College and Job 职业教育发展, 2026, 15(11), 36-44
Published Online November 2026 in Hans. https://www.hanspub.org/journal/ve
https://doi.org/10.12677/ve.2026.1511459

Hans汉斯

生成式人工智能赋能职业院校程序设计课程教学改革

王　超

江西财经大学财经数据科学重点实验室，江西　南昌

收稿日期：2026年9月30日；录用日期：2026年10月26日；发布日期：2026年11月4日

摘　要

生成式人工智能特别是大语言模型的发展，为职业院校程序设计课程的个性化指导、代码调试和过程性评价提供了新的技术条件，但也带来答案依赖与学生主体性弱化等风险。针对个性化反馈不足、自动评测重结果轻解释、评价难以识别真实学习过程等问题，本文首先梳理从智能辅导系统、自动评测到生成式人工智能的程序设计反馈研究，指出大语言模型使反馈生成方式由预定义规则与模板转向按代码情境即时生成、反馈形态由结构化信息转向自然语言解释，带来反馈能力的质变，但反馈强度控制与答案依赖防范仍有赖教学设计；继而利用ProgFeed真实课堂开放数据开展探索性分析，发现反馈形式对学生是否继续提交和是否修改代码的影响有限，自然语言解释性反馈与更大的得分改善和更高的问题解决概率相关，多轮提交后各组最终结果趋于接近；在此基础上，结合职业院校学情特点，提出“教师规则设计－大模型受限反馈－学生自主修订－自动测试验证－教师诊断指导－学习反思”的人机协同教学模式构想，设计五级分层反馈机制，主张将生成式人工智能由“答案生成工具”转化为“形成性学习支架”，推动程序设计课程由结果导向评价转向过程证据导向评价。

关键词

生成式人工智能，程序设计课程，职业院校，人机协同，形成性评价

Teaching Reform of Programming Courses in Vocational Colleges Enabled by Generative Artificial Intelligence

Chao Wang

Key Laboratory of Data Science in Finance and Economics, Jiangxi University of Finance and Economics, Nanchang Jiangxi

Received: September 30, 2026; accepted: October 26, 2026; published: November 4, 2026

文章引用: 王超. 生成式人工智能赋能职业院校程序设计课程教学改革[J]. 职业教育发展, 2026, 15(11): 36-44.
DOI: 10.12677/ve.2026.1511459

Abstract

Generative artificial intelligence, especially large language models, provides new technical conditions for personalized guidance, debugging support, and process-oriented assessment in programming courses of vocational colleges, while introducing risks such as answer dependence and weakened learner agency. To address insufficient individualized feedback, result-oriented automated assessment, and the difficulty of identifying real learning processes, this study first reviews feedback research in programming education, tracing its evolution from intelligent tutoring systems and automated assessment to generative AI, and argues that large language models bring a qualitative change in feedback capability: feedback generation shifts from predefined rules and templates to context-aware generation, and feedback forms shift from structured messages to natural-language explanation, whereas controlling feedback intensity and preventing answer dependence still depend on instructional design. An exploratory analysis of the open ProgFeed classroom dataset then shows that feedback forms have limited influence on whether students continue submitting or modify code; natural-language explanatory feedback is associated with greater score improvement and higher problem-resolution probability; and final outcomes converge after multiple submissions. Based on these findings and the characteristics of vocational learners, the study proposes a teacher-led human-AI collaborative teaching model consisting of teacher rule design, constrained LLM feedback, student revision, automated testing, teacher diagnosis, and learning reflection, together with a five-level tiered feedback mechanism. It is argued that generative AI should be positioned as a formative learning scaffold rather than an answer-generation tool, and that assessment should shift from final results to process evidence.

