

数智化项目式教学在《Python 程序设计》课程中的研究与实践

李颖

哈尔滨剑桥学院 黑龙江 哈尔滨 150001

【摘要】：在“人工智能+高等教育”背景下，针对传统 Python 程序设计课程 PBL 教学痛点，本研究探索构建数智化 PBL 教学新范式，提出“理论构建—工具应用—教学实践—评估优化”四阶段闭环迭代教学模型。利用对话式 AI 大模型与 IDE 智能编程插件融入教学全流程，该模式有助于降低学生认知负荷，提升代码编写效率与项目完成率，激发学习驱动力。针对学生易陷入“AI 依赖”问题，课题组提出增设提示词工程模块与“分级设限”考核等优化策略。本研究实现教学“规模化”与“个性化”统一，为应用型本科高校计算机基础课程数智化教学改革提供可落地实践路径。

【关键词】：数智化教学；项目式学习（PBL）；Python 程序设计；人工智能；人机协同

DOI:10.12417/2705-1358.26.06.012

1 引言

随着大数据与生成式人工智能技术的爆发式增长，“数智化”理念快速且深入地渗透到教育领域，重塑高等教育教与学生态^[1]。2024 年 4 月，教育部高等教育司公布首批 18 个“人工智能+高等教育”应用场景典型案例，标志我国教育数字化转型迈出坚实一步^[2]。在计算机程序设计课程领域，这些案例有很强的示范效应，如北京邮电大学的“码上——大模型赋能的智能编程教学应用平台”、华中师范大学的“基于小雅平台的智能场景创设”、西安电子科技大学的“AI 赋能督导新模式”等。这些创新实践提升了课堂教学效果，激发了学生的探究兴趣与内驱力，促进了学生创新思维和实践能力培养。Python 程序设计课程语法简洁、应用广泛，是人工智能时代重要教学载体，需引入智能教学模式以适应产业对创新型人才的需求。

项目式教学（Project-Based Learning, PBL）是强调通过真实任务驱动学习的教学模式，其理论源于美国教育家杜威的“做中学”理念，该理念为其奠定基础。在国际计算机教育领域，PBL 已形成成熟体系，如麻省理工学院的“Scratch”图形化编程项目、卡内基梅隆大学的“Learning by Doing”理念^[4]。近年来，人工智能技术发展迅速，开发人员广泛使用 AI 编程助手，如 GitHub Copilot、Baidu Comate、iFlyCode 等，这些工具可自动完成部分编码和测试工作，提高编程效率^[5]。

虽数字技术在教育领域应用渐广，但教学改革要警惕传统模式与技术生硬结合。本研究聚焦 Python 程序设计课程，构建“数智化项目式教学”理论框架，突破传统 PBL 与数字技术简单叠加模式。将“数智化”与“项目式教学”深度结合，提出

闭环教学模型，为程序设计类课程教学提供新视角和方法。构建智能化教学平台，开展个性化学习，引入 AI 编程助手加强实践与反馈，为学生提供智能化、个性化、高效化学习体验，对推动应用型本科院校计算机程序设计类课程教学改革创新、提高学生编程兴趣和能力有重要实践意义。

2 数智化项目式教学（PBL）的理论框架构建

传统项目式教学（PBL）秉持建构主义学习理论，强调以学生为中心，通过真实复杂的任务驱动学生进行自主探究与协作学习^[6]。然而，在面对具有较高抽象性和逻辑性的计算机程序设计课程时，传统 PBL 往往面临教师指导难以兼顾全体、学生遇到代码报错容易产生挫败感、项目评价手段相对单一等局限。数智化教育理念的引入，并非是对传统 PBL 的简单否定，而是利用生成式人工智能、大数据等前沿技术对其进行赋能与重塑。数智化 PBL 将静态的任务驱动转化为动态的智能交互，使学习过程从“规模化标准化”向“规模化个性化”跨越。在此理念指导下，本研究将智能技术作为“认知脚手架”，深度融合于项目设计、实施与评价的各个环节，从而构建出适应新时代人才培养需求的数智化 PBL 理论新范式。

为保障数智化 PBL 在计算机程序设计课程有效落地，本课题提出并构建“理论构建—工具开发—教学实践—评估优化”四阶段闭环迭代教学模型。此模型以系统工程思想为指导，各阶段相互衔接、双向反馈。在“理论构建”阶段，依托相关理论确立数智化教学核心目标与评价标准，提供理论支撑；在“工具开发与集成”阶段，根据课程特性甄选或二次开发适切工具，匹配教学需求；在“教学实践”阶段，引入数智化平台，学生

