

基于 GenAI 的结对编程策略对初中生计算思维态度和编程自我效能感的影响研究

A study of the effects of GenAI-based pair programming strategy on middle school students' attitudes toward computational thinking and programming self-efficacy

王千姿^{1*}, 孟晓培¹

¹温州大学 教育学院

*wqz2316108432@163.com

【摘要】在数字化时代，计算思维已成为 K-12 阶段教育的重要议题。针对当前中小学编程教学面临的教学方法单一、学生编程信心不足等问题，本研究将 GenAI 技术结合结对编程策略，将温州市某学校七年级三个班的学生分为实验组 1（P-GenAI 编程学习方法）、实验组 2（C-GenAI 编程学习方法）和对照组（常规编程学习方法）。经过 10 周的教学实验，采用 Mann-Whitney U 检验对三组学生的计算思维态度和编程自我效能感进行了比较分析，结果显示，虽然三组学生在计算思维态度和编程自我效能感上均无显著差异，但实验组 1 的学生在各维度上的表现普遍优于实验组 2 和对照组，呈现出积极的趋势。基于此，研究提出了强化编程韧性、突破人机协作瓶颈以及构建多智能体协同生态等建议，以期为中小学编程教育提供科学依据和实践指导。

【关键词】生成式人工智能；GenAI；结对编程；计算思维态度；编程自我效能感

Abstract: This study explores the integration of GenAI technology with pair programming in K-12 education to address issues like limited teaching methods and students' lack of programming confidence. It involved three seventh-grade classes in Wenzhou, divided into three groups: Experimental Group 1 (P-GenAI programming), Experimental Group 2 (C-GenAI programming), and a control group (traditional programming). After a 10-week teaching experiment, the results showed no significant differences in computational thinking attitude or programming self-efficacy among the groups. However, Experimental Group 1 generally performed better in all aspects, suggesting positive trends. The study recommends enhancing programming resilience, overcoming human-computer collaboration challenges, and building a collaborative multi-intelligence ecosystem to improve programming education.

Keywords: Generative artificial intelligence, GenAI, Pair programming, Computational thinking attitudes, Programming self-efficacy

1. 引言

计算思维(Computational Thinking)作为数字时代公民必备的核心素养，已成为全球教育研究的重要议题，并逐渐向 K-12 阶段延伸(孟鸿伟, 2024)。近年来世界各国特别是美国、英国、新西兰等国家，已经将计算思维的培养纳入人才培养计划和课程体系(孙丹&李艳, 2019)。国内教育也在“双减”政策的推动下，积极推动中小学教育从单一的知识传授转向素养导向的全面发展，《义务教育信息科技课程标准（2022 年版）》中就明确提出计算思维是学科核心素养的关键要素。在诸多提升儿童计算思维的方法中，编程教育以其特有的潜能受到广泛关注(陆霞&董永权, 2024)，也逐步被列为我国中小学信息技术课程的核心内容。然而，当前中小学编程教学仍面临诸多挑战，如教学方法单一、学生编程信心不足以及教师忽视形成性评价(李彤彤 et al., 2022)，这些问题在一定程度上影响了编程教育的教学效果。随着生成

式人工智能(Generative Artificial Intelligence, 简称 GenAI)的快速发展, 其在编程教育中的独特优势也逐渐突显。GenAI 不仅能提供即时的智能反馈, 还能辅助学生进行代码理解、调试与优化, 从而提升学习效果和编程体验(Yilmaz, 2023)。

在此背景下, 本研究以 GenAI 支持的编程学习策略为切入点, 设计并比较三种不同学习方法对学生计算思维态度与编程自我效能感的影响。研究选取温州市某中学七年级三个班共 104 名学生为研究对象, 随机分为两个实验组和一个对照组。实验组 1 采用 GenAI 支持下融入结对编程策略的编程学习方法, 简称为 P-GenAI(Pair Programming-Generative Artificial Intelligence), 即在学生两两结对的过程中, GenAI 作为智能伙伴提供辅助建议与代码反馈; 实验组 2 采用 GenAI 支持下的常规编程学习方法, 简称为 C-GenAI(Conventional - Generative Artificial Intelligence), 即学生个体在 GenAI 的智能引导下完成编程任务; 对照组采用传统的编程学习方法, 独立完成编程任务, 简称为 C-C(Conventional - Conventional)。本研究通过对比三组学生在实验前后的表现变化, 探讨不同编程学习策略对学生计算思维相关非认知变量的影响, 以期为中小学编程教学的发展提供实践指导。

