序逻辑的理解;综合实践阶段若禁止 AI,又会使教学过程脱离真实的智能化编程环境。因此,本文设计阶段性 AI 介入机制,引导学生在不同学习阶段合理使用 AI。

限制使用 AI。在基础筑基阶段,AI 使用应以限制为主,重点培养学生对基本语法的理解能力。学生应独立完成代码阅读和简单程序编写任务,避免使用 AI 生成答案。

引导 AI 辅助。在 AI 辅助阶段,允许学生借助 AI 生成代码、修改代码、迭代代码,要求学生保留 AI 交互记录,包括提示词、AI 建议、修改过程等,使 AI 使用过程成为可追踪、可评价的学习证据。

开放 AI 协同。在专业应用阶段，鼓励学生合理使用 AI 参与专业问题分析、任务拆解、代码框架生成、错误修复和结果优化。该阶段强调真实情境下的人机协同编程，学生仍需保持主体地位。

审视 AI 输出。在项目落地阶段，引导学生对 AI 使用过程进行反思，分析 AI 生成代码的正确性和局限性，识别逻辑漏洞、冗余设计等问题。学生需提交 AI 使用说明和项目反思报告，从会使用 AI 进一步转向会评价和修正 AI 内容。

3.5 建立多元评价体系

为适应 AI 协同教学要求，课程评价不能仅以程序运行结果和语法正确性为依据，而应同时关注学生的基础编程能力、AI 协同过程、专业应用水平和综合实践表现。为此，本文构建多元评价体系，推动评价方式由结果考核转向过程性评价与成果性评价相结合。基础能力评价。考查学生在限制 AI 使用条件下对

Python 核心语法、程序结构和基本逻辑的掌握情况。评价内容包括代码阅读、程序改错、代码补全和简单编程任务，确保学生具备必要的独立编程基础。

AI 协同过程评价。考查学生使用 AI 辅助编程的过程能力，包括提示词设计、AI 交互记录、代码修改过程、AI 输出判断和结果验证等。该评价重点不在于学生是否使用 AI，而在于其是否能够合理使用 AI、理解 AI 生成代码并对其进行修正和优化。

专业应用评价。考查学生能否将 Python 知识和 AI 工具应用于本专业任务，如数据处理、可视化分析、自动化操作、图像处理等。重点关注学生能否将编程知识迁移到具体专业问题中。

综合项目评价。考查学生在综合项目中解决实际问题的能力。评价内容包括项目选题、需求分析、功能实现、代码质量、AI 使用过程、成果展示、答辩表现和反思总结等。

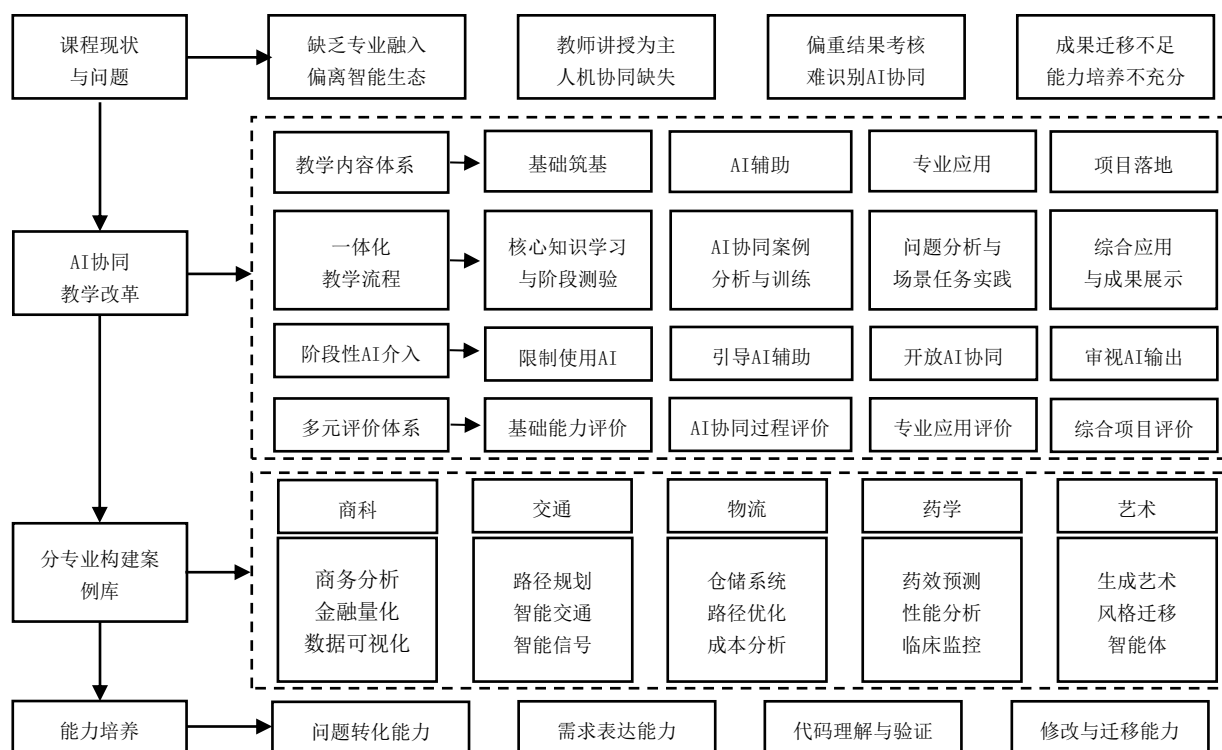


图1 AI 协同教学改革方案

4 改革成效

为检验改革成效，本文以本校非计算机专业 Python 程序设计课程为实践对象，选取改革前后两个教学阶段的学生学习数据进行比较分析。数据主要来源于课程平台记录、实验作业提交情况、阶段测验成绩、综合项目评价结果以及课堂观察记录。结果表明，

