

生成式 AI 驱动的 Python 教学案例探索*

张伟伟 张静 吴怀广 史雯隼

郑州轻工业大学计算机科学与技术学院, 郑州 450001

摘要 针对编程教育中标准化培养与个性化需求的结构性矛盾, 以及教学内容与产业实践脱节的现实困境, 本研究提出生成式人工智能驱动的 Python 教学案例动态生成框架。通过构建“产业知识逆向解析-教学知识体系重构”的双向映射机制, 将工业级项目解构为可重组的知识单元, 形成覆盖基础语法、工程实践与系统设计的层级化知识图谱。然后, 结合学习者认知建模技术, 建立动态适配模型, 通过人工智能技术生成匹配个体认知水平的实践案例, 并以微服务架构异常处理为例构建典型案例。该框架可有效提升学生工程问题解决能力与代码规范意识, 为破解产教融合瓶颈、建立数据驱动的自适应教学模式提供理论支撑与实践路径。

关键字 人工智能驱动, 知识图谱, 教学案例, Python 语言

Exploration of Python Teaching Cases Powered by Generative AI

Weiwei Zhang Jing Zhang Huaiguang Wu* Wenjun Shi

School of Computer Science and Technology
Zhengzhou University of Light Industry,
Zhengzhou 450001, China;

Abstract—To address the structural contradiction between standardized training and personalized learning needs in programming education, as well as the disconnect between instructional content and industrial practice, this study proposes a generative AI-driven dynamic case generation framework for Python instruction. By constructing a bidirectional mapping mechanism—"reverse parsing of industrial knowledge and reconstruction of instructional knowledge systems"—the framework decomposes industrial-grade projects into reconfigurable knowledge units, thereby forming a hierarchical knowledge graph that spans fundamental syntax, engineering practices, and system design. Leveraging learner cognitive modeling techniques, a dynamic adaptation model is established to generate practical cases tailored to individual cognitive levels using artificial intelligence. A typical case involving exception handling in a microservices architecture is developed as an example. This framework effectively enhances students' capabilities in engineering problem-solving and code standardization awareness, offering both theoretical grounding and practical pathways to bridge the gap between education and industry through a data-driven, adaptive instructional model.

Keywords: AI-driven, knowledge graph, instructional case, Python programming language

1 引言

生成式人工智能的突破性进展, 为编程教育从“规模标准化”向“精准个性化”转型提供历史性机遇[1]。通过解构工业级项目形成可重组知识单元, 依托生成式 AI 生成适配个体能力的教学案例, 既可实现“一生一策”的精准教学, 又能确保教学内容的产业实践价值。该技术路径既符合“以学习者为中心”的教育理念, 构建产教资源深度融合的创新生态, 也与教育部高等教育司提出的“数据驱动的自适应学习机制”高度契合[2], 为破解传统教育中标准化培养与个性化需

求的结构性矛盾提供了新范式。

2 现状分析

在数字化转型背景下, Python 编程能力已成为人工智能、数据科学等领域的核心素养, 但当前教育体系正面临严峻的供需失衡挑战[3]。随着云计算、AI 等技术迭代加速, 产业界对开发者的能力需求已从“基础语法掌握”转向“复杂工程问题解决”[4], 而传统教学模式仍存在两大瓶颈: 其一, 教学内容与产业实践脱节。现有教材案例多聚焦基础语法训练, 缺乏并行控制、异常监控等工程能力培养, 导致学生即便掌握语言特性, 仍难以应对工业级项目需求[5]。某调研显示, 87%的应届生需经过 6 个月以上企业培训方能胜任实际开发工作。其二, 标准化教学与个性化需求矛盾突出。统一案例库难以适配差异化认知节奏——认知水平较低者常因案例难度跃升产生挫败感, 而能力突出者则陷入重复训练的“低水平熟练陷阱”[6]。以

