AI 时代计算思维导向的 Python 程序设计课程教学改革与实践*

陆冰琳** 刘芳 陈建松

广东东软学院，佛山 528225

摘 要 随着 AIGC 技术的普及，代码生成工具已成为学生编程学习的常用辅助手段，在降低编程门槛的同时，也催生了学生的学习惰性与代码验证能力缺失等问题。针对 AI 时代编程教育中学生计算思维培养不足、人机协作能力薄弱的痛点，本文依托 Python 程序设计课程，构建四层递进式数学动画项目式教学。教学设计中以 Manim 数学动画为可视化项目载体，借助可视化屏障、参数敏感性等形成天然 AI 反制机制，通过"静态—动态—交互—拓展"四层阶递进项目链，系统培养学生的验证、算法、系统与抽象思维。教学实践表明，该模式可在一定程度上遏制学习惰性，提升学生的问题拆解、逻辑建构与代码验证能力，形成了场景驱动、思维进阶的良性教学闭环，为 AI 时代程序设计课程的教学改革提供了可复制的实践路径。

关键词 Python 教学，计算思维，人机协同，Manim

Teaching Reform and Practice of Computational Thinking-Oriented Python Programming Course in the AI Era *

Binglin Lu Fang Liu Jiansong Chen

School of Information Management and Engineering Neusoft Institute
Foshan 528225, China
lubinglin@nuit.edu.cn

Abstract—With the widespread adoption of AIGC technologies, code generation tools have become prevalent aids in students' programming learning. While lowering the barrier to programming, these tools have also induced learning inertia and deficiencies in code verification capabilities. To address the inadequacies in computational thinking cultivation and human-AI collaboration skills in AI-era programming education, this study proposes a four-tier progressive mathematical animation project-based teaching model within a Python programming course. Leveraging Manim mathematical animations as the visualization vehicle, the instructional design forms a natural safeguard against AI over-reliance through visual barriers and parameter sensitivity. Through a four-level progressive project chain spanning static, dynamic, interactive, and extended phases, the model systematically develops students' verification thinking, algorithmic thinking, systems thinking, and abstract thinking. Empirical results indicate that this model effectively mitigates learning inertia, enhances students' problem decomposition, logical construction, and code verification abilities, and fosters a virtuous closed-loop teaching paradigm that is scenario-driven and thinking-progressive, offering a replicable practical pathway for programming course reform in the AI era.

Keywords—Python programming instruction; computational thinking; human-AI collaboration; Manim

1 引 言

在人工智能席卷各行各业的今天，教育也站在变革的十字路口，传统编程教育模式正面临前所未有的挑战。高校学生在编程课程中广泛使用 AI 代码生成工具已成为常态，AI 工具虽然可极速提高编程效率，降低编程门槛，但因其便捷性催生了新的教育困境。许多学生对于 AI 工具的使用仅停留在“复制-粘贴”的表层应用，在编程入门阶段即产生学习惰性，进而难以通过实践过程培养计算思维与问题解决能力。

这一矛盾凸显了技术应用与教育目标之间的深层冲突。《教育部等九部门关于加快推进教育数字化的意见》[1]提出：“坚持以人为本，着力培养学生高阶思维、思考判断能力、实践能力。”政策导向与当下困境形成鲜明对照，如何让 AI 工具从“答案供应商”转

*基金资助：广东省高等教育学会“十四五”规划2025年度高等教育研究课题——生成式AI与数学动画项目融合驱动——面向计算思维培养的教学模式研究（25GQN027）；粤港澳大湾区高校在线开放课程联盟2026年教育教学研究和改革项目——双线驱动·双元思辨 《程序设计语言》智慧课程建设与实践（WGKM2026139）；2023年度广东省本科高校教学质量与教学改革工程建设项目——大语言模型驱动的翻转课堂与同伴回答教学法在跨境电子商务课程中的应用研究（粤教高函[2024]9号
**通讯作者：陆冰琳 lubinglin@nuit.edu.cn

变为“思维训练伙伴”，成为编程教育改革的关键命题。围绕这一命题，广大教育工作者从计算思维培养、AI 视阈下的教学模式构建、智能化教学工具开发等多个维度展开探讨，形式了丰富的理论积累和实践经验。

