

认知负荷视域下基于 Trae 的人机协同教学模式构建与实践*

钱清** 唐瑞雪 申红婷 陈清容

贵州财经大学信息学院, 贵阳, 550025

摘要 针对传统本科编程类课程存在的知识抽象难懂、代码纠错滞后、学用转化脱节等问题, 本文基于认知负荷理论, 提出了一种依托 AI 原生集成开发环境 Trae 的人机协同教学模式。以《数据结构》课程为例, 构建动态过程呈现、实时纠错反馈与自适应闭环练习三位一体的人机协同教学模式。采用准实验研究法, 选取同年级两个平行班进行为期一学期的教学对照实验。实验结果表明, 引入 Trae 的实验组在学业成绩上显著优于对照组, 不及格率较对照组降低 24 个百分点, 80 分以上高分段人数占比高出 5 个百分点, 验证了 Trae 辅助编程教学在提升学业表现方面的成效。最后, 针对实践中暴露的学生过度依赖 AI 工具的风险, 提出伪代码先行等过程性约束策略, 为生成式人工智能在高校理工科课程中的深度融合提供实证参考。

关键词 认知负荷理论; 人工智能教育; 数据结构; Trae

Construction and Practice of a Human-AI Collaborative Teaching Model Based on Trae from the Perspective of Cognitive Load Theory

Qing Qian Ruixue Tang Hongting Shen Qingrong Chen

Information School
Guizhou University of Finance and Economics,
Guiyang 550025, China
QingQian@mail.gufe.edu.cn

Abstract—To address the prevalent issues in traditional undergraduate programming courses, such as abstraction of core concepts, delayed code debugging feedback, and a disconnect between theoretical learning and practical application, this study proposes a human-AI collaborative teaching model based on Cognitive Load Theory and facilitated by Trae. Taking the Data Structures course as a case study, a triadic instructional model was developed. It integrates dynamic process visualization, real-time corrective feedback, and adaptive closed-loop practice. Meanwhile, a quasi-experimental design was employed with two parallel classes from the same grade over one semester. The results indicate that the experimental group utilizing Trae demonstrated significantly higher performance compared to the control group. Specifically, the failure rate was 24 percentage points lower, while the proportion of students scoring above 80 was 5 percentage points higher. These findings validate the significant efficacy of Trae-assisted programming instruction in enhancing academic performance. Finally, to mitigate the risk of technological dependency, the study proposes process-oriented constraints such as requiring students to draft pseudocode before coding. These measures provide empirical insights for effectively integrating generative AI into higher education STEM curricula.

Keywords—Cognitive Load Theory; AI in Education; Data Structure; Trae

1 引言

生成式人工智能技术的快速发展正改变着高等教育形态^[1], 计算机类专业核心课程亟需探索智能化教学转型路径^[2]。在理工科教育中, 如数据结构、算法设

计与分析等编程类课程, 作为培养计算思维的核心载体, 承担着培养学生计算思维、问题求解能力和程序设计能力的重任, 其教学质量直接关系到学生的工程实践能力与创新素养。然而, 这类课程在实践过程中受制于传统教学模式的局限^[3]。首先理论存在抽象难懂的问题, 算法核心逻辑嵌套复杂, 传统教学手段难以直观展示其动态执行流程, 学生理解难度大。其次是反馈滞后问题, 学生在编写算法代码时, 面对逻辑漏洞通常只能依赖编译器的底层报错信息, 从代码编

*基金资助: 本文得到以下项目支持: 贵州省基础研究计划面上项目(黔科合基础 MS(2026)056); 贵州省本科教学内容和课程体系改革项目(GZJG2025124)

*通讯作者: 钱清 qingqian@mail.gufe.edu.cn

写、发现错误到代码修正的周期过长，容易打断学生连续学习状态、降低学习专注力，从而影响学习效率和积极性。最后存在学用脱节问题，教师课堂推演的解题思路仅停留在理论层面，学生即便听懂逻辑，课后独立编码时仍无从下手。