* **基金资助:** 本文得到河南省研究生教育改革与质量提升工程项目案例项目 (YJS2025AL37); 郑州轻工业大学研究生全英文课程 (群) 建设项目 (13109000016); 河南省高等教育教学改革研究与实践重点项目 (2024SJGLX0126); 郑州轻工业大学博士科研基金 (2022BSJJZK14) 的资助。

* * 通讯作者: 吴怀广 hgawu@126.com

上多重困境不仅造成教育资源的低效配置，更导致人才培养与产业需求的结构性错位。

生成式人工智能的技术突破，为 Python 编程教育的革新提供了前所未有的契机[7]。其核心能力与教育痛点的深度契合性，为重构教学逻辑提供新的可能[8]。本研究整合开源社区的前沿工程实践，构建与产业技术同步进化的教学资源池，通过生成式 AI 技术实现“千人千面”的案例难度调节。本研究为产教融合提供了新模式，为教师实现“精准化教学”与“及时性指导”的双重突破提供了新思路。

3 生成式 AI 驱动的 Python 教学案例生成框架

本框架遵循“产业案例解构-知识图谱构建-认知建模-动态生成”的技术路线，形成分层递进的教学案例生成体系。

3.1 基于产业案例和经典教学的双轨制知识体系构建

本研究的核心在于建立传统教学体系与工业实践的知识融合通道。首先对编程课程的知识点进行系统化梳理与评估，保留核心语法要素，剔除陈旧技术内容，同时预埋前沿技术接口，形成层次化的基础框架。在此基础上，通过对开源社区工业级项目的逆向工程解析，提取企业级工程场景中的关键知识点，例如在微服务架构中识别出断路器模式、分布式事务处理等传统教学缺失的实践性内容。

为解决教学体系与产业需求的错位问题，建立双向校准机制：将工业场景中的复杂技术拆解为梯度化教学模块，同时将传统知识点延伸至工程实践。最终形成的动态知识图谱，既完整覆盖教学大纲要求，又深度对接企业级项目的容错设计、性能优化等核心能力需求。这种“基础规范性与实践先进性”的融合路径，确保了知识体系的全面性与教学案例的产业价值。

3.2 学习者认知建模与动态难度适配

基于学习过程的多模态数据（代码调试路径、知识图谱访问轨迹、协作讨论记录等），通过特征融合技术识别学习者的能力发展阶段：在基础认知阶段聚焦语法规则的内化，在工程思维形成期强化异常处理等系统性能能力，在创新迁移阶段侧重复杂场景的问题拆解。同时，结合行为模式分析划分学习风格类型——为实验型学习者设计渐进式代码填空任务，比如引导其通过试错理解断路器模式的工作机理；为理论型学习者构建含架构原理图示的文案例，比如帮助其建立异常传播链的全局认知。动态适配引擎通过模糊

推理机制，在保留工业场景完整性的前提下调节案例复杂度：比如对处于认知过渡期的学生，保留订单服务核心业务逻辑但简化分布式锁异常场景；对高阶能力者则植入服务雪崩防护挑战，推动其向产业级问题求解能力跃迁。

3.3 教学案例动态生成

基于“约束引导-动态调节-风格适配”三位一体的生成范式，构建教学案例智能生成引擎。生成过程遵循三项核心原则：

（1）产业真实性约束：严格保留工业场景的技术约束与业务逻辑完整性，通过参数化封装实现核心要素的可配置性。

（2）认知发展适配：依据学习者能力坐标动态调节案例复杂度，采用“核心功能保留-辅助功能简化-高阶挑战植入”的渐进式生成逻辑。

（3）多模态表达适配：结合学习风格特征生成差异化案例形态，包括代码脚手架（实验型）、原理示意图解（理论型）、协作任务书（社交型）。生成引擎通过动态权重分配机制平衡知识覆盖密度与认知负荷强度，确保案例既满足教学目标要求，又符合学习者的最近发展区规律。