Keywords

Generative Artificial Intelligence, Programming Courses, Vocational Colleges, Human-AI Collaboration, Formative Assessment

Copyright © 2026 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

《国家职业教育改革实施方案》与《职业教育提质培优行动计划(2020—2023 年)》均对推进信息技术与职业教育教学深度融合、强化实践教学提出了明确要求[1] [2]。与此同时，以 ChatGPT 为代表的生成式人工智能正加速进入课堂教学、学习支持与教育评价等场景，其应用已从资源生成和知识问答拓展至教学的多个环节[3]。元分析表明，生成式人工智能总体上能对学习效果产生积极影响，但其作用受到知识类型、应用方式、学段和教学设计等因素的调节[4]，技术能力并不会自动转化为教育效果，关键在于其能否与具体课程任务和教学目标形成有效匹配。

程序设计课程是职业院校软件技术、大数据技术、人工智能技术应用等专业的核心课程，具有鲜明的实践性和过程性。学生需要反复经历“理解问题－编写程序－运行测试－发现错误－修改代码－再次验证”的完整循环，调试与修订能力本身即是软件技能人才的关键素养。然而在大班额上机实训中，教师和助教难以对每名学生的代码提供及时、个性化的解释；自动评测系统虽能快速给出测试结果，却更擅长回答“程序是否正确”，对“为什么出错、下一步如何修改”的支持相对有限[5] [6]。大语言模型能

够结合任务要求、学生代码与测试信息，以自然语言解释错误原因并提示检查方向，为程序设计反馈提供了新的技术路径[7]；但大模型也能按自然语言要求直接生成完整可运行代码，若被作为获取答案的工具，将绕过问题分析与调试过程，产生认知外包和主体性弱化风险[8]。人机协同学习强调教师、学生与智能工具之间的合理分工[9]，形成性评价强调利用学习过程证据促进后续调整[10]，二者为在释放技术红利的同时守住学习主体性提供了理论基础。

基于此，本文面向职业院校程序设计课程教学，采取“文献梳理－数据探索－模式构想”的研究路径：首先系统回顾程序设计反馈研究从智能辅导系统、自动评测到生成式人工智能的演进脉络，明确本研究的位置；继而利用真实课堂开放数据，对不同反馈形式下学生失败后的修订表现进行探索性分析，揭示解释性反馈的潜在价值与现有反馈模式的局限；在此基础上，结合职业院校学情特点，构建教师主导、学生主体、大模型辅助、自动评测验证的人机协同教学模式，以期为职业院校实践教学改革提供参考。

2. 相关研究综述

2.1. 智能辅导系统与自动评测中的程序设计反馈研究

反馈研究的一项经典综述将反馈列为影响学习成效最重要的教学变量之一，并将反馈内容划分为任务、过程、自我调节和自我四个层面，指出指向过程与自我调节层面的反馈比单纯的任务层面纠错更能促进学习[11]。为突破一对一辅导的资源约束，计算机辅助教学与智能辅导系统长期致力于以技术手段放大反馈能力：从早期按预设程序呈现练习与答案的计算机辅助教学，到依托领域模型和学生模型诊断错误、推送提示的智能辅导系统，反馈的适应性不断增强；范莱恩的比较研究表明，设计良好的智能辅导系统在多个学科中可逼近人类一对一辅导的效果[12]。

在程序设计领域，智能辅导系统多依托预定义的错误规则库与提示序列开展工作，自动评测系统则通过测试执行实现客观评分，其反馈大体属于任务层面的结果反馈[6]。柯宁等对程序设计自动反馈生成的系统综述显示，现有系统的自动反馈多集中于错误位置与测试结果等结果性信息，能够解释错误成因、提供下一步指导的系统相对较少，而这类反馈恰恰是初学者最需要的[13]；编译器错误信息抽象难懂的问题也长期没有得到根本解决[5]。总体来看，这一阶段的反馈价值已获充分确认，但反馈内容受制于预先构建的规则库与模板，覆盖知识域有限，构建与维护成本高，难以适应开放多样的代码错误情境。