认知负荷理论由澳大利亚教育心理学家 John Sweller 于 20 世纪 80 年代提出，该理论认为人的工作记忆容量有限，若同时处理的信息过多将会导致学习效率下降^[4]。在传统编程类课程中，学生往往需要同时思考程序逻辑、语法细节、代码错误原因等，这种多重任务极易导致认知超载，使其陷入学得累、学不会的困境。显然，仅依靠传统教学设计，难以从根源上突破该认知瓶颈^[5]。在数字化教学转型的背景下，如何借助智能技术手段优化认知负荷分配，成为亟待解决的关键问题。生成式人工智能的涌现为解决这一难题提供了新的契机，引入 AI 工具可在一定程度上将学生从语法记忆、代码调试等低阶认知任务中解放出来^[6]，使其有限的工作记忆资源能够集中于算法逻辑构建与计算思维培养等高阶认知活动，从而实现认知负荷的有效卸载与教学效率的显著提升。

《数据结构》包含递归、指针、动态存储结构等大量非线性知识点，极易引发学生认知负荷超载。因此，本文选取该课程作为研究对象，对于验证人机协

同模式在缓解复杂认知任务中的效能具有重要意义。基于此，本研究基于认知负荷理论，创新性地引入 AI 原生集成开发环境 Trae，该工具可承接语法校验、代码拼写修正等低认知负荷任务，将学生从繁琐的细枝末节中解放出来，使其得以聚焦于核心逻辑推演与编程思维建构。因此，本研究通过构建具象化动态过程呈现、实时纠错、闭环练习三位一体教学模型，系统探究其在本科编程课程中的落地路径与实施策略，为 AI 赋能计算机专业核心课程教学提供可复制、可推广的实践范例。

2 基于 Trae 的 AI 与教育新范式构建

Trae 深度融合代码解析与自然语言理解技术，可精准感知代码上下文，是具备完整逻辑解析能力的教学智能体。基于此，本研究将教师转型为学习架构师，学生转变为主动探究者，教学模式由先讲授、后练习的线性结构，转变为讲授、练习、思考螺旋融合的一体化模式。如图 1 所示，本研究聚焦三大核心：一是动态可视化呈现算法逻辑，降低认知门槛；二是实时纠错即时引导编程行为，实现精准辅助；三是自适应出题满足个性化练习需求。三者协同构建起学中练、练中思、思中悟的良性循环。该模型旨在通过降低外在认知负荷、优化内在认知负荷，实现学习效率与思维质量的双重提升。

