

从“机械试错”到“反思对话”：初中信息科技教学中苏格拉底式 AI 的应用研究

From "Mechanical Trial-and-Error" to "Reflective Dialogue": Research on the Application of Socratic AI Guidance in Junior High School Information Technology Teaching

郭大为^{1*}

¹ 上海市格致初级中学

* 3295432269@qq.com

【摘要】 针对初中信息科技教学中学生的“机械试错”现象，本研究构建了苏格拉底式 AI 智能体。通过检索增强生成与提示词工程，该系统通过苏格拉底式提问引导学生进行反思性建构。在初中六年级的实证研究表明，实验组学生的逻辑测试成绩显著高于对照组，且代码提交频次降低、单次修改成功率提升。研究证实，苏格拉底式 AI 引导能有效促使学生从机械试错转向反思对话，提升计算思维能力。

【关键词】 GenAI；苏格拉底式教学；初中信息科技

Abstract: Addressing "mechanical trial-and-error" in junior high science technology education, this study developed a Socratic AI agent, utilizing RAG to provide heuristic guidance rather than direct answers. Empirical results from 6th-grade classes demonstrate that the system significantly improved students' logic test scores. Furthermore, it reduced code editing frequency while increasing success rates. These findings indicate that Socratic AI effectively shifts students from mechanical operations to reflective dialogue, enhancing computational thinking.

Keywords: GenAI, Socratic Teaching, Junior High School Science Technology

1. 引言

随着基础教育“双新”改革的全面深化，教育数字化转型已进入“素养导向”的深水区。《义务教育信息科技课程标准（2022 年版）》明确指出学生应具备“抽象、分解、建模、算法设计”等能力，并强调在解决真实情境问题的过程中培养核心素养。

然而，初中生对于抽象的逻辑符号和真值表难以直观理解，图形化编程“低门槛”的特性反而容易让学生陷入“拼积木”式的盲目试错（Meerbaum 等，2010）。此外，AI 平台的引入使学生倾向于直接索取解决方案，导致了“认知外包”现象。如何触发思考，实现从“机械试错”向“反思对话”成为提升教学实效的关键（Bjork，2011）。

近年来，生成式人工智能（GenAI）为解决上述痛点提供了新契机（Ma 等，2024）；在计算机科学教育中，苏格拉底式教学也被证明能促进学生的批判性思维和元认知能力（Kao 等，2025）。为此，本研究尝试通过 GenAI 进行苏格拉底式诘问，构建动态“教学支架”，引导学生将内隐的逻辑偏差外化为显性的反思过程，实现从机械课堂活动向反思性建构的转变。

2. 相关研究综述

2.1. 苏格拉底教学的策略

提问策略是苏格拉底教学能否发挥优势的关键，应该具备重点突出与目标驱动两大特征（张誉月等，2025）：针对关键内容提出的问题能有效降低学生的认知负担；具有层次的问题能帮助学生完成课堂实践。

2.2. “机械试错”的负面影响

图形化编程已成为核心素养背景下信息科技教学必备的平台。然而，在处理逻辑运算等抽象任务时，学生倾向于“低成本”地随意更换程序积木，直到运行成功。虽然“拼凑”行为帮助学生探索了模块功能，但消极的“拼凑”缺乏明确规划，会导致疲劳和挫败感（Dong，2019）。

2.3. 大语言模型赋能的苏格拉底教学

Scarlatos 等提出的 SocraticAI 框架（2024）证明，通过提示词工程，LLM 能够避免直接生成答案，而是通过分层提问显著提升学生的学习留存率。然而，LLM 在信息科技教学的应用仍面临挑战：由于“幻觉”和不可控的生成内容，回复是否能够契合教师的教学设计；大班额信息科技课堂中，如何利用 AI 实现苏格拉底式教学仍需深入实证。

3.初中信息科技苏格拉底式智能体设计

本研究通过生成苏格拉底式对话，实现教学引导。智能体整体设计遵循“高内聚、低耦合”的原则，由 RAG 知识引擎、教学决策和对话生成等模块协同工作。具体如图 1 所示：