2 研究现状

2．1 计算思维内涵演进与培养路径研究

伴随时代的变革，计算思维的概念经历了从 1.0 到 3.0 的演进历程。“计算思维”概念于 20 世纪 80 年代由麻省理工学院 S.Papert[2]首次提出。其后，Jeannette M. Wing[3]将“计算思维”进一步明确定义为“利用计算机科学的一些基本概念解决问题、设计系统和理解人类行为。”标志着计算思维 1.0 时代的开启。随着大数据和人工智能技术的发展，陈国良等[4]阐述了在大数据推动下计算思维所赋予的新内涵和形式，将其称为计算思维 2.0。

自计算思维的概念提出以来，关于“计算思维培养策略”、“计算思维培养模式”等研究也相继展开。郁晓华等[5]指出，计算思维培养的核心在于将计算机科学解决问题的方法和过程扩展到其他学科，构建目标问题域与自动化解决方案的桥梁。姜洋[6]提出的认知分成模型中，基于数学思维与计算思维的高度同构性，将编程求解数学问题的过程进行横向和纵向演绎，进而培养计算思维。项目式教学方面，丁世强等[7]基于计算思维发展四要素设计了项目式教学模式，王亚杰等[8]则采用"课内贯穿式项目+课外专项综合性项目"思路，将项目式教学与计算思维的培养相结合，助力学生形成系统思维与问题解决能力。

然而，以上研究多聚焦通用编程能力的整体培养，尚未形成结合 Python 语言特性与应用场景的计算思维阶梯式训练体系。随着 AI 技术的普及，桂小林[9]指出，计算思维正向 3.0 时代进化，教学层面需应对人机协作的新挑战。如何回应 AI 代码生成工具普及带来的范式冲击成为关键。

2．2 AI 时代编程教育模式转型探索

在 AI 辅助编程成为常态的情况下，迫使编程教育从语法习得向思维培养转型，也为计算思维 3.0 时代的教学模式转型提供了新的可能。

在教学模式层面，翟雪松等[10]提出逆向工程教学模式，主张以答案为起点，以问题为导引，通过人机协作反向拆解知识体系，契合计算思维的培养内涵。李秀等[11]等则以人机双师互补、协同为理论框架，构建由 GenAI 赋能的人机双师协同教学模式。贺桂娇等则[12]通过动态案例设置，智能工具辅助和工程实践引导形成三阶递进的教学模式。两者均强调教师与 AI 助教在“课前-课中-课后”的教学角色配置及教学内容的迭代优化，助力学生 AI 数字能力和计算思维的培养。张金等[13]结合布鲁姆认知学习层级，提出大模型时代的计算机系统能力分层培养目标，将编程任务与不同阶段问题相映射，借由与大模型的反复交互达到评测、遴选、迁移、交互等能力的锻炼，为计算思维的分层递进培养提供保障。

在智能工具层面，厉旭杰等[14]针对传统在线编程平台的缺陷，提出了集成大语言模型的在线编程实验平台设计方案，引入 GLM4 和 GPT4 为学生提供代码评估、诊断和优化的交互场景，为计算思维的个性化培养创造了条件。蒋杰等[15]提出以 AI 数字人行为为驱动的编程教学设计方案，突出了数字人在陪伴式教学中进行个性化的信息收集、分析及反馈的优势，实现技术工具单项传递到思维过程双向互动的角色跃迁。何升[16]搭建了以数字徽章为驱动的教学机制，由智能伴学系统评判学生学习数据，发放低阶徽章，教师则基于交互质量、计算思维能力的评判发放高阶徽章。

2．3 现有研究的局限与本文研究定位

综上所述，计算思维的研究已从 1.0 时代的概念界定演进至 3.0 时代的人机协作新阶段，AI 时代下的编程教育探索也在教学模式和工具开发层面取得进展。然而，两者尚未形成深度耦合。现有“AI+编程”的教学模式多聚焦于协同增效，针对 AI 幻觉与惰性对思维发展的抑制效应研究较少。特别是如何结合特定编程语言特性和应用场景构建计算思维阶梯式训练，如何在教学中通过系统性设计抑制 AI 诱发的学习惰性问题，仍是亟待解决的重要问题。