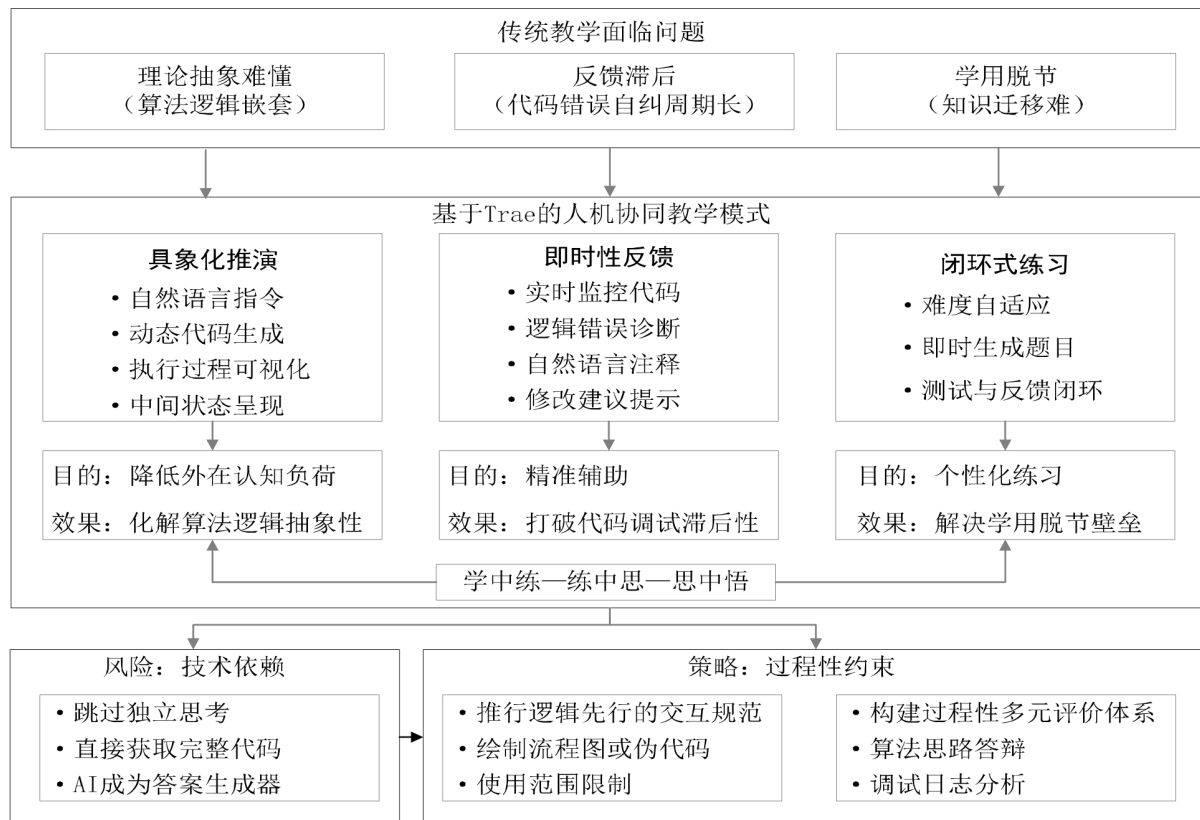


图1 基于Trae的人机协同教学模式框架

2.1 具象化推演：动态可视化呈现，化解算法逻辑抽象性

本研究借助Trae的对话能力实现算法执行过程的动态可视化推演。针对编程知识抽象难懂的问题，Trae打破了传统IDE仅用于代码编辑的局限，将其转化为对话形式的交互界面。在教学过程中，教师无需再耗费大量时间在黑板上手写复杂的变量交换流程，而是可以直接向Trae输入自然语言指令，Trae能够即时生成对应的代码片段，并通过注释或伪代码形式，将算法执行的中间状态以可视化的方式呈现出来。通过将抽象代码符号转化为具象执行流程，帮助学生在认知记忆中建立完整、清晰的算法执行模型，有效突破了理解瓶颈。这种基于生成式AI的逆向工程思维模式，已被证明能显著提升学习者的感知度和协作意愿^[7]。

2.2 即时性反馈：打破代码调试的滞后性

在传统课堂中，教师需要在黑板上逐行书写算法基准代码，耗时较长且容易因语法细节干扰学生对算法核心逻辑的关注。引入Trae后，教师在授课时首先通过自然语言描述算法核心逻辑，令Trae快速生成基准代码。这使得教师能够将宝贵的课堂时间从繁琐的代码录入中解放出来，转而聚焦于讲解不同算法解题思路的逻辑推导。

学生随堂编码环节，Trae实时纠错功能能够显著降低反馈滞后。传统集成开发环境仅能抛出运行异常或超时提示，学生需要自行逐行排查。Trae环境中，当代码存在逻辑缺陷时，Trae能够结合上下文进行实时错误诊断，不仅能定位错误行，还能通过自然语言解释错误产生原因，并给出修改建议。这种从语法报错升级为逻辑层面解释的即时反馈机制，加深了学生对

算法底层原理的理解，同时大幅缩短纠错周期，使学生保持连贯的学习状态，显著提高程序调试效率。研究表明，即时反馈能够有效提升学习者的自我效能感，增强学习动机，促进学习投入。在编程类课程这一高认知负荷的学习任务中，即时反馈的价值尤为突出。

