

融合DeepSeek的《高级语言程序设计》 实践教学创新探索

冯丙文¹, 张晓倩^{2*}, 张继连¹, 刘志全¹, 耿光刚¹

¹暨南大学网络空间安全学院, 广东 广州

²暨南大学信息科学技术学院, 广东 广州

收稿日期: 2025年8月25日; 录用日期: 2025年9月22日; 发布日期: 2025年9月29日

摘要

随着人工智能技术的快速发展, DeepSeek大模型凭借其强大的推理能力与多模态交互功能, 为编程教育提供了新的解决方案。本文以《高级语言程序设计》实验教学为例, 探讨DeepSeek如何通过代码纠错、个性化学习支持、智能题库生成等核心功能, 重塑实验教学模式, 提升学生编程能力与教学效率。

关键词

高级语言程序设计, DeepSeek, 编程教育

Innovative Exploration of Practical Teaching in “Advanced Programming Languages” through the Integration of DeepSeek

Bingwen Feng¹, Xiaoqian Zhang^{2*}, Jilian Zhang¹, Zhiquan Liu¹, Guanggang Geng¹

¹College of Cyberspace Security, Jinan University, Guangzhou Guangdong

²College of Information Science and Technology, Jinan University, Guangzhou Guangdong

Received: August 25, 2025; accepted: September 22, 2025; published: September 29, 2025

Abstract

With the rapid development of artificial intelligence, the DeepSeek large-scale model, leveraging its powerful reasoning capabilities and multimodal interaction functions, offers new solutions for programming education. Using the laboratory teaching of “High-Level Language Programming” as an

*通讯作者。

文章引用: 冯丙文, 张晓倩, 张继连, 刘志全, 耿光刚. 融合 DeepSeek 的《高级语言程序设计》实践教学创新探索[J]. 教育进展, 2025, 15(10): 208-219. DOI: 10.12677/ae.2025.15101822

example, this paper explores how DeepSeek, through core functions such as code error correction, personalized learning support, and intelligent question-bank generation, can reshape laboratory teaching models and enhance students' programming proficiency and instructional efficiency.

Keywords

High-Level Language Programming, DeepSeek, Programming Education

Copyright © 2025 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

作为计算机类专业的核心基础课程,《高级语言程序设计》构建了网络空间安全、软件工程等专业的底层能力框架。该课程通过 C/C++、Python 等典型语言载体,系统传授结构化与面向对象程序设计范式,培养算法思维、模块化开发能力及工程伦理意识,在信息技术人才培养中具有良好的效果。随着大数据、人工智能、5G 等新兴技术的快速发展与普及应用,人工智能对 IT 行业产生深远影响。人工智能辅助的编程自动化已成为技术趋势,这种能力不但减少了开发人员的代码编写负担,还为非专业开发者提供了通过自然语言来生成代码的便利性,使得编程门槛进一步降低。如何在技术背景下保障《高级语言程序设计》课程技术与教学内容的先进性成为了一个问题。而基于人工智能的实践教学也已是教育发展的必然趋势。当前技术生态正经历三重变革:首先, GitHub Copilot 等 AI 代码生成工具已广泛普及,其次,工业界对算法优化与系统级编程的复合型人才需求增长,美国劳工统计局数据显示,2025~2030 年“初级软件工程师”岗位需求将下降 15%,“AI 架构师”需求增长 41%,全栈工程师需求增长 28%。正如 LinkedIn《2024 全球技术人才趋势报告》中指出,“未来五年,不会使用 AI 工具的开发者的竞争力将像不会使用 IDE 的工程师一样失去竞争力”。反观传统《高级语言程序设计》实践教学存在显著滞后,实验项目仍以控制台应用为主(占比 85%),缺乏智能算法集成;代码评估依赖静态语法检查,未建立动态质量评价体系;教学反馈周期平均长达 72 小时,难以实现精准指导。这限制了《高级语言程序设计》的授课效果,使得这门课难以在新技术发展趋势面前发挥其应有的作用。本文针对《高级语言程序设计》实践教学面临的问题,探讨 DeepSeek 如何通过代码纠错、个性化学习支持、智能题库生成等核心功能,重塑实验教学模式,提升学生编程能力与教学效率。

2. 大语言模型辅助编程类课程发展现状

2.1. 大语言模型辅助编程技术发展现状

大语言模型(LLM)辅助编程正快速超越传统“代码补全”的范畴,向更专业、更具情境感知的软件工程任务持续深化。

在核心软件工程任务上, LLM 的能力与应用范式不断演进。程序修复研究已从简单的缺口填充转向更契合 LLM 训练目标的“程序精炼”范式,即通过重写整个函数以提升修复效果[1]。同时, Cref 等对话式框架开始模拟人类专家的调试流程,融合导师式指导等多源信息以实现协作修复[2]。在自动化测试方面, ChatUniTest 等体系引入“自适应焦点上下文”与“生成-验证-修复”的闭环机制,能够产出质量更高且可自我纠正的测试用例[3]。