AI 协同教学改革在提升学生学习兴趣、优化学习行为和增强实践能力等方面取得了较好成效。

学生参与度有所提升。改革前，学生对 Python 课程的认知多停留在学习语法和完成编程作业层面，部分学生认为课程与自身专业关联不强，学习主动性不足。改革后，课程通过专业案例导入和任务驱动教学，学生能够更直观地理解 Python 的应用价值。从课堂观

察以及课程平台统计数据看，改革后学生课堂注意力更加集中，前排就座率、课堂抬头率和出勤率均较以往有所提高。

实验作业完成质量持续优化。从平台统计和课堂记录看，学生实验作业完成情况较之前有显著提升，完成率从 65% 提升到 73%。其中，商科类专业学生平均完成率甚至达到 90%，AI 辅助学习在一定程度上降低了学习难度，提升了学生完成实验作业的积极性 and 质量，如表 3 所示。

表 3 改革前后教学效果数据对比

评价指标	改革前	改革后	变化情况
平均出勤率	75%	82%	提升 7%
实验作业完成率	65%	73%	提升 8%
阶段测验平均分	54 分	64 分	提升 10 分
综合项目完成率	76%	89%	提升 13%

综合应用能力显著增强。学生能够结合编程知识与 AI 辅助工具，完成路径设计、商务数据分析等与专业相关的综合项目实践，综合项目实践完成率提升 13%，体现出较好的知识整合和问题解决能力，如图 2 所示。

课外拓展能力不断提升。部分学生将课堂学习延伸至课后自主探索，继续深入学习 AI 工具和编程技术，积极参加 Python 程序设计及相关竞赛并获得省级及国家级奖项。



图 2 改革成效

5 结语

人工智能的发展正在重塑程序设计课程的教学环境。对于非计算机专业 Python 课程而言，教学改革不能停留工具引入层面，而应围绕课程目标、内容体系、教学流程、AI 介入机制和评价方式进行系统重构。

针对当前课程中语法训练偏重、专业场景融入不足、人机协同过程缺失和评价方式单一等问题，本文构建了四层递进式课程内容体系，设计了一体化教学流程，提出了阶段性 AI 介入机制和多元评价体系。教学实践表明，该改革有助于提升学生课堂参与度、代码理解能力、AI 协同编程能力和专业问题解决能力，促进 Python 课程由语法训练向能力培养导向转变。

后续仍需进一步完善 AI 使用规范，细化 AI 参与过程评价标准，持续建设“AI+专业”案例库，并根据不同专业学生的基础差异优化分层任务设计，以提升该教学模式的可操作性、可评价性和可推广性。

参考文献

[1] 中华人民共和国教育部. 教育部关于印发《高等学校人工智能创新行动计划》的通知[J]. 中华人民共和国教育部公报, 2018(04):127-135.

[2] 郭文佳, 傅向华, 吴庆阳, 等. 基于 CDIO 理念的 C++ 程序设计课程项目式教学实践[J]. 计算机技术与教学学报, 2021(09):69-73.

[3] 孟文龙, 张策, 张小东. 新工科背景下“计算机程序设计基础”课程教学改革探索[J]. 计算机技术与教学学报, 2024(07):85-90.

[4] 陈小强, 黄志鹏, 胡翰. 生成式 AI 赋能 Python 程序设计课程教学设计与实践[J]. 计算机教育, 2025(12):109-113.

[5] 俞文昌, 马小琴, 赵盼. 工程认证背景下 Python 程序设计课程改革[J]. 计算机教育, 2025(11):154-159.

[6] 李知菲, 彭浩, 王晖. 基于知识图谱的 Python 程序设计课程个性化 AI 助学模式探索[J]. 计算机教育, 2025(08):200-205.

[7] 周丽蓉, 全华凤, 李宜汀, 等. 财经高校 Python 程序设计课程改革实践路径[J]. 计算机教育, 2025(09):202-207.

[8] 嵩天, 黄天羽. Python 语言程序设计教学案例新思维[J]. 计算机教育, 2017(12):11-14, 19.

[9] 肖敏, 陈永强. 面向人文社科专业的 Python 程序设计课程改革[J]. 计算机教育, 2021(05):45-48.

[10] 易发胜, 李立, 赵丽琴. 新工科背景下 Python 程序设计课程教学方法研究[J]. 计算机教育, 2021(07):148-151, 156.

[11] 李莉, 郭祥云, 吕金娜, 等. 大语言模型背景下 Python 程序设计课程教学改革探索[J/OL]. 软件导刊, 1-5 [2026-05-04]. <https://link.cnki.net/urlid/42.1671.TP.20260414.1404.029>.

[12] 李辉东, 朱琼瑶. 数智化转型下教师+AI 二元协同教学实践研究——以高职院校 Python 程序设计课程为例[J]. 电脑知识与技术, 2026, 22(02):144-146.

[13] 王瑞. 人工智能技术在“Python 程序设计”课程中的应用研究[J]. 科技资讯, 2026, 24(01):223-226.

[14] 胡辉, 马驰. 项目驱动的 Python 程序设计课程教学模式[J]. 福建电脑, 2023, 39(08):109-112.

[15] 张越, 罗必露, 杨志超, 等. 基于“Python 程序设计”课程的教学改革路径探究[J]. 科技风, 2025(06):123-125.

[16] 鲁迪, 李粟, 刘生智, 等. 项目式教学在“Python 语言程序设计”课程中的应用与实践[J]. 科技风, 2024(27):120-122.