2.2. 生成式人工智能赋能程序设计反馈的新进展

大语言模型的出现改变了反馈的生成方式。研究表明，大模型能够针对具体代码与错误生成形式规范、具有可操作性的解释性反馈[7]，学生普遍认为此类反馈及时、具体并贴合自身代码情境，但也存在准确性不稳定和可能被用于直接索取答案的顾虑[14]；国内研究也已从课程模式与流程设计等方面探索其在程序设计类课程中的应用[15]。课堂随机对照研究进一步提供了效应证据：大模型生成的解释性反馈可改善失败提交后的即时修订表现与问题解决概率[16]，而对学生是否坚持尝试的影响相对有限[17]。与此同时，苗逢春警示，生成式人工智能的答案生成能力与教育所需的思维培养之间存在内在张力，必须以教学设计加以约束[8]；人机协同学习研究则主张教师、学生与智能工具按各自优势分工，由教师掌控教学决策、智能工具承担即时支持[9]。

2.3. 研究述评：反馈能力的质变与尚待解决的问题

与智能辅导系统和自动评测相比，生成式人工智能带来的质变体现在三个方面：一是反馈生成方式

由预定义规则与模板转向依据任务要求、学生代码和测试情境即时生成，摆脱了知识域限制；二是反馈形态由结构化错误信息与固定提示链转向自然语言解释与多轮追问，可理解性显著增强；三是反馈内容由有限的备选项扩展为按需定制的解释、提示、追问与反例，能够覆盖长尾化的错误情境。

然而，反馈能力的质变并不等于教育效果的自动提升。经典反馈研究早已表明，反馈的教育价值取决于其层面、时机与强度和学习者状态的匹配[11]；生成式反馈在填补“解释缺口”的同时，也带来直接给出完整答案、削弱学生自主诊断的新风险[8]。既有课堂研究多考察单一形式反馈的平均效应，较少关注反馈强度如何随错误类型与学生状态动态调节，更缺乏面向职业院校大班额、强实训情境的反馈机制设计。因此，有必要在教师主导的教学框架内重新设计生成式反馈的介入边界、层级结构与配套评价：既保留智能辅导系统按学生状态调节支持强度的自适应思想，又利用大语言模型的即时生成能力提供自然语言解释，并以教师规则防范答案依赖。这正是本研究尝试回应的问题。

3. 职业院校程序设计课程教学的现实困境

3.1. 个性化反馈供给不足

程序设计学习的个体差异显著，同一题目中，学生可能在语法、变量类型、循环逻辑、边界条件等不同环节出现问题。职业院校大班额上机实训中，教师与助教需同时处理大量调试请求，难以持续跟踪每名学生的代码与思考过程，部分学生只能依赖反复试错甚至直接寻找完整答案[15]。而且，个性化反馈并非“每个学生得到不同回答”，而是与其当前错误状态和已有知识相匹配的支持：对已能定位错误的学生，过多解释会压缩自主空间；对读不懂测试结果的学生，仅提供失败信息又会使其陷入停滞。如何在不显著增加教师负担的前提下扩大个性化反馈能力，是课程改革首先需要解决的问题。

3.2. 自动评测重结果、轻解释

自动评测系统能够快速执行测试、计算得分并判断是否通过，为学生提供即时验证，但其优势主要体现在结果判定而非原因解释。传统编译器错误信息对初学者仍较为抽象难懂[5]；失败测试、实际输出与期望输出虽提供客观证据，但从“测试现象”到“代码原因”的推断仍须由学生自行完成[6]。这一“解释缺口”正是初学者最易受挫之处，也是大语言模型最适合介入的层面——不是替代测试系统判断对错，而是在测试证据与学生理解之间提供适量的解释。

3.3. 低效试错与答案替代风险并存

多次提交本有利于形成“尝试－反馈－修订－验证”的学习循环，但缺乏有效反馈时也可能演变为盲目修改：学生持续提交，却未形成新的错误诊断与验证策略，提交次数和代码变化并不能直接等同于学习质量的提高。与此同时，大模型能够按自然语言要求快速生成完整代码，学生若直接复制，将绕过问题分析、算法设计、调试与验证的全过程，教师也难以判断其真实能力[8]。因此，课程管理的重点不在于“允许或禁止使用 AI”的二元选择，而在于明确模型可以解释错误、提示方向，同时限制其在关键学习任务中直接给出完整可提交答案。