在人机交互范式上,研究致力于将 LLM 更无缝地嵌入开发者工作流。一方面,工具由被动响应转向

“主动式”助手，可前瞻性地提供代码优化与测试建议，如 CodingGenie [4]。另一方面，为降低认知负担，GILT 等工具探索“免提示”交互，将常见的代码理解需求固化为 IDE 内快捷操作，显著提升易用性与任务完成率[5]。

在教育应用场景中，关键在于平衡辅助效果与学习目标。为避免削弱学生的独立思考，CodeAid 等工具引入“教学护栏”(pedagogical guardrails)理念，通过提供伪代码、带注释的错误反馈与概念解释等方式引导学习，而非直接给出完整答案，从而确保技术用于“辅助学习”而非“替代学习”[6]。

总体来看，LLM 正由“代码生成器”演变为能够深度参与软件开发生命周期的智能伙伴。当下技术演进的核心趋势体现在任务专业化、交互智能化与应用情境化。面向未来，研究有望在更深层的任务集成、更高效的人机协作模式以及教育等关键领域的负责任应用上持续推进。

2.2. 大语言模型在计算机类专业课程的教学应用

随着以大语言模型(LLM)为代表的生成式人工智能(Generative AI, GenAI)迅速发展，其在计算机科学教育中的应用已成为学术与教学实践的前沿议题。当前焦点已由“是否使用”转向“如何有效且负责任地整合”，力图在提升教学质量与学习体验的同时，正视其潜在挑战。国内外研究主要围绕教学框架、学习支持、课程改革与实践难点等方面展开。

在教学框架与系统性整合方面，研究者正构建理论与实践并重的体系，以指导 LLM 在教学中的深度融合。一类研究着眼于教学的底层逻辑，提出创新理论模型。例如，李永智等[7]借鉴大模型技术思路提出“教学思维链”，旨在显化推理过程，将优秀教师内隐的教学逻辑与决策路径结构化表征，推动通用大模型向教育专用模型演进，以缓解“黑箱”推理与“知识幻觉”等问题。另一类研究强调可操作的教学干预。Dickey 等[8]设计并评估了“AI-Lab”这一结构化干预，通过脚手架引导学生以审慎、高效的方式使用 GenAI。结果显示，该干预显著提升了学生在概念理解与调试等情境中使用 GenAI 的舒适度与开放度，并促使其从“朴素探索”转向“战略性应用”，而非单纯提高使用频率[9]。

在强化学习支持与自动化反馈方面，LLM 广泛用于提供个性化支持与即时反馈。一个重要方向是与自动化评估工具(Automatic Assessment Tools, AAT)集成。Pereira Cipriano 等[10]将 GPT 与 AAT 融合：当 AAT 发现学生代码错误时，学生可在 IDE 中直接请求 LLM 给出针对性的解释、调试建议或代码示例，既减少上下文切换，又弥补传统 AAT 反馈的不足。此外，LLM 在支持不同编程基础学生方面亦展现潜力。董珊珊等[11]在面向生物类专业的 Python 项目式教学中，利用 GenAI 帮助学生理解理论、拓展解题思路，支持编程基础薄弱的学生跟上进度。相关实证亦表明，基于 LLM 的编程学习环境能够显著提升学习动机与编程能力[12]。

在课程内容与实践教学改革方面，LLM 的出现正深刻影响传统课程内容与实验设计。为应对其“可直接生成代码”的新现实，改革更强调系统性思维与高阶能力培养。穆玲玲等[13]在汇编语言实验教学改革中提出面向“系统观”的任务设计原则，通过延长任务逻辑链、设置综合性与开放性实验，促使学生从软硬件协同角度思考问题，避免对大模型的过度依赖。类似地，王佳等[14]在操作系统课程的实践教学采用“LLM+ 知识图谱”的方案，依据学生就业兴趣与职业规划构建个性化实践路径。上述改革表明，未来实践教学将更关注学生借助 LLM 解决复杂工程问题的综合能力，而非单一的代码编写技能。

尽管 LLM 带来诸多机遇，其引发的挑战与学生行为问题同样不容忽视，尤其是学术诚信与过度依赖。研究显示，在缺乏监督的情境下，学生易依赖 LLM 进行代码生成，进而弱化独立问题求解与计算思维等核心能力的培养[15]。通过有效的教学干预，可引导学生形成更成熟、批判性的使用策略。因此，当前教学界的核心任务在于设计既能发挥 LLM 优势、又能规避其负面影响的教學模式与评价体系，确保学生在人机协同的新范式下实现真正的学习与成长。

3. 《高级语言程序设计》实验教学的痛点与 DeepSeek 的适配性

3.1. 实验教学面临的主要挑战

《高级语言程序设计》作为计算机科学的基础课程，其实验教学环节对培养学生的编程能力至关重要。然而，当前的实验教学模式存在几个突出问题，严重影响了教学效果。

学生代码错误定位困难：初学者在编写 C 语言程序时，常常遇到各种编译错误和运行时错误。面对编译器给出的错误提示，学生往往难以理解其真正含义，更难以找到问题的根源。特别是涉及指针操作、内存管理等复杂概念时，调试过程变得异常困难，这极大地打击了学生的学习积极性。