基于此，本研究提出如图 1 所示的 AI 协同下计算思维导向的四层递进项目式教学模式。

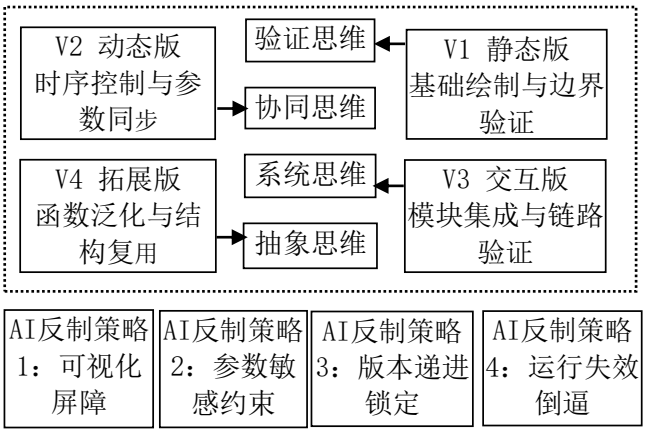


图 1 计算思维导向的四层递进项目式教学模式

本研究在以 Python 为教学语言的《程序设计语言》课程中践行了该教学模式，教学中以 Python 第三方模块 Manim 数学动画引擎为项目载体，该模式的研究重点在于从探讨如何使用 AI 辅助编程，转向设计 AI 倒逼思维的培养机制，通过数学问题的确定性约束和动

画效果的即使反馈，以及针对性的 AI 反制机制，使学生的编程形式从“代码复制”向“代码读、验、改”转变，为 AI 时代 Python 教学中计算思维的培养提供可落地的实践方案。

3 计算思维导向四层递进教学模式

3．1 教学模式的整体设计

本研究的教学模式构建基于《程序设计语言》课程展开，该课程为本校信息管理与工程学院电子商务专业开设的专业基础课程，基于 Python 语言开展项目式教学。教学模式以 Python 第三方模块 Manim 为桥梁，将数学动画项目融入编程实践，实现数学与编程跨学科融合。数学作为基础学科，既是编程逻辑底层支撑，也是跨学科融合的核心纽带。教学设计中采用“数学问题+编程问题”的双线驱动，通过数学问题明确动画目标，通过编程问题指引实现路径，使学生在抽象思维与具象代码之间不断转换，可更有效地培养学生问题拆解与逻辑构建的能力。

数学动画项目以线性函数的动画制作为起点，设置了“静态-动态-交互-拓展”四个版本，与验证思维、协同思维、系统思维、抽象思维四个维度相互映射。对于电商专业的大一学生而言，可基于相对成熟的数学思维借题发挥，借数学问题发挥其对计算思维发展的正向迁移作用。动画形式的项目既保留了其可视化成果的即时反馈优势，也为学生预留了更多的创意表达的空间，相较于传统代码项目更易于激发学生的学习兴趣。通过培养学生可视化表达与数据呈现能力，为后续电商新媒体运营、商业数据分析等专业场景奠定基础。

3．2 AI 反制策略的整体设计

AI 代码生成工具的普及，使编程从手动编写向人机协作转变，这一改变使得学生一定程度上无需学习编程语言即能获取可运行的代码，使得课程中学生本因基于实践积累的问题拆解和逻辑推导的思维环节被架空。针对这一情况，本文研究的教学模式中蕴含了四项 AI 反制机制，层层设障，促使学生在人机协作的基础上培养“读-验-改”代码的能力，实现从复制代码到思维培养的转型。

策略 1：可视化屏障。常规 Python 入门课程中的编程项目需求描述和运行结果均为文字形式，学生易于直接将题目要求作为提示词生成精准的代码。本项目以动画效果为项目需求，直接使用大语言模型难以一步复现准确代码。学生操作中需在理解 AI 生成代码与视觉呈现效果之间映射关系的前提下，再进行手动调试。

策略 2：参数敏感约束。数学动画中公式参数与坐标轴参数的组合效果高度敏感，AI 生成的代码参数配置往往无法直接适用，需要学生在多轮调试中验证渲染效果，可深化学生对数学问题与编程问题之间的转换逻辑。

策略 3：版本递进锁定。项目设计中四个版本由易至难层层递进，前置版本的理解不足将影响后续版本的实现效果。版本递进中学生需对前期项目代码进行重构，难以通过 AI 跳跃生成，使学生在按序完成认知构建的过程中熟练掌握从 Python 单一模块到多模块的应用。