2.文献综述

2.1.计算思维态度与编程自我效能感的相关研究

计算思维的具体内涵和维度尚未完全统一。在诸多定义中, 计算思维三维框架(Brennan & Resnick, 2012)基于可视化思维和过程思维, 在教育实践中有较强的可操作性(Zhong et al., 2016), 而且计算概念, 计算实践和计算观念也能很好地对照教学的三维目标(王旭卿, 2014)。在计算思维三维框架的基础上, 郁晓华等(2020)界定了计算观念(具备以计算的视角洞察与理解自身和周围世界的良好品质)的范畴。从这一定义出发, 计算观念可以从计算思维态度和编程自我效能感两个方面来观测。其中, 计算思维态度(Computational Thinking Attitudes)作为计算思维的非认知组成部分, 已有研究通常强调个体在面对复杂问题时所表现出来的信心、坚持性和不确定性的容忍度等特质。Sun 等人(2022)在研究中使用编程态度量表来评估学生的编程态度, 结果表明, 学生的编程态度显著预测了他们的计算思维技能。此外, Zapata-Cáceres 等人(2022)在研究教师对计算思维的态度和自我评估时, 强调了教师在教学中对计算思维重要性的认知和态度。自我效能感(Self-efficacy)的培养也被证实在促进编程学习中具有重要作用。Bandura(1997)提出, 自我效能感是个体对自己完成某一任务能力的主观判断, 直接影响其行为选择、努力程度与应对策略。自我效能感在计算机领域的应用时, 其表现为计算机编程自我效能感(Programming Self-Efficacy)。编程自我效能感是指个人有能力在计算机领域发挥作用的信念。Ramalingam 等人(1998)将编程自我效能分为四个维度, 即独立和毅力、复杂编程能力、自我调节能力和简单编程能力。已有研究表明, 计算思维态度和自我效能感不仅对学生的编程学习成效具有显著预测作用, 也对其计算思维的持续发展构成支撑。例如, Durak(2019)发现, 在编程方面表现出色的学生既有较高的自我效能感水平, 也有较高的计算思维技能。因此, 对学生的计算思维态度和自我效能感进行探究, 有利于使编程教学达到更理想的效果。

2.2.GenAI 在编程教育中的应用研究

近年来, 随着预训练技术的发展和计算硬件的提升, 生成式人工智能取得了突破性进展, 为编程教育提供了新的可能性。GenAI 能够通过自然语言理解、代码生成、实时交互和智能反馈等方式辅助编程学习(Aljanabi, 2023), 提高学生的编程能力和学习效率(Piccolo et al., 2023)。例如, Yilma(2023)研究表明 ChatGPT 纳入的编程中, 可以为学生提供关于编程主题

的清晰解释、编程示例以及应用程序，提高学生的认知能力，减少在编码任务上花费的时间；Piccolo 等人(2023)对来自生物信息学入门课程的 184 个编程练习进行了实验，结果表明，ChatGPT 第一次尝试就正确解答了 139 个练习(75.5%)。经过几次尝试和自然语言反馈之后，ChatGPT 的正确率达到了 97.3%；李秀(2024)结合布卢姆教育目标分类理论，设计了一套生成式人工智能应用于编程教学的“三学”策略，展示了生成式人工智能在编程教学中的教育价值。此外，GenAI 还能通过自动化调试功能，帮助学生发现和修正错误，提高其问题解决能力。尽管 GenAI 在编程教育中的应用展现出明显优势，但也有研究者担忧长期依赖 GenAI 可能会削弱学生的批判性思考、探索、验证和主动总结的能力(Yu, 2023)，容易产生“机械接受”的现象，使其在问题求解过程中过度依赖智能工具，从而影响深层次的知识建构。因此，在教学设计中，需要合理规范 GenAI 的使用方式，强调其作为辅助工具的角色，同时结合教师指导与策略支持，鼓励学生在使用 GenAI 工具的过程中主动思考与反思，从而实现计算思维与编程能力的同步发展。