3.4. 结果导向评价难以识别真实学习过程

传统课程常以最终代码是否通过或最终得分作为主要评价依据，但在允许多次提交和 AI 介入后，相同成绩可能对应完全不同的学习路径：一名学生通过理解反馈逐步解决问题，另一名学生靠大量试错或复制生成代码达到同样结果。课程评价需要从最终结果延伸到提交轨迹、反馈后得分变化、修订说明与学习反思等过程证据[10]，否则将掩盖真实的学习差异。

4. 真实课堂数据的探索性分析：反馈形式与失败后修订

4.1. 数据来源与分析方法

为给反馈机制设计提供经验起点，本文利用 ProgFeed 开放课堂数据集开展探索性分析，考察不同反馈形式与学生失败后修订行为及结果之间的关联。该数据来自美国高校 2025 年秋季一门入门程序设计课程，记录了学生代码提交、自动评测结果和随机分配的反馈条件等信息[16]。本文选取同时包含自然语言反馈、测试用例反馈和无反馈随机条件的四次实验，聚合为提交事件后共 1849 次提交，其中 767 次为失败提交，739 次失败后仍存在下一次提交。分析依次考察失败后是否继续提交、下一次是否修改代码、得分变化以及是否全部通过，并控制当前得分、提交顺序、测试数量和实验固定效应，以学生为聚类单位进行自助法重抽样。需要说明的是，该数据来自国外高校课堂，且反馈形式由系统随机分配而非教学设计，此处仅将分析结果作为反馈机制设计的经验参考，不能直接视为国内职业院校的实际教学效果，也不作因果解释。

4.2. 探索性结果：过程有差异、终点趋同

767 次失败提交中 739 次仍有下一次提交，总继续提交率为 96.35%，三种反馈条件之间差异均不稳定，说明在允许多次提交的情境中学生普遍保持尝试意愿，反馈形式的主要作用并不在于“是否让学生继续做”。在 739 次失败后修订转移中，三组下一次修改代码的比例(80.3%, 73.7%, 79.0%)亦未呈现稳定差异；但自然语言反馈组下一次平均得分变化(+0.359)高于测试用例组(+0.262)和无反馈组(+0.261)，下一次全部通过的比例(50.2%)也明显高于后两组(38.0%, 38.9%)，见表 1。

Table 1. Post-failure revision outcomes under different feedback conditions
表 1. 不同反馈条件下的下一次修订结果

反馈条件	转移次数	下一次修改代码	下一次得分变化	下一次全部通过
自然语言反馈(NL)	203	80.3%	+0.359	50.2%
测试用例反馈(TC)	274	73.7%	+0.262	38.0%
无反馈	262	79.0%	+0.261	38.9%

控制变量后，自然语言反馈相对无反馈的下一次得分变化差异为 0.081 (95%区间[0.015, 0.142])，相对测试用例反馈为 0.084 ([0.024, 0.142])；下一次全部通过的概率分别高约 13.0 个百分点([0.037, 0.217])和 13.2 个百分点([0.044, 0.214])，测试用例反馈与无反馈之间的差异则接近于零。这一结果与相关随机对照研究关于“大模型反馈可改善即时问题解决表现”的发现相呼应[17]，表明解释性反馈的价值主要体现在修订质量而非行为数量。值得注意的是，经过多轮提交后，三组最终通过率均接近 96%，最终成绩差异很小，呈现“过程有差异、终点趋同”的特征：若只观察最终成绩，反馈在修订效率与错误理解上的作用将被掩盖，这为由结果导向转向过程证据导向的评价改革提供了直接依据。