学生问题的开放性和多样性：在实验过程中，学生提出的问题千差万别，从基础语法到算法设计，从环境配置到代码优化，问题类型繁多且具有很强的个性化特征。传统的集中答疑模式难以满足所有学生的即时需求。

教师批改反馈滞后：由于学生人数众多，教师批改作业需要大量时间，导致反馈往往不够及时。当学生收到批改结果时，可能已经忘记了当时的编程思路，错过了最佳的学习时机。此外，人工批改往往只能给出简单的对错判断，缺乏深入的代码质量分析。

个性化学习资源匮乏：学生的编程基础和学习进度存在差异，但传统教学提供的是统一的学习资源。基础好的学生需要更有挑战性的练习，基础薄弱的学生需要更多基础训练，而现有资源难以满足这种差异化需求。

3.2. DeepSeek 技术特性与教学痛点的精准对接

DeepSeek 作为先进的 AI 辅助工具，其技术特性与上述教学痛点形成了良好的对接，为解决这些问题提供了新的可能。

(1) 自然语言处理与代码分析能力

DeepSeek 能够深入解析学生代码的逻辑结构，不仅可以精准识别语法错误，还能发现潜在的逻辑漏洞。更重要的是，它能够用自然语言解释错误原因，将晦涩的技术术语转化为学生容易理解的表述。例如，当学生出现数组越界错误时，DeepSeek 不仅指出错误位置，还会解释什么是数组越界、为什么会发生、如何避免，并提供修正建议。这种实时的、易懂的反馈极大地提升了学生的调试效率。

(2) 个性化学习路径设计

基于对学生实验数据的分析，DeepSeek 能够识别每个学生的知识薄弱点和学习特征。系统可以据此生成定制化的练习题，为不同水平的学生提供适合的学习内容。对于基础薄弱的学生，系统会推荐更多基础练习并提供详细指导；对于进阶学生，则提供更具挑战性的算法问题。这种因材施教的方式确保每个学生都能在适合自己的节奏下学习进步。

(3) 多模态交互支持

DeepSeek 支持多种形式的知识表达和交互方式。它可以将抽象的程序逻辑转化为直观的流程图，帮助学生理解程序的执行过程；可以用思维导图展示知识点之间的关联，帮助学生构建完整的知识体系；还可以通过代码示例和动画演示，让复杂的概念变得生动易懂。这种多模态的教学方式适应了不同学生的学习风格，提高了知识传递的效率。

4. DeepSeek 在 C 语言实验教学中的核心应用场景

4.1. 代码纠错与实时诊疗

(1) 智能错误识别与定位

DeepSeek 在代码纠错方面展现出强大的能力，能够识别 C 语言编程中的各类典型错误。系统不仅可以发现表层的语法错误，更能深入检测逻辑缺陷和潜在隐患。

对于常见的指针误用问题，DeepSeek 会精准定位错误代码行，并通过清晰的说明帮助学生理解问题本质。比如当出现野指针访问时，系统会解释：“指针 p 未初始化就被使用，这会导致程序访问未知内存地址”，并提供正确的初始化方式。对于内存泄漏问题，系统能够追踪动态内存分配与释放的匹配情况，提醒学生在适当位置添加 free() 函数。

(2) 可视化错误解析

当学生编写的循环语句出现逻辑错误时，DeepSeek 采用图形化方式展示问题。系统会高亮标注错误代码段，同时生成两个对比流程图：一个展示当前错误逻辑的执行路径，另一个展示正确逻辑的执行流程。这种直观的对比让学生快速理解循环条件设置不当或循环体逻辑错误的影响。

(3) 交互式调试指导

借鉴编程猫与 DeepSeek 的成功集成经验，系统通过渐进式的问答交互引导学生自主发现和解决问题。当检测到错误时，AI 助手不会直接给出答案，而是通过提问启发思考：“你的循环终止条件是什么？”“这个变量在使用前是否已经赋值？”这种苏格拉底式的教学方法有助于培养学生的独立思考和调试能力。

4.2. 实验案例设计与教学资源生成

DeepSeek 极大地简化了教师的备课工作。教师只需用自然语言描述实验需求，系统就能快速生成完整的实验方案。例如，当教师输入“设计一个 C 语言链表操作的实验案例”时，DeepSeek 会生成：

1. 完整的实验目标和要求说明
2. 包含详细注释的代码框架，涵盖链表的创建、插入、删除、遍历等基本操作
3. 分步骤的实验指导，帮助学生循序渐进地完成实验
4. 测试用例和预期输出，便于学生验证程序正确性

DeepSeek 支持创新性的跨学科实验设计，在网络空间安全专业的教学中，系统能够结合硬件编程生成综合性实验案例。例如，设计基于 Arduino 的简单加密通信实验，让学生在编写 C 语言程序控制硬件的同时，理解基础的密码学原理。这类实验将抽象的编程知识与具体的安全应用相结合，提升学生的工程实践能力和专业认同感。