4 案例——以微服务架构的异常处理能力培养为例

选取某开源社区高星项目微服务架构案例中的订单服务模块，聚焦其异常处理机制设计，实现知识点提取和难度量化，通过生成式 AI 生成动态案例。

4.1 知识点匹配

知识点匹配流程采用双向交互机制，包含两个核心环节：首先基于工业级应用案例实施自上而下的解析，通过代码逆向工程抽取异常处理模式与关键知识点；继而依托异常处理知识体系开展自下而上的梳理，通过知识缺口分析进行系统性补全，最终构建出层次化、结构化的异常处理知识图谱。该流程通过双向知识提取机制，既保证了知识提取的实践落地性，又确保了知识体系的完整性和系统性。

（2）代码解析与知识点提取

针对微服务架构的订单处理模块，识别出其异常处理设计模式包含断路器模式（Hystrix）和重试策略（Tenacity），然后使用 AST 解析器提取关键异常处理知识点。例如解析其中部分代码片段如下，从中提取的异常知识点包括基础语法 try-except 捕获异常，异常的抛出 raise，web 框架内置异常 HTTPException 等知识点。

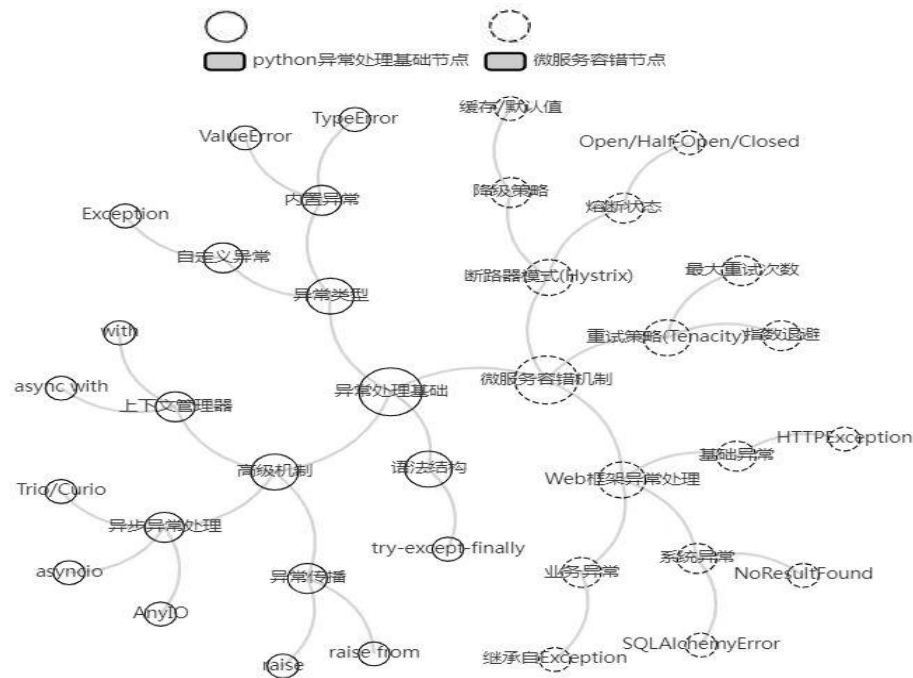


图 1 异常处理章节知识点生成的知识图谱

```
# 工业代码片段解析
try:
    order =
db.session.query(Order).filter_by(id=order
_id).one()
except NoResultFound:
    raise HTTPException(404, "Order not
found")
# 知识点：资源不存在异常处理
except SQLAlchemyError as e:
    logger.error(f"Database error:
{str(e)}")
    raise HTTPException(503, "Service
unavailable") # 知识点：基础设施异常处理
```

(2) 知识补充与图谱构建

基于案例中提取的知识点，结合异常处理知识体系，补全缺失的知识点构建完整的知识图谱。构建异常处理模块知识图谱包含两步，首先是节点定义，包含 Python 异常处理基础节点和微服务容错机制节点两类父级节点，每个节点又包含多个子知识点，然后是边的定义，具有上下级关系的节点之间有一条边，于是知识图谱构建如图 1 所示，其中虚线圆圈节点是案例抽取知识点，实线圆圈节点是补充知识点。