2.3. 结对编程中计算思维的相关研究

结对编程(Pair Programming)是一种合作学习策略，在编程教育中被广泛应用。该模式最早由 Williams 和 Kessler (2000)提出，指两名学生共同使用一台计算机，其中一人作为“驾驶员”(Driver)编写代码，另一人作为“观察员”(Navigator)进行代码审查和问题思考，并定期交换角色。结对编程的这种合作模式既能通过实时协作即时检测和纠正代码，从而减少缺陷并改进整体代码结构，也能在交流合作的过程中进行知识共享，提高学生的参与度。研究表明，与单独的编程任务相比，结对编程可以导致更高水平的学生满意度和减少的挫折感的报告(Wei et al., 2021)。在传统的教育环境中，结对编程的实施也面临诸多挑战。一是学生间的个体差异可能导致学习不均衡，影响合作质量。为此，研究者提出了多种配对策略，如基于性别(Choi K S, 2015)、学习风格(Demir, Ö. & Seferoglu, S. S, 2018)和伙伴关系(李彤彤 et al., 2022)等方式进行分组，以优化结对编程的教学效果。二是难以安排同伴之间的协作会话。尤其是在高等教育环境中，学生往往有不同的时间表和约定(Govender & Grayson, 2006)。考虑到传统结对编程存在的局限性，研究者开始将 AI 引入结对编程的实践中，探索人机协作的新范式。如 Fan 等人(2025)对 234 名学习 Java 网络应用程序开发课程的本科生进行了准实验研究，结果显示，虽然人工智能辅助结对编程提高了学习动机、减少了焦虑并改善了成绩，但它并不能完全与通过人-人结对编程实现的协作深度和社会存在相媲美。因此，如何在保证人工智能辅助结对编程优势的情况下，科学地提高学生在合作过程中的学习体验，是亟待解决的问题。

综上所述，计算思维态度与编程自我效能感作为计算思维的重要非认知维度，在提升学生编程学习动机与成效方面发挥着关键作用。GenAI 的发展为编程教学提供了便利的智能支持，通过自然语言交互与即时反馈降低了编程门槛，有助于增强学生的学习信心与持续投入。但同时也引发了对学生批判性思维削弱与过度依赖技术的担忧。将 GenAI 融入结对编程，有望在保留人际合作优势的基础上，拓展人机协作的深度与广度，提升学习适应性与支持度。当前针对该融合情境下学生非认知因素变化的实证研究较少，因此，深入探究 GenAI 支持下结对编程对学生非认知因素的影响，有助于拓展相关研究视角，丰富智能时代背景下编程教育的理论基础与实践路径。

3. 研究设计

3.1. 研究对象

本研究选取了温州市某学校七年级三个班的学生为研究对象，随机设置实验组和对照组。其中实验组 1 采用 GenAI 支持下融入结对编程策略的编程学习方法，简称为 P-GenAI 编程学习方法(Pair Programming-Generative Artificial Intelligence)；实验组 2 采用 GenAI 支持下的常规编程学习，简称为 C-GenAI 编程学习方法(Conventional - Generative Artificial Intelligence)；对照组采用的常规编程学习方法，简称为 C-C 编程学习方法(Conventional - Conventional)。本研究以初中生的计算思维态度和自我效能感为因变量。为了保证实证研究的科学性与严谨性，本研究三组的授课教师、教学内容、教学环境、教学时长和样本年级均相同。此外，在实验前，实验组 1、实验组 2 和对照组三个组的计算思维态度和自我效能感成绩均无显著差异。

表 1 研究对象统计

班级	组别	编程学习方法	N	男生	女生
七（5）班	实验组 1	P-GenAI	34	19	15
七（8）班	实验组 2	C-GenAI	33	18	15
七（7）班	对照组	C-C	37	23	14

3.2. 研究工具

本研究采用计算思维态度量表由 Korkmaz 等人(2017)编制，共 8 道题，涵盖表达和创造力两个指标；编程自我效能感量表则引用 Volkan Kukul 等人(2017)编制的《计算机编程自我效能感量表》，共 20 道题，涉及独立性与毅力、复杂编程、自我调节及简单编程四个维度。为确保测量工具的可靠性，本研究对两个量表进行了信度检验，结果显示计算思维态度量表的 Cronbach’s α 系数为 0.875，编程自我效能感量表的 Cronbach’s α 系数为 0.860，表明两个量表均具有良好的内部一致性与测量稳定性。除此之外，研究者根据学生的课堂表现，在实验组中从不同性别、不同知识水平的学生中选取学生进行访谈，访谈大纲改编自 Fang et al.(2022)进一步为假设结论提供一定程度的参考。