策略 4：运行失效倒逼。AI 生成的代码常因数学动画引擎原版 ManimGL 和社区版 Manim 的版本差异、边界适配、渲染逻辑等问题导致运行失败或有效果偏差。学生需在读取错误提示的基础上，理解代码逻辑，对错误根源与预期差异进行验证，激发其逆向思维能力。

4 教学模式的实践应用

4．1 课程实践项目教学目标

《程序设计语言》作为电子商务专业大一学生的编程入门课程，希望学生通过 AI 协同下计算思维四层递进的项目式教学模式，在掌握 Python 核心语法的前提下，以 Manim 模块的应用为起点理解模块化与面向对象思想，熟练掌握 Python 内置、第三方、自定义模块的应用，通过数学动画项目系统培养学生的计算思维，为其后续各专业课程中基于 Python 语言进行新媒体交互设计、数据分析与可视化等综合性项目的开展奠定坚实基础。

项目实施过程中采用过程性评价与成果性评价相结合的方式，前者主要通过在调试日志记录人机协作问题与解决路径，后者则聚焦于动画作品与代码质量。最终提交报告和项目源代码，作品择优在课堂进行展示。

4．2 各版本项目设计思路

（1）V1 静态版：验证思维培养

静态版项目要求学生针对线性函数 y=mx+b，为其参数斜率 m 和截距 b 设置固定值，利用 Manim 模块生成基础动画。该阶段的核心任务在于建立数学问题向编程问题的准确映射，具体涉及坐标系的合理划分、显示边界的包含关系判定等数学要素。在编程实践中需转换为 NumberPlane 对象的参数配置，plot()方法中匿名函数的调用以及坐标轴取值范围的自适应计算等具体技术细节。

项目实施过程中，由于 AI 倾向于生成固定组合的坐标范围，学生在无精确提示词描述的前提下使用大

语言模型生成代码可能会导致特定参数组合下对应图像效果显示不佳的问题。由此反向督促学生必须通过自主调试，在“代码语法无误”与“视觉效果良好”之间建立验证机制，也为项目下一阶段的动态版参数追踪奠定数学基础。

（2）V2 动态版：协同思维培养

动态版在静态版的基础上引入参数动态变化的机制，要求学生在动画中展示斜率 m 和截距 b 的连续变化对线性函数产生的综合影响。该版本的核心编程问题转向基于 ValueTracker 的状态追踪机制，学生需通过设置自定义函数建立参数间的协同关系，并重点解决动画呈现过程中函数图形和函数公式变化的同步性问题。

此阶段学生将面对的 AI 陷阱为动画时序幻觉，由于 AI 无法完美预测实际动画效果，可能会导致函数参数在坐标系内的图像和公式标签之间异步显示等问题。要求学生需通过设计统一的 update 回调函数建立参数协同关系，培养从识别要素关联性到建立协同规则的思维习惯，并为 V3 交互版本提供多要素协同模版。

（3）V3 交互版：系统思维培养

交互版要求学生借助 Python 第三方模块 Gradio 构建如图 2 所示的 Web 交互界面，实现用户通过页面自定义线性函数参数并获取对应动画视频的完整功能。该版本将程序边界从命令行拓展至 Web 系统，其中涉及从用户输入到视频反馈的完整链路处理。

在这一复杂链路中，AI 工具易出现跨框架上下文断裂，即大语言模型在处理 Gradio 前端组件与 Manim 后端渲染的衔接时容易发生各类错误。如选择的页面输入组件与各类动画参数匹配度的差异，参数取值范围与数学动画所要传递意图不符，缺失部分参数的合法性验证步骤、Jinja2 模版渲染问题以及文件传输路径参数等导致的技术环节脱节问题。

为妥善处理此类问题，学生需系统追踪 UI 组件，处理 Manim 动画与视频文件之间的完整数据流，验证端到端之间的一致性。且在无针对性提示词的情况下，大语言模型倾向于将整个处理流程堆叠在单一文件中，学生需在此阶段有意识地将其拆分为职责清晰的自定义模块，以区分界面组件层、业务逻辑层和动画渲染层，从而掌握分解模块、验证链路、端到端调试的系统思维方法，为 V4 版本扩展提供系统框架。