(3) 难度量化

为了便于不同等级的案例生成，初始为每个知识点标注复杂度分值，按照难度等级从低到高分五个等级，分值是1-5分。异常处理的知识点难度量化如表1

所示。

4.2 知识图谱驱动的动态难度适配

(1) 学生画像构建

数据采集：通过学生的学习历史、作业完成情况、测试成绩、互动反馈等，构建学生的学习档案。

能力评估：基于知识图谱中的知识点和难度分值，评估学生对每个知识点的掌握程度。

学习风格分析：根据学生的学习速度、偏好（如偏好代码实践、理论讲解或视频学习）等，分析其学习风格。

(2) 知识图谱与难度映射

知识点关联：基于知识图谱，将知识点按照依赖关系（如先学习基础异常捕获，再学习断路器模式）和难度分值进行组织。

动态难度划分：将知识点划分为多个难度阶梯（如基础、中级、高级），并根据学生的掌握情况动态调整难度。

(3) AI互动式案例生成

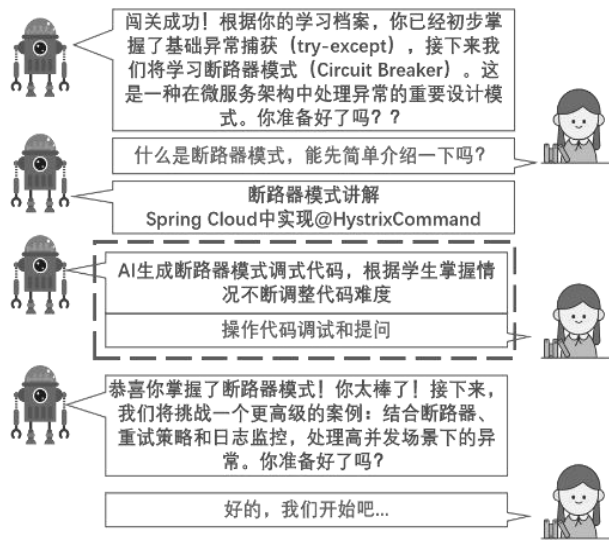
案例生成逻辑：如果学生对某个知识点未掌握，生成基础案例，侧重于该知识点的概念讲解和简单代码实践。例如，使用try-except处理简单的资源不存在异常。

如果学生初步掌握，生成中级案例，侧重于知识点的应用场景和代码优化。例如，在微服务架构中实现断路器模式。

表 1 知识点难度量化

知识点	难度等级
基础异常捕获 (try-except)	1 分
捕获特定异常 (内置异常)	2 分
捕获所有异常 (Exception)	3 分
异常传播 (主动抛出、异常链)	3 分
资源清理 (上下文管理器 'with' 语句)	4 分
Web 框架异常处理	4 分
异步异常处理 (async/await 异常处理逻辑)	5 分
断路器实现	5 分
重试策略实现	5 分