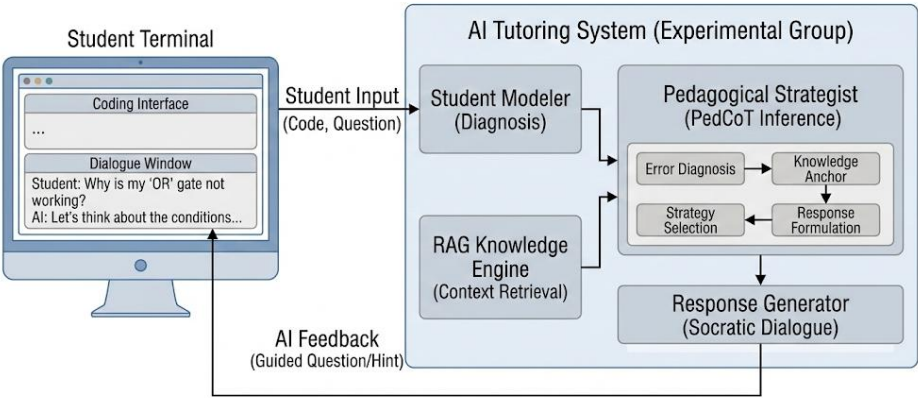


图 1 初中信息科技苏格拉底式智能体设计

在课堂活动中，学生需要先明确课堂活动的内容，然后借助学习支架，利用知识和核心素养形成解决方案；最终完善代码和学习活动单，形成学习成果。基于以上过程，图形化编程平台先将程序积木转换为便于 LLM 理解的 Python 代码。系统收集代码并识别学生状态：若短时间内连续提交代码且改动微小，或运行后反复出现同一报错，都可能为“机械试错”。

之后，为避免 LLM 产生脱离教材和教学设计的幻觉，系统使用检索增强生成(RAG)技术分层加载所需内容：通用知识（基础概念、新课标和教学指南等）、教学资料（单元设计、教学设计与 PPT 等）、活动任务（学生活动单或要求），确保 LLM 的引导与教学内容对齐。

如果 LLM 直接输出答案，苏格拉底式教学则难以保证。为此，系统使用教学决策模块充当中间层，基于学生状态与 RAG 内容，以预设的教学策略进行指导，包括明晰问题（引导学生明确困难点）、构建认知冲突（通过反直觉场景促使自我修正）、问题拆解（降低认知负荷）、回顾反思（复盘解决路径，促进知识内化）。

最后，对话生成模块接收 RAG 内容和教学决策，以苏格拉底式教学引导学生思考、解决问题并反思总结。本模块使用零样本学习构造提示词，避免依赖提示词示例（Jia 等，2024）：角色为精通新课标的教师；任务为基于 RAG 检索识别认知困难点并构建苏格拉底式提问；约束条件为严禁直接输出代码和答案、优先使用教材案例、单次回复仅聚焦一个逻辑步骤等。

基于教学思路设计思维链，保证教育任务中的逻辑连贯性（Jia, 2025）和可解释性：1.明确错误类型；2.RAG 检索内容；3.确定引导策略；4.生成符合苏格拉底式教学的回复文本。

4.实验设计与初步验证

本研究在某初中六年级开展了等组前后测准实验设计。对照组采用传统教学模式，即教师讲授、巡视答疑，学生自主课堂活动；实验组主要使用苏格拉底式智能体，当学生遇到认知困难时，由 AI 助教构建苏格拉底式引导，学生仅在必要时求助教师。具体过程如图 2 所示。

本研究以初中六年级信息科技的“数据运算有逻辑”单元为例，结合《义务教育信息科技课程教学指南 过程与控制 六年级全一册》，采用了一线教师精心撰写的教学设计、PPT 及学生活动单，据此构建了 RAG 向量知识库，以确保 LLM 回复与课时教学内容一致。