（4）V4 拓展版：抽象思维培养

拓展版要求学生在完成线性函数交互界面的基础上，扩展实现幂函数动画生成界面，其中需重点关注负指数情况下的渐近线特性和定义域在零点的处理等数学问题。

此阶段 AI 工具可能表现出长程上下文遗忘，即倾向于忽略前期可复用的架构，生成一套独立的幂函数处理代码，导致项目结构违背模块化设计原则。学生则需通过对比两类函数的差异性提取共性模式，将 V3 阶段的交互框架进行抽象迁移。此阶段学生仍可通过结构化提示词辅助编程，明确要求 AI 基于 V3 自定义模块进行功能扩展，这一过程的目的在于使学生从被动接受 AI 输出转向主动控制生成逻辑，从而完成从具体到抽象的思维跃迁。

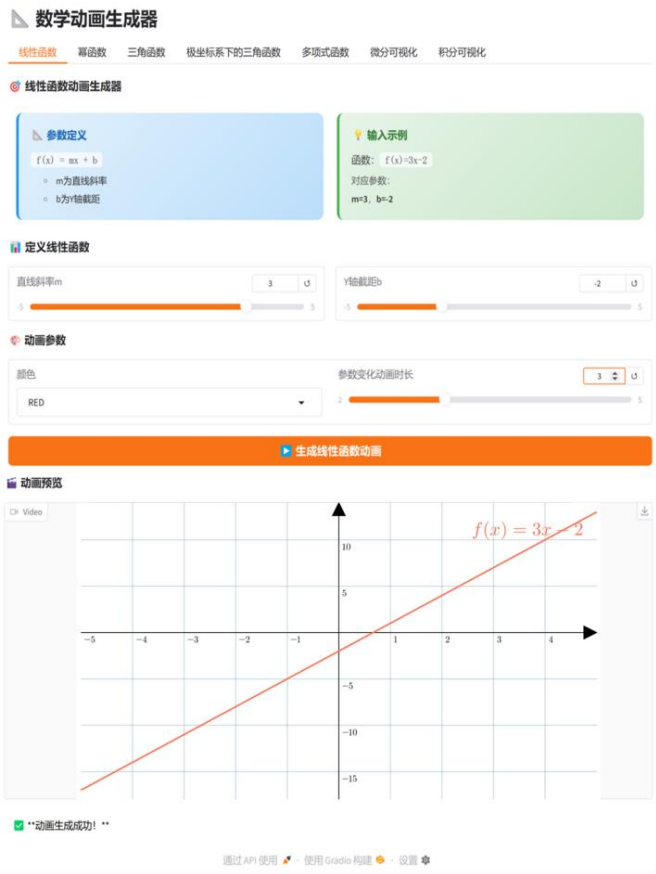


图 2 V3 交互版界面效果参考图示

5 教学成效与反思

5.1 初步成效

本次课程已面向电子商务专业 2024 级和 2025 级学生开展数学动画项目实践，要求学生以小组为单位完成系列项目并提交实践报告，并在课程结束后填写调查问卷。问卷以每学期选课的学生为对象，通过学习通进行线上填写，每学期有效问卷均在 90%以上。在此基础上，从学习动机、能力转型、AI 协同三个维度分析教学模式的初步成效。

相较于传统的文本输出型编程项目，数学动画项目更易于激活非计算机专业学生的学习兴趣，根据调研数据显示 61%的学生对动画编程项目较有兴趣，仅 9%的学生完全不感兴趣；74.3%的学生愿意将动画作品

发布至 B 站、抖音等主流视频平台，数据间接体现了可视化成果带来的成就感一定程度上高于传统代码作业，可为学生从“完成作业”向“作品创作”转变提供空间，推动其学习动机发生质变。

在能力培养方面，项目通过 AI 反制策略的层层设障，推动学生逐步建立人机协作中的主动验证意识，初步实现从单纯的复制代码到基于“读-验-改”代码实现计算思维培养的能力转型。调研统计中显示，学生在借助 AI 生成 Manim 动画代码的过程中，仅 8.6%的学生表示使用过程中未遇到问题，5%的学生认为项目编程无明显难点。这一项目门槛恰能促使学生从被动接受 AI 代码向主动修改 AI 错误的方向转变。