如果学生熟练掌握，生成高级案例，侧重于复杂场景下的综合应用（如多个知识点的组合使用）。例



如，结合断路器、重试策略和日志监控，处理高并发场景下的异常。

(4) 反馈与动态调整

实时反馈：通过学生的案例完成情况（如代码正确性、完成时间、错误类型等），实时评估其掌握程度。

动态调整：根据反馈结果，动态调整后续案例的难度和内容。例如：

如果学生完成基础案例后仍未掌握，生成更简单的基础案例或提供额外讲解。

如果学生完成中级案例后表现优异，生成高级案例或引入新的知识点。

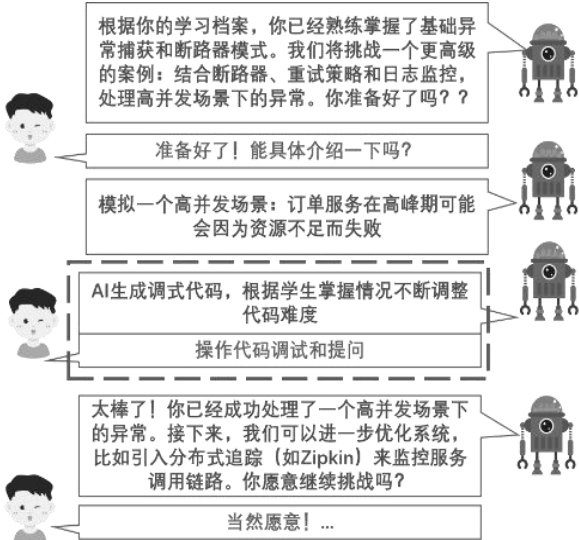


图 2 学生A和B的学习场景示例

学习档案更新：将学生的案例完成情况和反馈结果更新到其学习档案中，为后续案例生成提供依据。图 2 展示了针对学生 A 和 B 的不同学习路线。

在个性化学习路径的实现中，系统依据学生能力画像动态生成梯度化案例：对于掌握基础异常处理但欠缺工程经验的学生（如学生 A），生成融合断路器模式与重试策略的中阶案例，通过模拟订单服务并发异常场景引导其理解容错机制设计；当检测到学生对核心概念存在认知偏差时，自动追加包含日志监控与事务补偿的拓展训练模块。对于能力突出者（如学生 B），则直接推送高并发场景下的全链路异常处理挑战，并在其完成高阶任务后无缝衔接分布式追踪等新技术点，形成“掌握即进阶”的持续成长路径。该机制通过精准识别学生能力边界，实现“巩固薄弱点-强化优势项-拓展新领域”的个性化培养闭环。

4.3 教学成效分析

郑州轻工业大学自从2019年开始在校内课程平台上开展Python教学活动，配套有大量的视频、习题、论坛等相关学习互动内容。自从2023年初上线生成时AI驱动

的python教学案例，极大促进了教学效果。通过课程信息系统的数据，从2019 年至2024年底，该课程的评教成绩保持在90分以上，学生总体满意度较高，同时课程目标达成度也保持了较高的水平，具体数据如表2所示。

通过课程平台信息后台数据显示，不同学生的进度对应了不同的学习路径，图3显示自从引入生成式AI驱动的python教学案例，大大提升了学生的学习兴趣，学生在课程各个环节的互动参与率都有较大的提高。任课老师针对本课程设计的教学案例进行了问卷调查，并时获得反馈信息，如图4显示，97%的学生认为AI的引入在教学环节起到了积极的作用，尤其是对代码能力和学习效率有极大的提升。

表 2 学生平均成绩、评教成绩和课程目标达成度

学期	平均成绩	评教成绩	课程目标达成度
2019 年秋	70.1	90.3	0.792
2020 年春	73.2	92.1	0.742
2020 年秋	73.5	92.2	0.751
2021 年春	72.7	91.6	0.762
2021 年秋	74.5	93.1	0.812
2022 年春	74.2	92.1	0.842
2022 年秋	73.8	90.8	0.837
2023 年春	75.6	93.2	0.865
2023 年秋	76.2	93.4	0.873
2024 年春	77.1	92.7	0.847
2024 年秋	78.2	93.8	0.855