3.3. 教学过程

3.3.1. 教学平台

目前，中小学中编程教学大多借助 Python、Scratch、编程猫等软件。文本编程工具如 Python、MaLT2 等在语法上更接近自然语言，易于阅读和维护(陆霞&董永权, 2024)；而图形化编程工具如 ScratchJr、Alice 等则只需要像摆积木一样拖放编程模块，就可以为作品赋予新的能力和实现更多创意玩法，在避开复杂语法的同时完美保留了编程思维。



图 1 Paracraft 软件用户编辑界面

实验教学选用的教学软件是 Paracraft 编程软件，用户编辑界面如图 1 所示。Paracraft 是一款免费开源的 3D 动画与编程创作软件，具有多语言编程环境，支持图形化、NPL、Python、C++ 等语言，软件中的图形化编程和文本编程还可以无缝切换，通过编写程序，能够控制动画，实现复杂的程序效果。Paracraft 相比其他编程软件创造了一个符合编程习得的环境，先建造、后动画、最后添加逻辑，学生学习兴趣浓厚；并且具备“多人联网”功能支持结对策略的开展（如图 2），适用于初中生的编程教学。



图 2 Paracraft 软件的“多人联网”功能

3.3.2.GenAI 学习工具

本研究选用的 GenAI 工具是 Keepwork，其深度融合 Paracraft 编程环境，并针对编程学习提供专业化的知识库支持，在一定程度上能弥补 GenAI 工具因通识性互联网数据训练而导致的知识偏差。具体来说，Keepwork 提供了多个版本(免费版、Turbo 和 pro)，满足不同用户的需求，能够在 Paracraft 编程环境中为学生实时解答疑惑，优化代码，提供更有针对性的指导和反馈。



图 3 Keepwork 平台主界面

3.3.3.教学计划与内容

实验周期共 10 周，分为准备阶段（2 周）、学习阶段（6 周）、后测阶段（2 周）。教学内容主要参考了《Paracraft 编程入门》（哈尔滨工业大学出版社）和《Paracraft 青少年 3D 动画编程入门》（清华大学出版社）两本教材，以及 Paracraft 编程软件内包含的课程、资源以及优秀作品等，根据学生现有知识水平，选取了五子棋和开心大闯关两个项目。五子棋项目涉及坐标定位、事件处理和游戏规则逻辑的编写，而飞行的小鸟项目则主要涉及物理模拟、碰撞检测和角色控制，让学生在掌握基本的编程技能同时提升逻辑思维和问题解决能力。

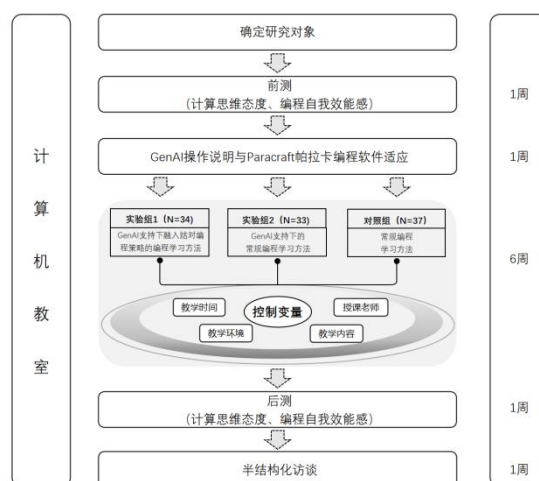


图 4 实验流程

4 研究结果与分析

4.1. 量化数据分析

本研究对计算思维态度和编程自我效能感数据进行了 Shapiro-Wilk 检验，发现数据非正态分布，因此使用 Mann-Whitney U 检验来比较三组学生在计算思维态度和编程自我效能感上的差异。所有检验均使用 SPSS 26.0 和 jamovi2.3.28 软件完成。

表 2 为实验组 1 和实验组 2 Mann-Whitney U Test 的分析结果。计算思维态度后测的 Mann-Whitney U Test($U=366, p=.291>.05, Z=-1.063$)和编程自我效能感后测的 Mann-Whitney U Test($U=386, p=.459>.05, Z=-.748$)结果表明，实验组 1 和实验组 2 在计算思维态度和编程自我效能感后测上均不存在显著差异。此外，本研究还通过 Mann-Whitney U 检验分析了计算思维态度和编程自我效能感的各个维度。结果显示，计算思维态度和编程自我效能感各维度均无显著差异，但在中位数(Md)上实验组 1 均大于实验组 2。这些结果表明，实验组 1 和实验组 2 在计算观念各维度的差异上并不显著($p>0.05$)，效应量较小，但实验组 1 在各维度上的表现存在普遍优于实验组 2 的趋势，该趋势并不排除偶然性因素的影响。