2.3 闭环式练习：解决学用脱节的壁垒

针对传统课堂教学中学用脱节的问题，本研究设计了基于Trae的讲练闭环流程。在传统课堂教学过程中，课后练习模式让教师无法实时掌握课堂学情，造成讲授与练习环节相互割裂；同时，标准化课后作业难以适配不同基础学生的个性化学习需求。为缓解这一现象，本研究借助Trae的强大内容生成能力，教师可依据课堂实时学情，快速生成分层难度的配套习题与学习辅助资源。该功能不仅大幅提升教师备课效率，还为学生动态化、个性化学习提供支撑。此外，学生在课堂上即时作答并提交课堂练习，教师能够实时验证其知识掌握情况。该流程有效缩短知识内化周期，大幅提升课堂互动频次与单位课时教学密度。

3 教学实践案例：以快速排序算法为例

快速排序是《数据结构》课程公认的学习难点，其中递归调用、分区（partition）操作、边界条件处理等内容对学生认知能力要求较高。鉴于此，本研究选取快速排序算法作为典型案例，深入阐述引入Trae IDE后的课堂教学新模式。该教学模式可有效降低学生认知负荷，使其将更多精力聚焦算法核心逻辑，切实推动计算思维养成。

3.1 具象化推演实施

图2演示了Trae赋能快速排序算法的推演过程。

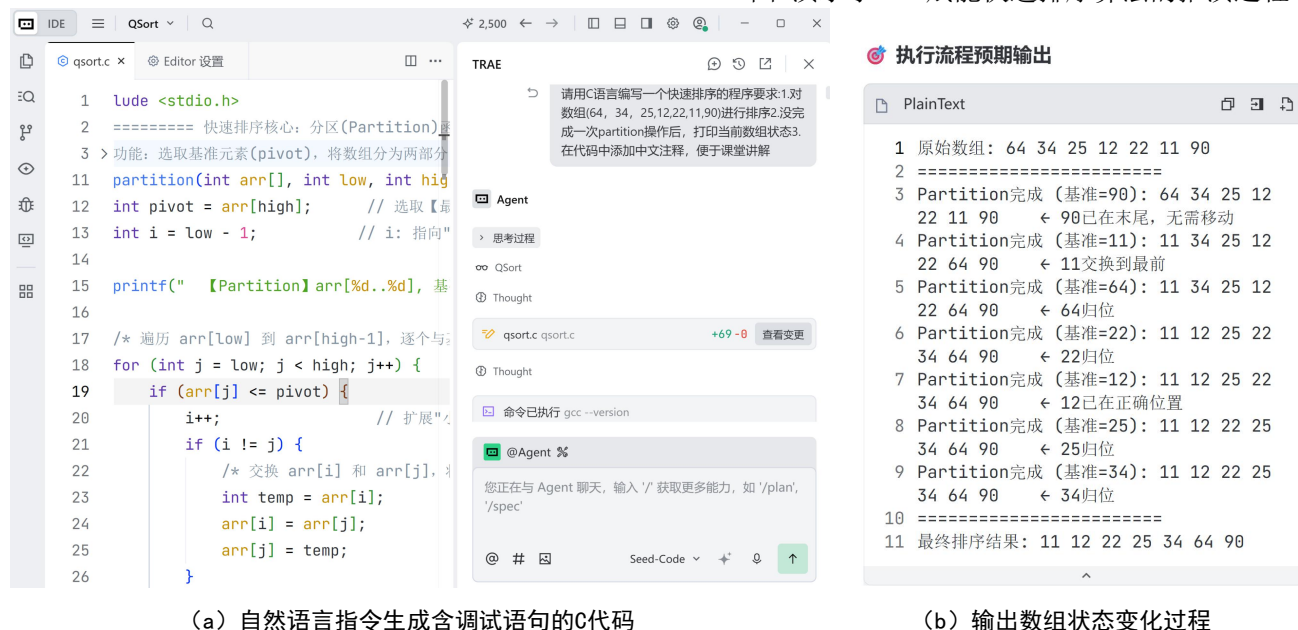


图2 Trae生成快速排序动态推演代码及可视化结果

传统教学仅依靠 PPT 展示数组划分过程, 学生很难将排序流程与代码执行顺序建立对应关系。通过引入 Trac 赋能快速排序算法的学习, 教师引导学生进行逆向工程思维训练, 利用 Trac 即时生成包含详细中间状态打印的快速排序代码。通过对比代码逻辑与控制台输出的动态变化, 引导学生反推递归调用的边界条件, 将隐性的思维过程显性化。教师向 Trac 发出指令: “请用 C 语言编写一个快速排序的程序。要求: 1. 对数组 {64, 34, 25, 12, 22, 11, 90} 进行排序; 2. 每完成一

次 partition 操作后, 打印当前数组状态; 3. 在代码中添加中文注释, 便于课堂讲解。” Trac 即时生成包含辅助打印语句的完整代码。运行后, 控制台以缩进层级清晰地展示了递归深入与回溯的全过程。这种代码与实时执行过程对照的教学方式, 实现了抽象算法逻辑到具象运行结果的直观映射, 显著提升了学生对算法逻辑结构的构建效率。

3.2 实时纠错: 强化底层逻辑认知

```
60 void quickSort(int arr[], int low, int high, int tot)
-   if (low > high) {
61+   if (low < high) {