基金项目：本文支持黑龙江省高等教育学会“党的二十届三中全会、2024 年全国教育大会专项课题”《数智化项目式教学在计算机程序设计课程中的研究与实践》（项目编号：24GJZXC020）的阶段性研究成果；本文支持哈尔滨剑桥学院 2024 年度校级教育教学改革项目《探索 PBL-C 教学模式在计算机专业课程中的构建》（项目编号：JQJG2024002）的阶段性研究成果。

在教师引导下用智能工具开展项目分析等工作；在“评估优化”阶段，通过智能平台采集学生学习数据进行多维评价，将结果反哺至理论构建与工具配置，形成闭环生态持续优化。

将 AI 编程助手融入 IDE，核心目的不仅是“提高写代码的速度”，更重要的是“培养计算思维与问题解决能力”。借助“智能代码生成与补全”为学生提供基础代码结构或语法指引，帮助初学者突破 Python 语法记忆障碍，集中精力设计算法逻辑；“精准错误诊断”功能在程序运行失败时，让 AI 助手快速定位问题代码，用自然语言说明报错缘由，替代教师手动排查；“启发式优化建议”在代码成功运行后，从时间复杂度、空间复杂度或代码规范性等方面给出改进意见，推动学生优化代码；“伴随式交互学习”让学生通过对话框随时向 AI 咨询，获得 24 小时“个性化助教”支持。

3 基于 Python 课程的数智化 PBL 教学流程设计

将数智化项目式教学（PBL）的理论框架真正落地于真实的教学环境，需要依托科学严谨的教学流程设计。鉴于从零开发底层 AI 大模型的高昂成本与技术壁垒，本研究主张“敏捷集成与应用”的理念，即全面引入国内成熟且免费的生成式人工智能工具（如对话式大模型 Kimi、文心一言，以及 IDE 智能编程插件“通义灵码”、BaiduComate 等），并将其深度嵌

入 Python 课程的教学全生命周期中。针对应用型本科学生的认知特点，本教学流程设计涵盖了“需求分析→项目设计→开发实践→动态反馈”四个核心阶段。

在流程的前期（需求分析与项目设计阶段），教学的重点在于帮助学生跨越从自然语言到计算逻辑的认知鸿沟。传统教学中，学生面对现实编程项目常无从下手，本研究通过引入对话式大模型作为“思维引导者”，使学生在与 AI 的启发式交互中，逐步完成复杂任务的拆解与底层数据结构的规划，培养其计算思维。

在流程的中后期（开发实践与动态反馈阶段），教学场景转移至集成开发环境（IDE）。智能编程插件在此发挥了“伴随式助教”的核心作用，不仅提供沉浸式的编码辅助与排错支持，降低初学者的编程挫败感，还能在项目完成后提供代码重构建议。与此同时，依托教学管理平台采集的多模态交互数据，教师能够全面掌握个体与班级的学情，实现“智能底层辅助+教师顶层监控”的双重反馈闭环。

为直观呈现上述数智化工具在项目式教学各环节中的具体赋能机制与实操细节，本研究以 Python 基础项目“简易个人记账本”为例，详细梳理了全流程的教学设计方案，具体如表 1 所示。

表 1 基于 Python 的数智化 PBL 教学流程及实践案例设计表

教学阶段	核心教学任务	应用的数智化工具及核心功能	“简易个人记账本”案例具体实践	预期教学成效
一、需求分析	识别现实问题 拆解任务边界	对话式大模型 功能：启发式对话、需求拆解	教师提供提示词引导，学生向 AI 提问：“如何用 Python 写记账本？”AI 协助将模糊需求拆解为：记录收支、查看账单、计算结余等具体功能模块。	降低认知门槛，帮助初学者将非结构化的自然语言转化为清晰的结构化需求列表。
二、项目设计	规划数据结构 构建算法逻辑	对话式大模型 功能：逻辑梳理、伪代码生成	学生用自然语言描述业务逻辑，AI 建议使用“字典”存储单笔账单，“列表”存储所有账单，并引导构思基于 while 的主循环架构。	形成人机协同设计，在动手写代码前建立清晰的计算逻辑基座，培养计算思维。
三、开发实践	代码编写实现 排错与功能测试	IDE 智能编程插件 功能：代码补全、错误诊断	编码时：AI 实时补全 <code>account_list.append(...)</code> 等语句。报错时：针对初学者的常见错误，AI 一键生成易懂的中文排错解释。	大幅减少因基础语法拼写或环境配置导致的挫败感，提供沉浸式的伴随编码体验。
四、动态反馈与迭代	代码重构优化 多维学情评价	IDE 智能插件与教学管理平台 功能：代码审查、过程数据采集	优化环节：AI 建议将繁琐的循环优化为“列表推导式”，或增加文件读写操作持久化数据。 评价环节：教师通过后台数据查看交互频次与代码修改轨迹。	提升学生代码规范性与工程化意识；实现“智能底层辅助+教师顶层监控”的双重反馈闭环。