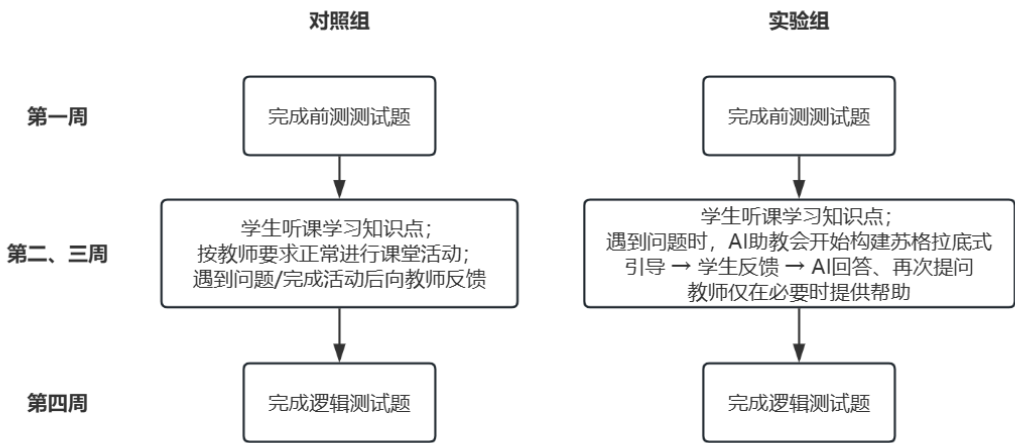


图 2 实验实施过程

课堂活动中，系统实时引导实验班学生的学习行为。以下为对话示例：

表 1 苏格拉底式教学与交互

交互步骤	对话内容与行为	系统分析
学生行为	设置判断条件为“光线强度> 600”；发现程序运行效果与预期不符，但反复修改其他代码积木。	缺乏对环境数值的感知，填入无效数值。
系统诊断	检测学习者认知状态。	盲目试错。
系统引导	“我看到你把阈值设为了 600。在当前的教室环境下，你能先读取一下光线传感器的实时数值是多少吗？”	明晰问题：引导学生关注实时数据。
学生反馈	（操作后）“是 350。”	获取真实环境数据。
系统引导	“很好！当前教室里的光线强度是 350lux，而程序的判断条件是‘光线强度>600’，那这个条件能满足吗？你要怎样调整阈值才能让灯在白天熄灭呢？”	问题拆解：引导学生建立“环境值”与“阈值”的比较关系。

实验结束后，对两组学生进行逻辑测试。逻辑测试基于《过程与控制》教学指南，涵盖“数据运算”单元的核心知识点，以确保测试工具的内容有效性。舍弃无效测试数据后，对测试结

果进行独立样本 t 检验。结果表明，实验组和对照组的后测成绩有显著性差异 ($p < 0.05$)，这表明苏格拉底式教学有效地促进了学生对逻辑运算的理解。

表 2 逻辑测试成绩对比

组别	有效样本 (N)	均值 (Mean)	标准差 (SD)	P 值
实验组	38	88.5	7.2	0.032
对照组	36	76.2	9.4	

* $p < 0.05$ *

此外，本研究定义“提交频次”为学生在课堂活动中向智能体提交代码的次数；“单次修改成功率”为“修正错误并使程序运行通过”的动作占总提交次数的比例。通过分析学生的编程行为，发现实验组在提交频次上低于对照组，但单次修改的成功率提升明显。这也说明系统有效引导学生进行修正与反思，初步实现了学生从“机械试错”向“反思对话”的转变。

5. 结论与展望

本研究想要解决初中信息科技教学中，学生在课堂活动时进行“机械试错”的问题。以上数据显示，实验组学生在逻辑测试中取得较好成绩，提交代码的频率降低但成功率反而上升。这表明通过提示词和思维链，智能体能够转变为进行苏格拉底式教学的导师，促进学生进行思考与反思，提高教学成效。此外，系统能够对“阈值设定”等教学难点有效引导，将复杂的逻辑问题转化为层层递进的对话探究过程，显著提升了学生的概念理解与知识迁移能力。

然而，由于实验条件限制，研究时间较短且规模较小；当前系统主要依赖文本和代码交互，在多模态诊断方面存在不足。未来，研究可扩展至覆盖全学段的广泛群体，以验证系统构建的普适性；引入多模态大模型，使系统能够截图理解程序积木和代码，更加深入地探索苏格拉底式引导和教学在初中信息科技教学中的应用。

参考文献

- 张誉月, & 贾积有. (2025, June). 利用大语言模型多智能体探索复杂推理数学题的苏格拉底式教学方法. In *The Global Chinese Conference on Computers in Education (GCCCE) Proceedings* (Vol. 2025, pp. 840-847).
- Bjork, E. L., & Bjork, R. A. (2011). Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning. *Psychology and the real world: Essays illustrating fundamental contributions to society*, 2(59-68), 56-64.
- Dong, Y., Marwan, S., Catete, V., Price, T., & Barnes, T. (2019, February). Defining tinkering behavior in open-ended block-based programming assignments. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1204-1210).
- Jia, J., Wang, T., Zhang, Y., & Wang, G. (2024). The comparison of general tips for mathematical problem solving generated by generative AI with those generated by human teachers. *Asia Pacific Journal of Education*, 44(1), 8-28.
- Jia, L., Qi, C., Wei, Y., Sun, H., & Yang, X. (2025). Fine-Tuning Large Language Models for Educational Support: Leveraging Gagne's Nine Events of Instruction for Lesson Planning. arXiv preprint arXiv:2503.09276.
- Kao, S., Grant, P., & Woltering, S. (2025). Socratic AI in K-12 Science Classrooms: Effects on Critical Thinking, Motivation, and Self-Regulation in a Randomized Controlled Trial.

- Ma, H., Wang, C., Zhu, H., Yang, S., Zhang, X., & Zhang, X. (2024, March). Enhancing cognitive diagnosis using un-interacted exercises: A collaboration-aware mixed sampling approach. In Proceedings of the AAAI conference on artificial intelligence (Vol. 38, No. 8, pp. 8877-8885).
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2010, August). Learning computer science concepts with scratch. In Proceedings of the Sixth international workshop on Computing education research (pp. 69-76).
- Planning, L. B. M. T. Instruct, Not Assist: LLM-based Multi-Turn Planning and Hierarchical Questioning for Socratic Code Debugging.