表 2 实验组 1 和实验组 2 计算观念水平后测分析结果

Variable	Group	N	Mean	SD	Md	U	p	r	Z
计算思维态度	实验组 1	34	3.68	1.11	4.00	366	0.291	0.160	-1.063
	实验组 2	33	3.32	1.10	3.13				
表达	实验组 1	34	3.76	1.13	4.00	356	0.228	0.182	-1.213
	实验组 2	33	3.34	1.19	3.00				
创造力	实验组 1	34	3.60	1.12	3.75	374	0.353	0.140	-.937
	实验组 2	33	3.30	1.11	3.25				
自我效能感	实验组 1	34	3.60	1.11	3.40	386	0.459	0.113	-.748
	实验组 2	33	3.32	1.03	3.20				
独立和毅力	实验组 1	34	3.66	1.17	3.60	376	0.363	0.137	-.917
	实验组 2	33	3.34	1.09	3.20				
复杂编程	实验组 1	34	3.68	1.12	3.50	364	0.273	0.163	-1.105
	实验组 2	33	3.30	1.01	3.00				
自我调节	实验组 1	34	3.53	1.14	3.33	384	0.434	0.118	-.790
	实验组 2	33	3.28	1.01	3.00				
简单编程	实验组 1	34	3.56	1.07	3.20	411	0.712	0.056	-.377
	实验组 2	33	3.35	1.16	3.20				

表 3 为实验组 1 和对照组 Mann-Whitney U Test 的分析结果。计算思维态度后测的 Mann-Whitney U Test($U=281, p=.248>.05, Z=-1.165$)和编程自我效能感后测的 Mann-Whitney U Test($U=303, p=.454>.05, Z=-.759$)结果表明，实验组 1 和对照组在计算思维态度和编程自我效能感后测上均不存在显著差异。此外，本研究还通过 Mann-Whitney U 检验分析了计算思维态度和编程自我效能感的各个维度。结果显示，计算思维态度和编程自我效能感各维度均

无显著差异，但在中位数(Md)上实验组 1 均大于对照组。这些结果表明，实验组 1 和对照组在计算观念各维度的差异上并不显著($p>0.05$)，效应量较小，但实验组 1 在各维度上的表现存在普遍优于对照组的趋势，该趋势并不排除偶然性因素的影响。

表 3 实验组 1 和对照组计算观念水平后测分析结果

Variable	Group	N	Mean	SD	Md	U	p	r	Z
计算思维态度	实验组 1	34	3.68	1.11	4.00	281	0.248	0.187	-1.165
	对照组	37	3.21	1.24	3.25				
表达	实验组 1	34	3.76	1.13	4.00	272	0.189	0.212	-1.323
	对照组	37	3.24	1.23	3.25				
创造力	实验组 1	34	3.60	1.12	3.75	271	0.182	0.214	-1.343
	对照组	37	3.18	1.28	3.00				
自我效能感	实验组 1	34	3.60	1.11	3.40	303	0.454	0.122	-.759
	对照组	37	3.33	1.12	3.15				
独立和毅力	实验组 1	34	3.66	1.17	3.60	303	0.449	0.122	-.766
	对照组	37	3.39	1.28	3.40				
复杂编程	实验组 1	34	3.68	1.12	3.50	302	0.437	0.125	-.786
	对照组	37	3.35	1.24	3.25				
自我调节	实验组 1	34	3.53	1.14	3.33	288	0.301	0.167	-1.043
	对照组	37	3.20	1.15	3.20				
简单编程	实验组 1	34	3.56	1.07	3.20	313	0.566	0.093	-.582
	对照组	37	3.41	1.04	3.00				

表 4 为实验组 2 和对照组 Mann-Whitney U Test 的分析结果。计算思维态度后测的 Mann-Whitney U Test($U=313, p=.711>.05, Z=-.380$)和编程自我效能感后测的 Mann-Whitney U Test($U=331, p=.971>.05, Z=-.046$)结果表明，实验组 2 和对照组在计算思维态度和编程自我效能感后测上均不存在显著差异。此外，本研究还通过 Mann-Whitney U 检验分析了计算思维态度和编程自我效能感的各个维度。结果显示，计算思维态度和编程自我效能感各维度均无显著差异，但在中位数(Md)上实验组 2 的创造力、编程自我效能感、简单编程大于对照组。这些结果表明，实验组 2 和对照组在计算观念各维度的差异上并不显著($p>0.05$)，且效应量较小。