4 多维教学效果评估与持续优化分析

在 Python 程序设计课程中引入 AI 工具的数智化 PBL 教学模式后，为了科学衡量该模式的实际成效，本课题打破了传统“以期末纸笔考试为主”的单一评价方式，构建了“过程与结果并重、人机数据结合”的多维教学效果评估体系，并基于应用型本科院校的真实学情，对教学实践进行了深入的反思与优化。

1. 多元化与全过程教学评价体系：数智化 PBL 模式核心是过程动态交互，评价体系需全周期覆盖。本研究从三个维度构建评价指标：一是基础编程与工程能力，评估项目代码运行情况，通过 IDE 后台及教学管理平台提取学生代码提交频次、Bug 修复时间等数据；二是计算思维与问题解决能力，分析“提示词交互记录”，评估学生将自然语言转化为结构化逻辑及独立排查错误的能力；三是协作沟通与创新素养，关注学生在项目互评和代码审查环节的表现。

2.教学成效对比与优势分析:将采用数智化 PBL 模式的教学班与采用传统讲授模式的平行班对比,教学成效显著提升。一是编程效率与项目完成率大幅提高,传统模式下初学者常因环境配置或缩进报错停滞,智能编程插件辅助下,学生解决基础语法耗时锐减,项目按时交付率上升。二是有效激发学生内驱力与自主学习意愿,AI 助手“伴随式解答”打破上课下时间壁垒,让学生摆脱对教师的绝对依赖,实现从“要我学”到“我要学”的转变。此外,借助 AI 代码重构建议,学生代码规范意识明显增强,写出的 Python 代码更具可读性和工程化标准。

5 结语与展望

本研究立足“人工智能 + 高等教育”背景,针对传统计算机程序设计课程教学痛点,创新提出基于 Python 课程的数智化项目式教学(PBL)新范式。通过构建闭环迭代模型,将

国内成熟生成式 AI 大模型与智能编程辅助工具融入项目教学全周期,实现教学“规模化”与“个性化”统一。实践表明,该模式打破传统课堂时空局限,降低初学者编程认知负荷,提升代码编写效率与项目交付质量,重塑学生计算思维,培养人机协同探究与工程实践素养。

展望未来,数智化教育探索任重道远。基于本阶段研究成果与反思,课题组未来着重推进三方面工作:一是拓宽学科应用,推进跨课程群建设,将数智化 PBL 教学模型从 Python 基础程序设计课程推广至 Java 等更复杂的计算机专业核心课程,验证并完善其普适性与灵活性;二是深化智能教学资源生态建设,依托教学实践,持续建立并扩充“数智化项目库与教学资源库”,设计适配多领域的真实项目案例,为跨学科编程教学提供实战素材;三是开展学生长效发展追踪,建立长效评估机制,对参与教改的学生群体进行长期学习能力与职业发展跟踪,关注“人机协同学习经历”的后续影响。

参考文献:

- [1] 梁俊斌,谢贇,唐振华.生成式人工智能赋能教育数字化转型的机理、挑战与对策[J].现代教育管理,2025,(11):15-23.
- [2] 教育部高等教育司.教育部高等教育司关于公布首批“人工智能+高等教育”应用场景典型案例的通知[Z].教高司函(2024)3号,2024.
- [3] Bell,S.(2010).Project-based learning for the 21st century:Skills for the future.The Clearing House:A Journal of Educational Strategies,Issues and Ideas,83(2),39-43.
- [4] Resnick,M.,Maloney,J.,Monroy-Hernández,A.,et al.(2009).Scratch:programming for all.Communications of the ACM,52(11),60-67.
- [5] 邢万勇,廖伟军,杨智帆.基于 AI 编程助手的软件研发技术效能提升实践[J].数字技术与应用,2025,43(01):28-30.
- [6] 陈伟.PBL 教学法在农业高职院校 Python 程序设计教学中的应用研究[J].郑州铁路职业技术学院学报,2025,37(03):92-95.