4.3. 探索性发现及其教学启示

上述探索性分析虽源于国外课堂，但为反馈机制设计提供了三点启示。其一，解释性反馈的价值集中在修订质量而非行为数量。三种反馈条件下学生的继续提交率与修改代码比例均无稳定差异，说明学生并不缺乏尝试意愿，缺乏的是有效诊断支持；而自然语言解释性反馈与更大的得分改善和更高的问题解决概率相关，既印证了反馈研究关于“过程层面反馈优于任务层面纠错”的经典结论[11]，也说明自动评测“重结果、轻解释”的解释缺口确实构成学习瓶颈。其二，单一形式的反馈难以适配差异化的错误

类型与学习者状态。数据中反馈形式由系统随机分配，无法依据错误性质和学生水平调节强度：对仅需结果提示的简单错误，自然语言解释可能构成过度支持；对复杂逻辑错误，仅有测试证据又明显不足。其三，“过程有差异、终点趋同”支持评价重心转向过程证据。多轮提交后各组最终通过率与成绩趋于接近，若仅以最终结果评价，反馈在修订质量与错误理解上的作用将被系统性掩盖。对生源基础差异大、实训课时占比高的职业院校而言，这三点启示尤为切要：既要以自然语言解释填补解释缺口，又要以分层递进方式控制反馈强度、保留学生自主诊断空间，还要将过程证据纳入课程评价，这构成了下文教学模式设计的直接依据。

5. 面向职业院校的人机协同教学改革构想

基于上述现实困境、数据探索启示与文献基础，本文提出“教师主导、学生主体、大模型辅助、自动评测验证”的人机协同教学模式构想。该构想的核心是把生成式人工智能定位为受规则约束的形成性学习支架：模型何时介入、解释到什么程度、何时升级为教师介入，均由教师预设的反馈规则决定；教学模式的整体成效，则有待后续教学实践的检验。

5.1. 改革目标与四方协同教学模式

针对上述困境与数据探索发现，本文将生成式人工智能的课程定位由“代码生成工具”调整为“形成性学习支架”。改革目标不限于学生最终写出正确程序，更关注其能否识别错误、理解原因、提出修改方案、通过自动测试验证结果，并将调试经验迁移到新任务，服务于职业院校技术技能人才实践能力的培养。

教学模式由教师、学生、大模型和自动评测系统构成四方协同结构(图 1)。教师位于教学决策端，确定知识目标、任务难度、反馈边界和评价标准；学生位于学习活动中心，承担分析问题、代码修订、结果解释和学习反思；大模型围绕学生当前困难提供解释、提示和追问；自动评测系统执行测试、生成得分并验证程序行为。教师可根据课程知识点和典型错误建立反馈规则库，明确哪些问题允许模型自动解释、哪些只能提供提示、哪些必须由教师介入；模型输入应结合任务要求、学生代码和测试结果，其输出经自动测试与学生再次提交进行校验，形成解释与证据相互校验的反馈机制[9]。在课堂组织上，形成“机器处理常见问题、教师处理复杂问题”的分流：语法错误、简单边界条件等由系统提供规范化支持，核心概念误解、多轮无改善等问题提升至教师层面，既提高反馈覆盖率，也避免教师被大量重复问题占用。