4.3. 个性化学习与分层教学

DeepSeek 通过持续跟踪学生的编程表现，构建个性化的学习档案。系统分析每个学生在不同知识点上的掌握情况，识别其学习模式和常见错误类型。基于这些数据，自动将学生划分为不同的学习层次：基础巩固组：需要加强基本语法和简单算法训练的学生；稳步提升组：已掌握基础，需要深化理解和拓展应用的学生；进阶挑战组：基础扎实，可以尝试复杂算法和系统设计的生。

(1) 差异化资源推送

针对不同组别，DeepSeek 推送相应难度的学习资源。基础巩固组获得更多带有详细解释的示例代码和基础练习；稳步提升组接收结合实际应用的编程任务；进阶挑战组则面对算法优化、性能调优等高阶挑战。

(2) 动态难度调节

系统根据学生的实时表现动态调整任务难度。对于初学者，实验任务可能只要求实现基本的输入输出功能；随着能力提升，逐步增加数据验证、异常处理等要求；对于高水平学生，则引入时间复杂度优

化、空间效率提升等进阶目标。这种渐进式的难度设计确保每个学生都在适合的挑战水平上学习。

4.4. 自动化评估与教学管理

(1) 作业批改与评分

DeepSeek 的自动评分系统可以从多维度代码评估学生代码：功能完整性，检查程序是否正确实现了所有要求的功能；代码规范性，评估变量命名、缩进格式、注释完整度等编码规范，运行效率，分析算法的时间和空间复杂度；错误处理，检查是否有适当的输入验证和异常处理机制。这种全方位的评价让学生不仅关注程序能否运行，更注重代码质量的全面提升。

(2) 教学数据可视化

DeepSeek 为教师提供直观的教学分析工具，通过收集到的数据给出代码生成雷达图展示班级整体在各知识点上的掌握程度，绘制热力图显示常见错误的分布情况，帮助教师快速识别教学重点和难点。基于这些可视化数据，教师可以及时调整教学策略，在学生普遍薄弱的环节增加讲解和练习，实现精准教学。

5. DeepSeek 启发的课程评价机制改革

5.1. 融合 DeepSeek 元素的评价机制

通过合理设计教学目标，严格监控教学流程，并周期性地对各核心教学环节进行质量评估，可以确保教学任务的圆满完成。遵循成果导向教育(OBE)的原则，工科类人才培养毕业要求可被精细划分为工程知识、问题分析、设计/开发解决方案、研究能力、现代工具运用、工程与社会责任等十二个维度，以此全面保障人才培养目标的顺利达成。在这其中，高级语言程序设计课程为工程知识、设计/开发解决方案两大关键能力的培养提供支撑。然而，随着以 DeepSeek 为代表的人工智能相关技术在计算机设计领域的迅猛进步，课程要求也随之提升，面临着新的挑战。为应对这一变化，我们重新审视并优化课程对人才培养目标的支撑结构，设计融合人工智能元素的计算机组成原理教学目标。这一创新举措旨在更好地适应技术发展趋势，培养出既具备扎实理论基础又掌握前沿技术的网络空间安全人才。课程目标定义可以如表 1 所示。

Table 1. Formulation of instructional objectives incorporating DeepSeek components

表 1. 融合 DeepSeek 元素的教学目标设计

序号	教学目标	DeepSeek 技术要求	对应毕业要求
1	掌握程序设计思想、工程开发基础和专业基础知识，具体为：掌握 C 语言程序的基本构成，C 语言的基本概念，C 语言的编译与运行方法，基本算法的设计与实现等内容。	能够通过 DeepSeek 辅助编程，高效的完成特定功能的模块设计。	掌握相关的数学、自然科学、工程基础和专业基础知识，用于网络空间安全领域复杂工程问题的恰当表述。
2	能够以 C 语言为工具，对网络空间安全的具体问题，得到合适的解决方案，通过团队合作，设计并实现软件模块。具体为：掌握“自顶向下”的结构化程序设计思路，掌握模块化的程序设计方法，初步具备团队合作开发意识。	能够利用 DeepSeek 建模和仿真，对实际编程问题中的复杂问题进行精确建模和描述。 能够通过 DeepSeek 辅助分析，探索多种解决方案，评估其可行性和有效性。 能够运用 DeepSeek 评估和验证，分析各影响因素及其关联性，验证解决方案的合理性。	能够根据网络空间安全系统的特定需求，完成软件模块的设计。

根据教学目标,进而设计课程的考核方式和成绩评定标准。为保证考核的全面性,本门课程的考核由平时作业、课内实验、课外实践、期末考试四部分组成。并设立多样化的课程质量评价体系,包括客观评价和主观评价两部分。客观评价包括课程目标达成度情况合理性审核表、成绩分析表,主观评价包括学生评教、教师自评、课程目标达成情况评价合理性审核、课程质量调查问卷数据汇总及分析,以及毕业生座谈等。