表 4 实验组 2 和对照组计算观念水平后测分析结果

Variable	Group	N	Mean	SD	Md	U	p	r	Z
计算思维态度	实验组 2	33	3.32	3.13	1.10	313	0.711	0.062	-.380
	对照组	37	3.21	3.25	1.24				
表达	实验组 2	33	3.34	1.19	3.00	324	0.876	0.029	-.176
	对照组	37	3.24	1.23	3.25				
创造力	实验组 2	33	3.30	1.11	3.25	299	0.528	0.103	-.641
	对照组	37	3.18	1.28	3.00				
自我效能感	实验组 2	33	3.32	1.03	3.20	331	0.971	0.008	-.046
	对照组	37	3.33	1.12	3.15				
独立和毅力	实验组 2	33	3.34	1.09	3.20	327	0.904	0.021	-.130
	对照组	37	3.39	1.28	3.40				
复杂编程	实验组 2	33	3.30	1.01	3.00	325	0.881	0.026	-.160
	对照组	37	3.35	1.24	3.25				
自我调节	实验组 2	33	3.28	1.01	3.00	311	0.682	0.068	-.419
	对照组	37	3.20	1.15	3.00				
简单编程	实验组 2	33	3.35	1.16	3.20	331	0.963	0.009	-.056
	对照组	37	3.41	1.04	3.00				

4.2. 访谈结果分析

为了更加全面地了解不同编程策略对学生计算思维态度与编程自我效能感的影响，本研究在实验结束后随机访谈了实验组 1 与实验组 2 的部分学生，进一步挖掘学生在学习过程中的真实体验和主观感受。访谈结果显示，实验组 1(P-GenAI)中的多数学生认为，与同伴协作加上 GenAI 的辅助能够提升他们面对任务时的信心，减少编程任务带来的焦虑。一位学生提到：“当我不太懂的时候，搭档可以讲给我听，不懂的地方我们也可以一起问 AI，我觉得更轻松也更有动力。”这种编程协作模式在一定程度上增强了学生在学习过程中的参与感

和安全感，帮助他们更持久地投入到编程任务中，尽管在量化数据中未体现出显著差异，但访谈反馈反映出其对非认知因素的潜在正向作用。

相比之下，实验组 2(C-GenAI)的学生在学习过程中虽能得到 AI 的技术支持，但部分学生表示“有时候 AI 说的我还是看不太懂”，或“没有人讨论的话，我不太确定是不是理解对了”。这在一定程度上说明，单人使用 GenAI 虽具效率优势，但缺乏协同交流与思维碰撞，可能影响其在认知建构与态度养成方面的积极作用。

访谈数据与量化结果相辅相成，实验组 1 虽然未在统计意义上显著优于其他组，但在学生主观体验和非认知反馈中体现出更高的积极性与参与感。这一结果表明，在注重 AI 工具支持的同时，我们仍需关注人际协作的情感价值和交互作用，为学生提供更加和谐的智能学习环境。

5.研究总结与建议

本研究基于生成式人工智能支持下的结对编程教学，探讨其对初中生计算思维态度与编程自我效能感的影响。通过准实验设计比较三组学生在实验前后的变化趋势，结果表明：尽管实验组在量化数据上未表现出统计意义上的显著差异，但从中位数和访谈数据来看，实验组 1 在学生信心、协作参与和主动调试等方面表现出更积极的态度和感受。研究结果表明，GenAI 在辅助编程教学的同时，应关注学生非认知因素的变化，并在教学设计中寻求技术支持与人际交互的平衡。因此，在未来的教学实践中，应更加重视技术支持与人际互动的有机融合，兼顾学习效率与心理发展，以推动计算思维的深入发展。基于本研究的发现，提出以下三点建议：