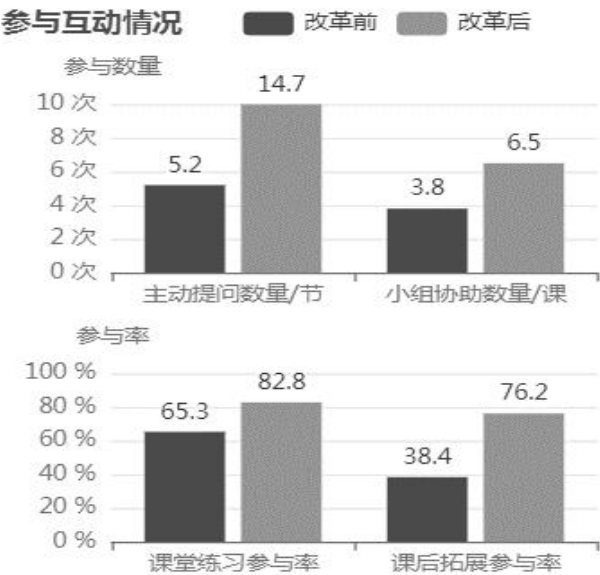


图 3 互动情况对比

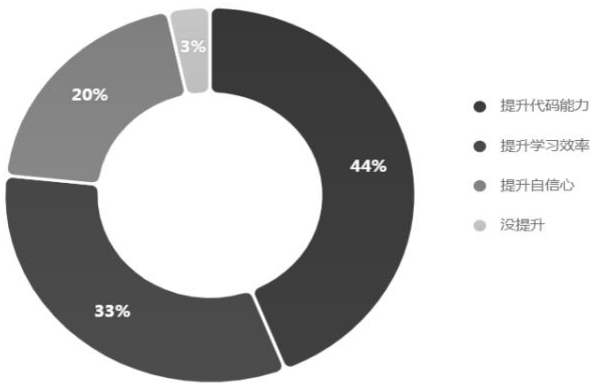


图 4 生成式AI的引入对学习的作用结果汇总

此外，依托本案例培养的工程实践能力，课程组指导学生参加多项学科竞赛表现优异：在蓝桥杯全国软件和信息技术专业人才大赛（Python程序设计组）中获国家级三等奖2项、河南省赛一、二、三等奖多项；在中国高校大数据挑战赛、MathorCup数据挑战赛以及计算机能力挑战赛等赛事中，亦斩获优异成绩（包括一等奖2项，二等奖2项，三等奖多项）。

5 结束语

本研究针对编程教育中“标准化培养与个性化需求”“教学滞后与产业迭代”的双重矛盾，构建了生成式AI驱动的Python教学案例动态生成框架，为教育数字化转型提供了创新性解决方案。通过动态难度适配，确保学生在每个阶段都能获得适合其水平的挑战，逐步提升能力。通过知识图谱的全面覆盖，确保学生能够系统性掌握Python异常处理的核心知识。

根据学生的学习档案和反馈，提供个性化的学习路径，最大化学习效果。这种动态生成的案例比固定教材更能激发学习动力——学生更愿意尝试解决有实际意义的工程问题，而非机械完成语法练习题。未来我们将优化案例的真实性，比如引入更多企业开源项目，同时探索如何让教师更方便地参与案例设计，让AI生成的案例既能满足个性化需求，又不失教学的系统性和规范性。

参考文献

[1] 穆肃,陈孝然,周德青.生成式人工智能赋能教学设计分析:需求、方法和发展[J].开放教育研究,2025,31(01): 61-72.

[2] 王繁,刘永强,周天华.人工智能引领高等教育数字化创新发展[J].中国高等教育,2024,(Z1):9-12.

[3] 沈建林.生成式技术在 Python 课程教学中智能化应用策略研究[J].中国培训,2025,(01):83-86.

[4] 郭曦,王建勇.生成式人工智能在 Python 教学中的作用与思考[J],计算机技术与教育学报,2024,(12) 02:31-36.

[5] 刘敏,王耀南.工业场景下的人工智能教学案例设计,计算机技术与教育学报[J],2023,11(3):99-104.

[6] 刘明剑,张思佳,李奇蔚,葛泰亨.依托“Python 程序设计”课程的学生创新能力培养模式研究与实践[J],计算机技术与教育学报,2024,12(4):40-44.

[7] 王明瑞.生成式AI在自适应程序设计教学系统中的应用探析[J].信息与电脑(理论版),2024,36(20):191-193.

[8] 裴榕.生成式人工智能赋能教育教学:变革影响、风险挑战与实践路径[J].当代教育论坛,1-10,2024-12-9.