基于上述评价结果,课程建立了持续改进机制。每次完成教学过程后,基于评价结果分析存在的问题,并提出具体改进措施,指导下一轮授课。例如,从《课程质量调查问卷数据汇总及分析表》中可以看到,学生们对于现有实验课的开设内容有丰富的要求,这也正说明我们亟需对实验课进行改革,使其符合网安人才的培养,匹配学生的现有知识体系,能够激发大家学习的热情。此外,还定期评估学生在满足各项毕业要求方面的能力达成状况,并及时反馈课程教学中存在的问题。

5.2. DeepSeek 运用能力作为实践能力培养目标

在强调 DeepSeek 技术掌握的同时,提升学生的 DeepSeek 运用能力同样重要。以大语言模型为代表的人工智能使知识和浅层经验的重要性日益降低,转而强调人工智能辅助的设计创新能力,因此实践能力培养目标必然需要对应调整。结合教育心理学家本杰明·布鲁姆的六级认知层次思维模型(记忆、理解、应用、分析、评价、创新),我们把 DeepSeek 运用能力界定为分析、评价和创新 3 个高阶思维能力层次,并依次设计实验评测方法,具体如表 2 所示。

Table 2. Development of training objectives aimed at evaluating competence in applying DeepSeek
表 2. DeepSeek 运用能力培养目标设计

学习能力	对应人工智能运用能力	实验评测方法
分析	数据解读	能够准确解读 DeepSeek 处理的数据,识别数据中的模式、趋势和异常,为决策提供数据支持。
	算法理解	深入理解不同 DeepSeek 算法的原理、适用场景、优缺点及性能表现,能够选择合适的算法解决问题。
	系统诊断	利用 DeepSeek 进行故障排查和性能分析,识别系统瓶颈和潜在问题,提出改进建议。
	场景适应	分析特定应用场景下 DeepSeek 结果的适用性、可行性及潜在风险,为技术选型和应用部署提供依据。
评价	评测	通过量化指标(如文本分类、情感分析、机器翻译等等)评估 DeepSeek 的性能表现,判断其是否满足预期目标。
	伦理	评估 DeepSeek 应用对社会、伦理、隐私等方面的影响,甄别技术应用的合法性和道德性。
创新	集成	将 DeepSeek 等人工智能辅助工具与其他信息检索能力与信息储备能力进行集成,提升整体任务完成能力。
	优化	能够通过 DeepSeek 等人工智能辅助工具的交互反馈,改造现有系统,提升效能。
	主观能动性	不会过度依赖 DeepSeek 辅助,保持自我反思与批判性思考,始终作为具备高阶思维能力的学习主体。

5.3. 增加“AI 协作任务的考核维度”

在课程评价机制中,本研究提出将“AI 协作任务”作为独立的考核维度。例如,学生是否能够识别

AI 生成代码中的逻辑漏洞，是否能够提出优化方案并实施改进，是否能清晰记录其分析与修改过程。这些维度不仅考查学生的编程技能，还进一步验证其批判性思维与反思能力。通过将 AI 协作纳入评价框架，课程考核能够更全面地反映学生在新型学习环境中的综合能力，具体如表 3 所示。

Table 3. The design of assessment dimensions for AI collaborative tasks

表 3. AI 协作任务的考核维度设计

AI 协作任务	具体考核维度	评价方式	权重比例
使用 AI 生成初步代码	正确调用 AI 工具并生成基本可运行代码	作业/实验	10%
对 AI 生成代码的评估	能否识别逻辑漏洞、潜在错误或冗余	学习日志 + 答辩	20%
对 AI 生成代码的优化	是否能提出优化方案并实施改进	项目报告	20%
批判性反思	总结 AI 的优势与局限并提出改进建议	课程总结	10%

6. 实践案例与效果验证

在暨南大学的《高级程序语言设计实验课》教学实践中，DeepSeek 被引入，扮演着每位学生“随身智能编程助手”的角色。这一举措旨在变革传统的实验课模式，将 AI 的即时性、交互性与知识广度融入教学环节。在实验课过程中，学生能够利用 DeepSeek 即时解决从基础语法到复杂算法实现的各类代码问题。

具体而言，当学生遇到编程障碍时，无论是简单的语法错误、函数用法混淆，还是需要实现如“求平均数”这类基础算法时，他们都可以通过与 DeepSeek 进行自然语言对话来寻求帮助。AI 不仅能快速生成结构清晰、语法规范的代码建议，更重要的是，它还能对代码的逻辑、不同实现方式的优劣进行解释，甚至提供优化方案。这种即时的、一对一的反馈循环，极大地缩短了学生在遇到问题后“卡壳”的时间，有效克服了传统实验课中因等待教师或助教指导而造成的学习中断。具体实践如图 1 所示。