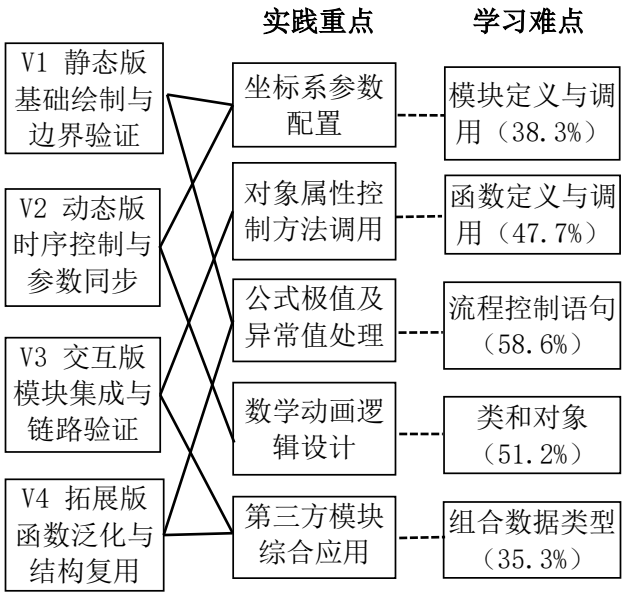
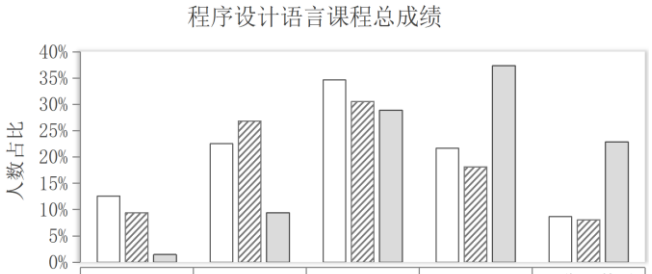


图 3 Manim-Python 技能进阶对应关系图



	100-90分	89-80分	79-70分	69-60分	59分及其以下
□25级	13%	23%	35%	22%	9%
▨24级	9%	27%	31%	18%	8%
□23级	1%	9%	29%	37%	23%

图 4 近三届学生成绩对比

如图 3 所示，Python 基础学习难点与 Manim 实践应用重点高度吻合。教学模式中正是通过在动画制作目标中创设真实应用场景，使学生在模块应用和代码调试中反向巩固 Python 基础语法，形成“场景及错误驱动学习”的良性循环。如图 4 所示，24 级和 25 级学生相比 23 级，优良率均在 30%以上，较 23 级的 10%有所提升。

5．2 教学反思

本次实践揭示了 AI 时代编程教学的深层矛盾，项目中学生使用 AI 越频繁，越能深刻体会基础能力的重要性。针对学生 AI 使用频率与风险意识的交叉分析显示，高频使用 AI 辅助项目编程的学生并未陷入盲目依赖，反而会逐步形成“高频使用-高度警觉”的理性认知。如图 5 数据显示，在“总是使用 AI”的学生中，40%“已深刻认知过渡依赖的风险”，35%“有所意识”，合计值为 75%。而“经常使用”的学生群体中，24.1%“深刻意识”，51.7%“有所意识”，合计 75.8%。由此可见高频使用者反而能更早遭遇 AI 幻觉，从而建立起对技术依赖的批判性认知。与之形成对比的是，“从未使用 AI”的学生群体中 66.7%“完全不认同”依赖风险，缺乏对 AI 风险的合理评估，这也一定程度体现了“AI 反制策略”设计的必要性。完全禁止 AI 并非培养计算思维的最优路径，让学生在可控范围内经历从 AI 失效、人工调试、反思改进的完整循环，才能建立基于实践理性的风险意识。