62       /* 步骤1: 分区, 获取基准的最终位置 */
63       int pi = partition(arr, low, high, totalSize);
64       /* 步骤2: 递归排序基准【左侧】子数组 */
65       quickSort(arr, low, pi - 1, totalSize);
66
67       /* 步骤3: 递归排序基准【右侧】子数组 */
68       quickSort(arr, pi + 1, high, totalSize);
69   }
70 }
71
72 /* ===== 主函数 ===== */
73 int main() {
74     /* 待排序数组 */
```



图3 Trac针对逻辑错误的实时诊断与修正建议

在学生编码实践环节, 教师布置独立实现快速排序算法的任务, 要求包含 partition 函数和递归调用。学生在 Trac 环境中进行编码, Trac 实时监控代码质量并提供反馈。如图3所示的实践过程, 当学生代码存在缺陷时, Trac 可精准定位问题、解析底层错误成因并给出可有效的修改方案。进一步地, 讲授基础快速排序算法后, 教师引导学生探讨当数组已基本有序时快速排序的表现。学生在 Trac 中输入测试用例, 发现虽然结果正确, 但递归深度骤增。此时教师引导学生向 Trac 提问: “当前快排在有序数组下出现了什么问题? 如何优化?” Trac 不仅指出了程序时间复杂度退化为 $O(n^2)$ 和最坏空间复杂度 $O(n)$ 的栈溢出风险, 还同步提供三数取中法、随机基准快排两类优化实现代码。教师要求学生选择一种优化方案进行实现, 并对比优化前后的运行时间。通过对比新旧代码在统一测试集下的运行耗时, 时间复杂度不再是抽象公式, 转化为了可直观测量的工程实践感受。这一过程的渐进式学习方式, 使得学生不仅掌握了快速排序的优化技巧, 更培养了算法分析和优化的意识。

3.3 即时生成随堂练习: 构建讲练闭环

学生通过课堂掌握了快速排序的思想, 若让学生即时编码实践, 实现找第 K 大元素的应用时, 多数学生无从着手, 相关训练只能布置为课后作业, 破坏学习反馈的连续性。因此, 通过即时练习巩固反馈就显

得尤为重要。如图4所示, 教师可以利用 Trac 的对话能力, 快速生成阶梯式练习。



图4 即时生成随堂练习

例如发送指令: “请基于上述快速排序代码, 为本科生设计2道随堂测试题, 要求包含1道分区逻辑的理解题, 和1道递归终止条件的逻辑题, 并附带详细解。” 随后, Trac 根据发出的指令, 生成不同难度的随堂测试题, 学生可以根据自己的理解, 选择不同难度的题目