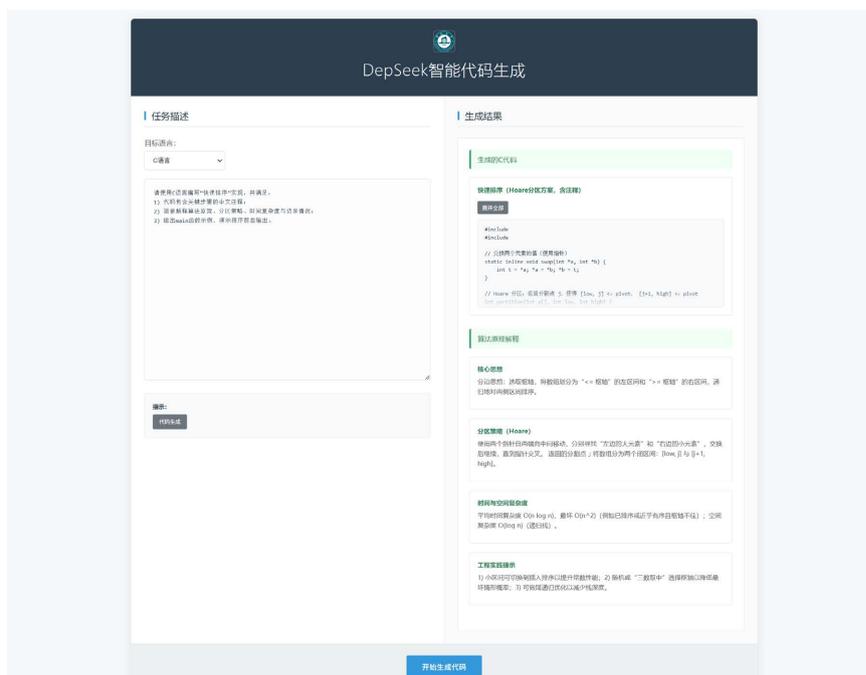


Figure 1. DeepSeek-assisted teaching practice

图 1. DeepSeek 辅助教学实践

在暨南大学的《高级程序语言设计实验课》中，DeepSeek 的应用被进一步拓展至编程作业的智能化批改与反馈环节，构成了教学评估闭环的关键一环。这一实践旨在利用 AI 的深度代码分析能力，为学生提供超越传统对错判断的个性化辅导。

具体而言，当学生完成并提交实验课的编程作业后，系统会调用 DeepSeek 对代码进行深度的静态与动态分析。它能够精准定位从细微的语法错误到深层次的逻辑漏洞(Bug)，并自动生成一份详尽的诊断报告。这份报告不仅清晰地标示出错误代码的位置，更重要的是，能够模拟资深教师的辅导思路：解释 Bug 产生的根本原因，提供具有启发性的修改建议，甚至会链接到相关的课程知识点，引导学生进行巩固学习。具体纠错实践如图 2 所示。

经过教学实践改革，《高级语言程序设计》课程的教学效果显著提升。学生的知识掌握程度明显增强，师生互动也更加高效。2024 年有超过 50% 的学生最终成绩达到 80 分以上，较往年有明显提升。同时，学生在多项重要专业竞赛中表现突出，包括国际大学生程序设计竞赛(ICPC)、中国大学生程序设计竞赛(CCP)以及 ACM SIGMOD 编程竞赛等，均获得了优异名次，充分体现了教学改革的有效性和学生的综合能力提升。