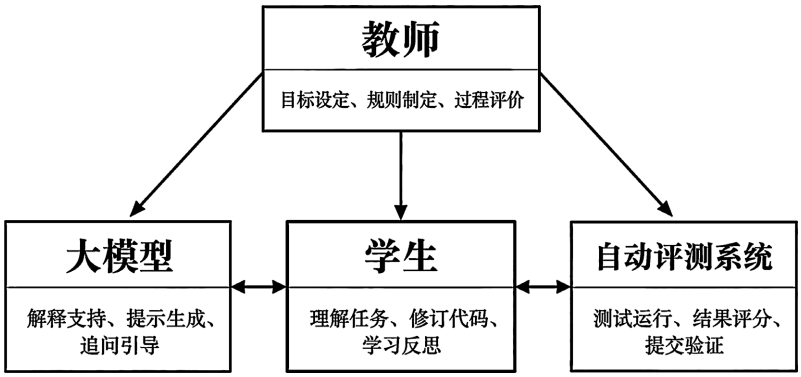


Figure 1. Teacher-led human-AI collaborative teaching model
图 1. 教师主导的人机协同教学模式

5.2. 五级分层反馈机制

为避免大模型一次性提供过多信息，课程设置五级反馈(表 2)：第一层为结果反馈，返回通过状态、

得分和错误位置，让学生先自行分析；第二层为测试证据，提供失败输入、实际输出和期望输出，帮助缩小问题范围；第三层为自然语言解释，由模型说明可能的逻辑原因但不直接给出完整代码；第四层为思考提示，通过追问引导学生检查变量变化、循环条件、边界值或算法步骤；第五层为教师介入，在连续多次提交无改善、同类错误反复出现或问题涉及关键概念时进行人工指导。

Table 2. Five-level tiered feedback mechanism and its pedagogical purposes
表 2. 五级分层反馈机制及其教学目的

反馈层级	主要反馈内容	学生主要任务	教学目的
第一层：结果反馈	通过状态、得分、错误位置	自主检查程序	保留自主诊断机会
第二层：测试证据	失败输入、实际输出、期望输出	缩小问题范围	建立测试现象与代码联系
第三层：自然语言解释	错误原因、逻辑解释	理解错误并形成修改假设	提高反馈可理解性
第四层：思考提示	追问、边界条件、检查方向	自主推理并修改代码	避免直接给出完整答案
第五层：教师介入	复杂诊断、概念纠偏、个别指导	与教师讨论并完成反思	处理高风险和复杂问题

五级结构的设计直接回应数据探索的发现：第一、二层保留自动评测的客观证据功能，测试证据虽不必然提高即时解决概率，却为自主诊断提供可验证的起点；第三层在测试证据不足以支撑诊断时引入自然语言解释，对应探索性分析中解释性反馈在修订质量上的优势；第四、五层则针对单一反馈形式无法适配个体差异的局限，通过追问与教师介入控制支架强度，防止反馈升级沦为答案索取的通道。

分层机制的重点不是增加反馈数量，而是控制反馈强度：学生能够独立解决时尽量保留自主诊断空间，确有困难时再逐步增加支架。在平台实现上，可将反馈层级与提交状态建立映射：首次失败先返回结果与测试证据；再次失败但得分有所改善，则维持较低强度支持；连续多次无改善，再触发自然语言解释与思考提示；同类错误长期重复则提示教师介入。同时允许学生主动申请更高一级支架，但申请须附带条件，如先说明已经检查的位置和尝试过的修改，防止反馈升级沦为快捷获取答案的通道。

5.3. 重构“反馈－修订－验证－反思”教学闭环

传统程序设计实验以“布置任务－提交代码－自动评分”为主流程。本改革将其扩展为“教师规则设计－大模型受限反馈－学生自主修订－自动测试验证－教师诊断指导－学习反思”的连续闭环：教师设置反馈规则与介入阈值；学生提交失败后，模型依据测试结果和代码生成有限度的解释；学生自主修改并记录修改理由；自动评测重新验证；连续失败由教师诊断；问题解决后，学生以“错误原因－关键修改－验证方法”的极简形式总结调试策略。该闭环强调反馈不是学习活动的终点，而是下一次行动的起点：每次反馈都通过学生修订和自动验证产生新的过程证据，教师据此判断学生是在有效学习还是陷入重复试错，使大模型在规则约束内真正进入形成性评价过程，而不是成为课程之外的代码问答工具。

5.4. 评价转向过程证据并明确职责边界

课程评价在保留最终程序正确性和任务完成度的基础上，增加首次提交质量、失败后修订次数、反馈后下一次得分变化、关键修改说明和学习反思等过程指标[10]。成绩构成上，可对关键实验要求提交简短的修改说明和错误反思，为连续失败但逐步改善的学生保留学习进步证据，对直接复制生成代码却无法解释逻辑的情况降低过程评价，既不削弱程序正确性的客观标准，又能体现真实学习投入。同时明确四方职责边界：教师负责目标设计、规则制定、复杂诊断和最终判断；学生负责自主尝试、代码修订、结果解释和反思；大模型只承担解释性和启发性反馈，不生成完整答案、不判定成绩；自动评测系统负责