在Trae环境中完成练习。面临未见应用陷入认知阻塞时,可以通过向Trae索要解题线索而非答案。例如:“我找不到如何利用快排求第K大元素的思路,请给我一个起点提示。”这类启发式线索可充分支撑学生自主思考,让每位学生在自身最近发展区获得学习成就感,真正实现了知识的当堂吸收与迁移。

此外,教师还可利用Trae生成快速排序的递归轨迹推演题,要求学生手动模拟数组元素的划分与交换过程。这种逻辑推演与代码实现双向印证的学习方式,能够深化学生对算法动态运行机制的理解,更有效检验计算思维的内化程度。

4 研究设计数据分析

表1 传统教学组与Trae辅助教学组期末成绩分布表(N=58/组)

分数段	传统组 (人数/占比)	Trae组 (人数/占比)	人数变化	占比变化
90分以上(优秀)	0人/0%	2人/3%	+2人	+3%
80-90分(良好)	6人/10%	7人/12%	+1人	+2%
70-80分(中等)	7人/12%	11人/19%	+4人	+7%
60-70分(及格)	9人/16%	16人/28%	+7人	+12%
0-60分(不及格)	36人/62%	22人/38%	-14人	-24%
总计	58人	58人	0人	0%

表2 Trae辅助教学组与传统教学组期末成绩统计表

班级	平均分	标准差	最高分	最低分	及格率
Trae组	61.27	20.39	98.0	31	62.0%
传统组	57.37	14.06	89.5	27.5	38.0%

为实证检验教学成效,本研究遵循准实验研究范式,选取同年级两个各58人的平行班级,开展为期一学期的准教学对照实验。实验组引入Trae构建AI交互式教学场景,实现编程核心逻辑的动态推演与即时反馈;对照组沿用传统讲授式教学模式,即以PPT演示为核心、辅以课后习题巩固。从表1、表2中呈现的数据可知,引入Trae并未牺牲中等生拔高少数尖子生,而是通过削减外在认知负荷、调控内在认知负荷,推动全班成绩整体提升。具体体现在以下几个方面:

(1) 学业赋能效应显著,低分群体实现有效转化

传统教学组不及格率达62%,Trae教学组不及格率降至38%,不及格学生绝对人数减少14人,不及格占比下降24个百分点。这一变化印证了Trae工具在降低初学者认知门槛上的核心价值,使其分担了学生在代码调试、逻辑解析、即时反馈等方面的外部认知。这种伴随式的智能支持有效降低了初学者的认知负荷与

试错挫败感,让传统课堂中极易陷入学习停滞的基础薄弱学生拥有持续探究的学习动力,成功跨越了及格门槛。这种将基础薄弱者从消极被动转化为主动达标的学业赋能效应,是传统单向讲授难以实现的。

(2) 学业进阶效应显著,中高分段覆盖面拓宽

通过观察发现,在60-70分、70-80分、80-90分三个中等及高分区间,Trae组人均多于传统组,三段合计多出12人,对应总占比提升1个百分点。其中60-70分段学生占比提升12个百分点,证明Trae的实时引导可有效助力及格临界生突破知识理解瓶颈,实现向及格线的跃升;同时,70-80分段也增加了4人,表明正在驱动中等生向更高思维层次迈进。此外,Trae组的平均分为61.27分,较传统组的57.37分提高了3.9分。这一增长是因为60-80分数段人数的显著增加,表明AI辅助教学能够有效促进中等生群体的学业提升。

(3) 高阶思维激活,优生培养实现突破

尽管两组平均分差值有限,但传统教学组无90分以上优秀学生,Trae组不仅诞生了2名优秀学生,最高分更是达到了98.0分。该变化证明引入Trae原生集成开发环境后,学有余力的学生拥有更高效率的自主探索渠道,可开展更深层次算法思维训练,具备冲击顶尖成绩的潜力。显然,这打破了AI仅适用于基础辅助的刻板印象,证明其同样具备高阶思维培优功能。