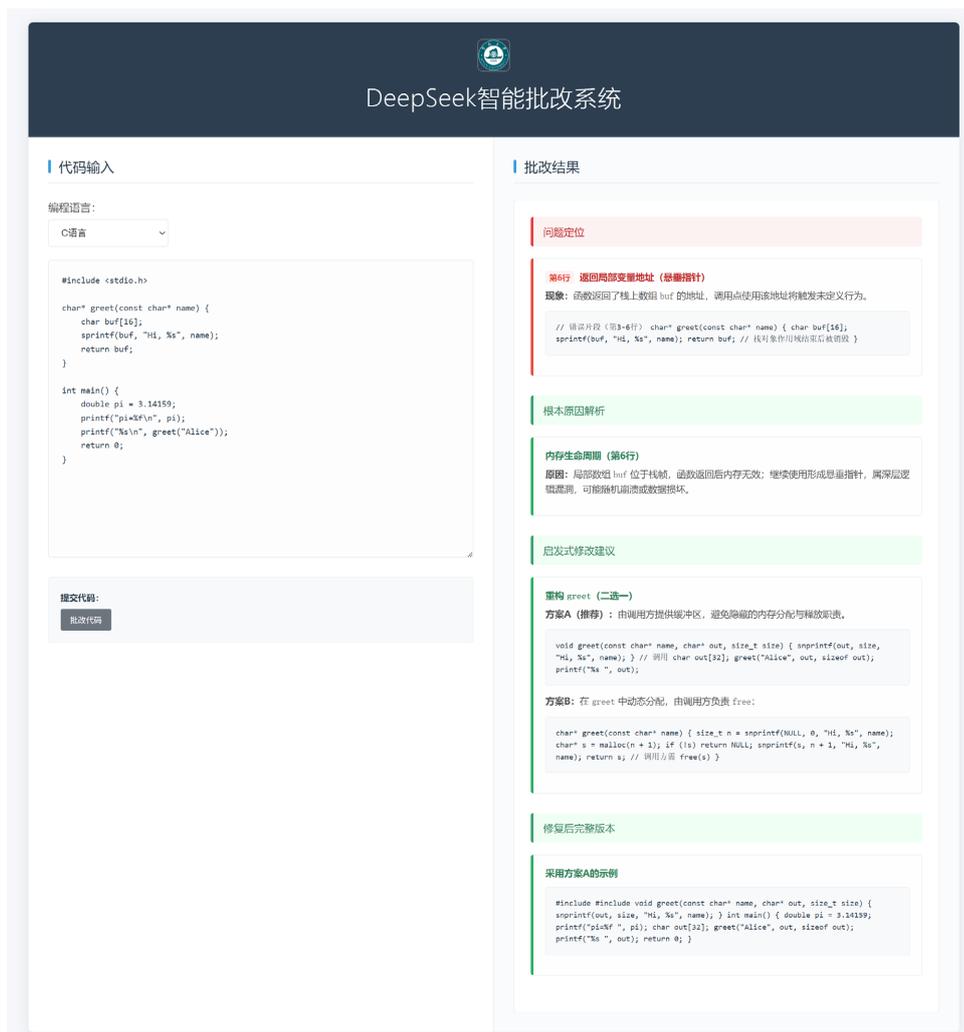


Figure 2. DeepSeek-assisted error correction practice
图 2. DeepSeek 纠错实践

7. 挑战与优化方向

尽管 DeepSeek 在《高级语言程序设计》实践教学中展现出巨大潜力，但在实际应用过程中仍面临诸多挑战，需要持续优化和完善。

7.1. 技术局限与突破路径

当前 DeepSeek 在处理简单语法错误和基础算法问题时表现优异，但在面对复杂算法场景时仍存在局限。特别是涉及多线程同步、并发控制、分布式算法等高级编程概念时，模型的代码诊断准确性和解释深度有待提升。例如，在死锁检测、竞态条件分析、内存一致性模型理解等方面，现有模型往往难以准确定位问题根源并提供有效的修复建议。为此，需要构建专门的领域知识库，融合操作系统原理、并发编程理论等专业知识，通过知识图谱技术增强模型的推理能力，使其能够更准确地理解和分析复杂的程序行为。

虽然 DeepSeek 能够生成语法正确的代码，但在代码效率、可维护性、安全性等非功能性需求方面仍需改进。未来应重点提升模型在时间复杂度优化、空间效率提升、代码重构等方面的能力，确保生成的代码不仅能够运行，更符合工业级标准。

7.2. 数据隐私与安全保障

学生的编程作业和实验数据涉及个人隐私和知识产权，必须建立完善的数据安全保护机制。本地化部署虽然能够在一定程度上保障数据安全，但还需要构建多层次的安全防护体系。借鉴山东大学的混合架构体系经验，可以采用“云-边-端”协同的部署模式：敏感数据在本地处理，通用模型能力通过 API 调用，实现安全性与便利性的平衡。同时，应用差分隐私、同态加密等技术，确保即使在模型训练和推理过程中，学生的个人信息也不会被泄露。

此外，合规性与伦理规范问题，需要建立明确的数据使用协议和伦理审查机制，确保 AI 辅助教学符合教育部门的相关规定和伦理要求。特别是在收集、存储、使用学生学习数据时，需要获得明确授权，并定期进行安全审计。

7.3. 教师数字素养提升

教师作为 AI 辅助教学的主要实施者，其数字素养直接影响教学效果。需要建立分层次、模块化的教师培训体系：基础层面包括 AI 工具的基本操作、提示词(Prompt)编写技巧；进阶层面涵盖多模态教学资源设计、AI 辅助教学策略制定；高级层面则聚焦于基于数据的教学决策、个性化学习路径设计等。通过工作坊、在线课程、实践社区等多种形式，帮助教师逐步掌握 AI 辅助教学的核心技能。

从传统的知识传授者向学习引导者转变，教师需要重新定位自己的角色。应提供教学设计模板、最佳实践案例库、同行交流平台等支持资源，帮助教师适应新的教学模式，充分发挥人机协同的优势。

7.4. 评价体系的重构

传统的结果导向评价已不能完全适应 AI 辅助学习的新模式。需要建立更加注重学习过程的评价体系，包括：AI 工具使用的合理性评估、问题解决思路的创新性评价、独立思考能力的考察等。通过学习行为分析、代码演化追踪等技术手段，全面记录和评估学生的学习过程。

在充分利用 AI 辅助功能的同时，必须警惕学生对 AI 工具的过度依赖。应设计分级的辅助机制，根据学生的能力水平动态调整 AI 的介入程度，确保学生在获得必要帮助的同时，仍能培养独立的问题解决能力和批判性思维。

7.5. 未来发展方向

展望未来, DeepSeek 在编程教育中的应用还有广阔的优化空间。短期内, 应重点提升模型的专业性和安全性; 中期目标是构建完整的智能教学生态系统, 实现教、学、评、管的全流程智能化; 长期愿景则是打造真正意义上的“AI+ 教育”深度融合模式, 让每个学生都能享受到个性化、高质量的编程教育。这需要高校、企业、研究机构的通力合作, 共同推动技术创新与教学实践的有机结合。