测试运行和过程记录。课程要建立的不是“完全禁止”或“完全放开”AI，而是可控使用机制——允许学生借助 AI 理解问题、形成思路，但关键修改与解释必须由学生完成。

6. 结论与建议

6.1. 研究结论

本文围绕生成式人工智能在职业院校程序设计课程中的定位与使用方式，通过文献梳理与课堂数据的探索性分析形成以下认识。第一，生成式反馈的价值主要体现在修订质量而非行为数量。探索性分析显示，反馈形式没有稳定改变学生是否继续提交或是否修改代码，但自然语言反馈与更大的得分改善和更高的问题解决概率相关，评价生成式人工智能的教学作用应关注修改是否真正促进问题理解与解决，而不能停留于提交次数等行为指标。第二，自然语言解释与测试证据应形成互补。测试用例提供客观、可重复验证的程序行为证据，自然语言反馈承担解释与认知支架功能，二者是同一学习过程中的不同支持层级；反馈并非信息越多越好，其有效性取决于能否在不替代学生思考的前提下提供恰当支架。第三，反馈机制应根据学生状态和任务情境动态调整。敏感性分析显示证据强度随样本定义和任务情境变化，简单语法错误可能只需结果提示，较复杂的逻辑错误才更需要自然语言解释与追问，改革重点不是寻找固定的“最优反馈”，而是建立能够依据错误类型、任务难度和学习状态调整支持强度的机制。

6.2. 职业院校实施建议

一是建立反馈质量与教师介入机制。将反馈入口与课程平台或自动评测环境对接，由教师预先规定提示模板、输出边界、禁止事项和升级条件，统一反馈口径并记录模型提示与学生后续行为；高频重复问题由系统分流处理，教师聚焦复杂诊断与个别指导。二是强化修订说明与学习反思。要求学生对关键实验提交简短的修改说明，把“是否继续、是否行动、行动是否有效”纳入平时成绩，促使学生在获得更多帮助之前完成必要的自我诊断。三是规范 AI 使用与数据安全。明确“模型能做什么、学生必须做什么”，禁止在关键任务中直接提交生成代码，同时妥善处理代码提交与学习行为数据的存储和使用。四是分步推进试点迁移。可先在程序设计类课程的实验和作业调试环节小范围试点，待规则稳定后扩展至数据结构、软件项目实训等课程，并结合前测后测、过程日志和延迟测试，检验不同反馈层级对学生独立调试能力和知识迁移的影响。

6.3. 研究局限与展望

本文所用数据来自国外单一入门课程，其课程组织、学生背景和评测环境与国内职业院校存在差异，且分析属探索性质，所揭示的关联不宜作因果解释；公开数据难以观察学生的认知过程，代码修改与得分提高也不必然意味着真正理解；核心修订分析基于失败后继续提交的条件样本，部分结果的不确定性随敏感性分析而增大。据此提出的教学模式尚属理论构想，其实际效果有待实践检验。后续可在国内职业院校程序设计和软件实训课程中开展教学试点，将受限大语言模型反馈接入现有实训平台，结合多源数据进一步检验不同反馈层级的效果。总体而言，教学改革的关键不是简单“接入大模型”，而是重新设计反馈、修订、验证与评价之间的关系，实现由“答案生成”转向“学习支架”、由“最终结果”转向“过程证据”、由“自由使用 AI”转向“教师规则约束下人机协同”的三个转变，使生成式人工智能真正服务于职业院校学生程序理解、调试能力和自主问题解决能力的培养。

基金项目

本文系教育部人文社会科学研究青年基金项目(项目编号：25YJCZH242)的研究成果。

参考文献

[1] 国务院关于印发国家职业教育改革实施方案的通知[EB/OL]. https://www.gov.cn/zhengce/content/2019-02/13/content_5365341.htm, 2019-02-13.