(4) 差异化教学启示

值得注意的是,表2中Trae组的标准差大于传统组。这一数据也反映出AI辅助教学是一把双刃剑。该模式在大幅提升及格率的同时,也增加了班级成绩分散度。这说明AI工具可能加剧了学生的学习分化:自主学习能力强学生收益显著;而部分自律性薄弱的学生虽达到及格标准,但未完成算法思维深度内化,这也为后续的教学改进指明了方向。

综合而言,Trae辅助教学带来的成绩分布整体右移,并非单纯依靠少数学生分数大幅提升,而是源于人机协同机制可精准适配不同层次学生,提供弹性化学习支撑。低分群体的有效转化与中高分段的系统性进阶,共同印证了该模式在突破个体学习瓶颈上的独特优势。这种交互式教学方式打破了传统课堂一刀切的问题,让技术赋能从单纯教学辅助工具,转变为助推学生算法思维进阶的核心载体,实现从基础达标到高阶思维培养的贯通,为编程类及工程实践类课程的智能化改革提供可复用的方法论。

5 教学反思与对策

实践发现,生成式AI的引入也带来了学生过度依赖的风险。在自主探究环节,部分学生倾向于跳过独立思考,直接获取Trae生成的完整代码。这种做法将辅

助工具变成了单纯的答案库，削弱了学生思维锻炼的机会。为防范技术依赖，教学中需明确人机分工，确保AI仅作为辅助工具，而非替代学生的思维过程。据此，本研究提出两项过程性约束策略：

（1）推行逻辑先行的交互规范。要求学生在向Trac提问前，必须先完成算法流程图或伪代码的绘制。Trac的使用范围应限制在语法补全、代码纠错、细节优化等外在认知负荷任务，避免直接生成完整的解题代码。

（2）构建过程导向的多元评价体系。通过优化考核权重，增加算法思路答辩、调试日志分析等过程性考核的比重。评价重心应从代码结果的正确性转向利用Trac进行假设验证的能力，从而引导学生利用技术培养高阶思维。

6 结 语

本文构建的Trac人机协同编程教学模型，依托具象化推演、实时纠错反馈、自适应闭环练习三大模块，有效化解传统编程课程算法抽象、调试反馈滞后两大痛点。实践证明，该模式不仅显著降低不及格率，更达成了学生整体成绩的提升。该人机协同教学模式，

推动编程类课程从单纯代码语法训练，转向复杂计算问题求解能力的系统性培养。后续研究将开展多学期、长周期纵向追踪实验，并基于学生多维学习行为数据，探索动态自适应习题推荐机制，推动课堂教学由单一工具辅助阶段，向完整智慧教育生态演进。

参 考 文 献

- [1] 别敦荣. AI 技术应用于大学教育教学的理论阐释[J]. 中国大学教学, 2024(05):4-9+2.
- [2] 叶鸿, 王海涛, 阮瑞, 等. 新工科背景下的数据结构混合式教学与 AI 实践探索[J]. 计算机教育, 2026(04):187-191.
- [3] 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(07):44-48.
- [4] 何思源, 卢玲. 生成式编程 AI 辅助计算机专业课程教学的探索与实践[J]. 计算机技术与教育学报, 2026, 14(2):123-129.
- [5] 刘凌, 吴永芬, 施蕾, 等. AI 协同教学在程序设计类课程教学中的应用场景研究[J]. 计算机技术与教育学报, 2026, 14(4):1-8.
- [6] 熊璋, 武迪. 教育强国建设背景下人工智能赋能教育创新的路径探索[J]. 人民教育, 2023(19):6-10.
- [7] 徐慧, 季玲. AI 时代逆向思维导向的教学模式重构及实现路径[J]. 中国大学教学, 2025(12):50-58.