基金项目

本研究获得以下基金项目的资助:

广东省本科高校教学质量与教学改革工程项目: “思政融通、虚实相济、智慧赋能、面向产出”的高级语言程序设计课程改革与实践;

教育部产学研合作协同育人项目: 网络空间安全专业“人工智能+”应用安全课程群改革探索;

2025 年暨南大学人工智能赋能研究生课程项目重点项目: 信息隐藏;

暨南大学实验教学改革研究专项: 网络空间安全实验教学标准化建设研究。

参考文献

- [1] Xu, J., Fu, Y., Tan, S.H., *et al.* (2024) Aligning the Objective of LLM-Based Program Repair. arXiv: 2404.08877.
- [2] Yang, B., Tian, H., Pian, W., Yu, H., Wang, H., Klein, J., *et al.* (2024) CREF: An LLM-Based Conversational Software Repair Framework for Programming Tutors. *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, Vienna, 16-20 September 2024, 882-894. <https://doi.org/10.1145/3650212.3680328>
- [3] Chen, Y., Hu, Z., Zhi, C., Han, J., Deng, S. and Yin, J. (2024) ChatUniTest: A Framework for LLM-Based Test Generation. *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, Porto de Galinhas, 15-19 July 2024, 572-576. <https://doi.org/10.1145/3663529.3663801>
- [4] Zhao, S., Zhu, A., Mozannar, H., Sontag, D., Talwalkar, A. and Chen, V. (2025) CodingGenie: A Proactive LLM-Powered Programming Assistant. *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, Clarion Hotel Trondheim, 23-28 June 2025, 1168-1172. <https://doi.org/10.1145/3696630.3728603>
- [5] Nam, D., Macvean, A., Hellendoorn, V., Vasilescu, B. and Myers, B. (2024) Using an LLM to Help with Code Understanding. *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, Lisbon, 14-20 April 2024, 1-13. <https://doi.org/10.1145/3597503.3639187>
- [6] Kazemitabaar, M., Ye, R., Wang, X., Henley, A.Z., Denny, P., Craig, M., *et al.* (2024) CodeAid: Evaluating a Classroom Deployment of an LLM-Based Programming Assistant That Balances Student and Educator Needs. *Proceedings of the CHI Conference on Human Factors in Computing Systems*, Honolulu, 11-16 May 2024, 1-20. <https://doi.org/10.1145/3613904.3642773>
- [7] 李永智, 曹杰杰, 武卉紫, 等. 基于教学思维链的教育大模型推理显化研究[J]. 开放教育研究, 2025, 31(4): 4-11.
- [8] Dickey, E., Bejarano, A., Kuperus, R., *et al.* (2025) Evaluating the AI-Lab Intervention: Impact on Student Perception and Use of Generative AI in Early Undergraduate Computer Science Courses. arXiv: 2505.00100.
- [9] Prather, J., Leinonen, J., Kiesler, N., Benario, J.G., Lau, S., MacNeil, S., *et al.* (2024) How Instructors Incorporate Generative AI into Teaching Computing. *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 2*, Milan, 8-10 July 2024, 771-772. <https://doi.org/10.1145/3649405.3659534>
- [10] Pereira Cipriano, B., Silva, M., Correia, R. and Alves, P. (2024) Towards the Integration of Large Language Models and Automatic Assessment Tools: Enhancing Student Support in Programming Assignments. *Proceedings of the 24th Koli Calling International Conference on Computing Education Research*, Koli, 12-17 November 2024, 1-2. <https://doi.org/10.1145/3699538.3699588>
- [11] 董珊珊, 郭燕, 杨铁林. 生成式人工智能辅助生物类专业 Python 课程的项目式教学[J]. 生物学杂志, 2025, 42(4): 27-30.
- [12] Choi, S. and Kim, H. (2024) The Impact of a Large Language Model-Based Programming Learning Environment on Students' Motivation and Programming Ability. *Education and Information Technologies*, **30**, 8109-8138. <https://doi.org/10.1007/s10639-024-13107-x>
- [13] 穆玲玲, 张行进, 职为梅, 等. 大模型背景下面向系统观的实验教学改革初探[J]. 计算机教育, 2025(7): 116-121.

-
- [14] 王佳, 张谦, 邱爽, 等. 大模型背景下个性化的操作系统实践教学[J]. 计算机教育, 2025(8): 109-115.
- [15] 刘雪峰, 陈玮, 韩旭, 等. 大模型时代计算机实验教学的挑战与机遇[J/OL]. 实验技术与管理: 1-12.
https://kns.cnki.net/kcms2/article/abstract?v=SzYTR_MU8MA7hpDmyP70-x7lbDp3smxchuiBDcYqZyviaOfy8-Y-OpwuwVT0cVJ-mewNqkV5tcEP_ImW-WG9EwITvdqsAX4g34I9KMkmIappTw8-h6DDhOhfnvaU4OCfGxYSnJLwSe30ksH_U3SoloO5CZXMuEJNxK9PVQ3VcRhNkPfu9V21Ew==&uniplatform=NZKPT&language=CHS, 2025-08-21.