[2] 教育部等九部门关于印发《职业教育提质培优行动计划（2020—2023 年）》的通知[EB/OL]. https://www.moe.gov.cn/srcsite/A07/zcs_zhgg/202009/t20200929_492299.html, 2020-09-16.

[3] 刘邦奇, 聂小林, 王亚飞, 等. 生成式 AI 赋能教育: 技术框架、应用场域及价值——2024 智能教育发展研究报告[J]. 中国电化教育, 2025(3): 61-70.

[4] 李丹丹, 成琴, 陈琦, 等. 生成式人工智能赋能课堂教学对学习效果的影响研究——基于 44 项实验与准实验的元分析[J]. 湖北师范大学学报(自然科学版), 2026, 46(3): 111-118.

[5] Becker, B.A., Denny, P., Pettit, R., Bouchard, D., Bouvier, D.J., Harrington, B., *et al*. (2019) Compiler Error Messages Considered Unhelpful: The Landscape of Text-Based Programming Error Message Research. *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*, Aberdeen, 15-17 July 2019, 177-210. https://doi.org/10.1145/3344429.3372508

[6] Paiva, J.C., Leal, J.P. and Figueira, Á. (2022) Automated Assessment in Computer Science Education: A State-of-the-Art Review. *ACM Transactions on Computing Education*, **22**, 1-40. https://doi.org/10.1145/3513140

[7] Kiesler, N., Lohr, D. and Keuning, H. (2023) Exploring the Potential of Large Language Models to Generate Formative Programming Feedback. 2023 *IEEE Frontiers in Education Conference* (*FIE*), College Station, 18-21 October 2023, 1-5. https://doi.org/10.1109/fie58773.2023.10343457

[8] 苗逢春. 生成式人工智能技术原理及其教育适用性考证[J]. 现代教育技术, 2023, 33(11): 5-18.

[9] 王一岩, 刘淇, 郑永和. 对话式人机协同学习: 本质内涵与未来图景[J]. 中国电化教育, 2024(11): 21-27.

[10] 卢国庆, 杨沁, 贺相春. 生成式人工智能赋能高等教育形成性评价的价值、挑战及路径[J]. 电化教育研究, 2024, 45(11): 84-91.

[11] Hattie, J. and Timperley, H. (2007) The Power of Feedback. *Review of Educational Research*, **77**, 81-112. https://doi.org/10.3102/003465430298487

[12] VanLehn, K. (2011) The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, **46**, 197-221. https://doi.org/10.1080/00461520.2011.611369

[13] Keuning, H., Jeuring, J. and Heeren, B. (2019) A Systematic Literature Review of Automated Feedback Generation for Programming Exercises. *ACM Transactions on Computing Education*, **19**, 1-43. https://doi.org/10.1145/3231711

[14] Zhang, Z., Dong, Z., Shi, Y., Price, T., Matsuda, N. and Xu, D. (2024) Students' Perceptions and Preferences of Generative Artificial Intelligence Feedback for Programming. *Proceedings of the AAAI Conference on Artificial Intelligence*, **38**, 23250-23258. https://doi.org/10.1609/aaai.v38i21.30372

[15] 张红卓, 周小宝, 许玉焕, 等. 生成式人工智能赋能计算机程序设计类课程教学创新[J]. 计算机教育, 2024(7): 44-48.

[16] Heickal, H. and Lan, A. (2026) A Classroom Study of LLM-Generated Feedback Intervention in Introductory Programming. arXiv:2606.08807.

[17] Zhou, Y., Pankiewicz, M., Paquette, L. and Baker, R. (2025) Impact of LLM Feedback on Learner Persistence in Programming. *The* 33*rd International Conference on Computers in Education*, Chennai, 1-5 December 2025, 1-10. https://doi.org/10.58459/icce.2025.5957