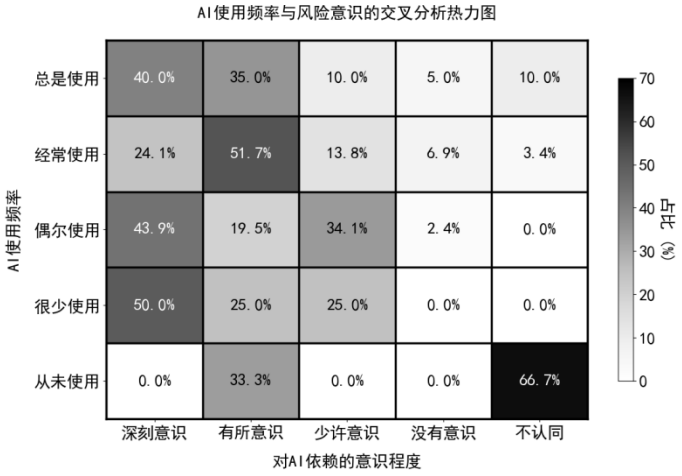


图 5 AI 使用频率与风险意识的交叉热力图

此外，动画项目虽显著提升了学生的学习兴趣，但通过调研数据和课堂反馈可以看出，学生在项目实施中仍存在较多难点，尽管这一编程困境确为 AI 反制策略的落地提供了可行性，但也提高了学生的入门门槛。在后续教学中将补充完善线上案例库，便于学生在入门阶段快速解决常规动画的设置问题，提升学习效率和实践体验。

6 结束语

本研究以计算思维培养为核心目标，开展 Python 编程实践教学改革，依托数学与编程的跨学科融合，构建“静态-动态-交互-拓展”的四层递进式项目体系，实践有效引导学生树立 AI 风险认知，强化验证、协同、

系统与抽象思维。下一步研究将着重完善线上教学资源库的建设，拓展更多跨学科融合应用场景，持续深化人机协同教学机制，为AI时代编程教育改革提供更为成熟的实践方案。

参考文献

[1] 教育部. 教育部等九部门关于加快推进教育数字化的意见 [EB/OL]. (2025-04-11)[2026-03-11]. https://www.gov.cn/zhengce/zhengceku/202504/content_7019045.htm.

[2] PAPERT S A. Mindstorms:Children,Computers,and Powerful Ideas[M]. Basic Books,1993

[3] WING J M.Computational thinking[J]. Communications of the ACM, 2006,49(3):33-35.

[4] 陈国良,李廉,董荣胜.走向计算思维2.0[J].中国大学教学,2020,(04):24-30.

[5] 郁晓华,肖敏,王美玲.计算思维培养进行时:在K-12阶段的实践方法与评价[J].远程教育杂志,2018,36(02):18-28.

[6] 姜洋,衡红军,李俊生.基于计算思维层次化认知的大学计算机教学改革实践[J].中国大学教学,2020,(11):59-63.

[7] 丁世强,王平升,赵可云,等.面向计算思维能力发展的项目式教学研究[J].现代教育技术,2020,30(09):49-55.

[8] 王亚杰,王晓岩,李飞,等.基于“两性一度”的“计算机程序设计”课程教学改革与实践[J].工业和信息化教育,2023,(06):1-5+33.

[9] 桂小林,何钦铭.AI赋能的大学计算机通识教育的体系化改革探索[J].中国大学教学,2024,(04):4-11+2.

[10] 翟雪松,张丽洁,夏亮亮,等.基于GAI的逆向工程教学思维在人机协作中的应用研究——以编程教育为例[J].电化教育研究,2024,45(09):61-68.

[11] 李秀,陆军,牛佳丽.GenAI赋能的人机双师协同教学研究——基于清华大学计算机基础课程的案例分析[J].现代教育技术,2025,35(03):34-43.

[12] 贺桂娇，郑耀东，路旭明，叶晨敏.生成式AI赋能程序设计实践教学的应用研究[J].计算机技术与教育学报,2026,14(01):67-72.

[13] 张金,宫晓利,高小鹏.面向大模型时代计算机系统能力培养的智能协同教学实验设计[J].实验室科学,2024,27(02):21-23.

[14] 厉旭杰,顾雨辰,姚持恩.集成AI大语言模型的在线编程实验平台设计与实现[J].实验技术与管理,2024,41(08):215-221.

[15] 蒋杰,杨若藜,戚瑞,等.AI数字人教师驱动的智能教学模式探索[J/OL].计算机科学,1-15[2026-02-15].https://link.cnki.net/urlid/50.1075.TP.20250925.1135.010.

[16] 何升.数字徽章驱动的高职人机协同教学机制研究与实践——以“Python程序设计”课程为例[J].广东轻工职业技术大学学报,2025,24(05):62-66+80.