5.1.强化编程韧性，筑牢计算思维的心理根基

编程韧性(Programming Resilience)是计算思维非认知维度培养的关键要素，强调学习者在编程过程中面对困难时的坚持与适应能力。王佑镁等(2023)基于“韧性”发展的 7C 能力结构模型与编程课程的实际特点，构建了编程韧性的“4C”能力结构模型，这为数字时代培养计算思维提供了新方向。在 GenAI 支持的结对编程教学中，学生在协作过程中可能因认知冲突和编程错误产生挫败感，而 GenAI 的即时反馈和自动补全虽然能够降低学习门槛，但若未能合理引导，可能导致学生过度依赖技术支持，从而削弱其独立解决问题的能力。因此，教学应从多个层面强化学生的编程韧性，帮助学生建立对挑战的积极认知。首先，采用任务递进式设计，让学生在挑战中不断适应编程逻辑，培养面对错误的耐挫力；其次，营造鼓励试错和反思的课堂氛围，要求学生记录调试过程，形成有效的问题解决策略；最后，结合项目式学习与竞赛机制，让学生在团队协作与竞争压力下不断提升自信和应变能力。通过以上策略，学生能够在真实的问题解决中逐步培养心理耐力，培养积极的学习态度与稳定的计算思维心理基础。

5.2.突破人机协作瓶颈，塑造批判性智能共创模式

GenAI 的融入让结对编程不再仅限于人与人之间协作，然而在人机协作过程中学生应该如何保持批判性思维，避免机械接受 AI 的输出内容，这成为当下智能时代亟待解决的关键问题。批判性思维强调学习者能够独立思考、分析和评估 AI 生成的代码，而非简单依赖 AI 提供的方案。在此基础上，可引入对弈性训练(Adversarial Training)模式(刘邦奇 et al., 2025)，让学生在代码编写过程中与 GenAI 进行“博弈”，即学生主动识别、挑战 AI 代码中的潜在漏洞，以提升代码优化与反思能力。此外，逆向工程(Rreverse Engineering)作为计算思维训练的高级形式，能够帮助学生透过代码表象理解其底层逻辑(翟雪松 et al., 2024)。在教学设计上，

可提供 GenAI 生成的代码, 让学生进行结构解析、错误修正与逻辑优化, 同时鼓励学生基于已有代码进行拓展和重构。为增强教学实效, 还可结合“代码挑战赛”或“人机对抗编程”模式, 要求学生在限定时间内优化 GenAI 生成的代码, 以此培养精确表达、逻辑推理和问题拆解能力。通过批判思维引导、对弈性训练及逆向工程策略的融合, 学生不仅能够提升计算思维的精细化程度, 同时也能在与 AI 的协作中培养更强的创造力与问题解决能力。

5.3. 构建多智能体协同生态, 拓展智能编程的教育边界

单一智能体的介入能够提升编程效率, 但无法满足复杂学习场景的多样化需求, 因此, 构建多智能体生态成为增强人机协作效能的重要方向(翟雪松 et al., 2024)。当前的 GenAI 主要用于代码生成和错误检测, 而未来的编程教育应进一步拓展其应用场景, 使多个智能体在学习过程中各司其职、协同运作。例如, 可引入三层智能体架构: 第一层为代码生成智能体, 如 ChatGPT, 负责提供基础代码; 第二层为优化与调试智能体, 专注于检测语法错误、逻辑漏洞与性能优化; 第三层为学习支持智能体, 结合数据分析技术, 为学生提供个性化学习建议。这种多智能体的协同生态模拟了真实的软件开发环境, 使学生在任务驱动式学习过程中, 学会合理调配不同智能体资源, 提升任务拆解、调试优化及团队协作能力。此外, 多智能体协作还能支持学习者的动态反馈机制, 根据学生的学习数据调整任务难度, 实现个性化编程教育。通过构建多智能体生态系统, 不仅能够提升人机协作的整体效能, 还能为学生提供更具交互性、适应性与智能化的学习体验, 从而为未来智能编程教育的深度发展奠定基础。

参考文献

- 白雪梅 & 顾小清.(2019).K12 阶段学生计算思维评价工具构建与应用.中国电化教育,(10),83-90.
- 傅骞,解博超 & 郑娅峰.(2019).基于图形化工具的编程教学促进初中生计算思维发展的实证研究.电化教育研究,40(04),122-128.doi:10.13811/j.cnki.eer.2019.04.016.
- 李彤彤,郝晴,文雨,苏新玉,方玉华 & 刘佳旋.(2022).基于学习风格和伙伴关系的配对编程对小学生计算思维的影响研究.远程教育杂志,40(03),105-112.doi:10.15881/j.cnki.cn33-1304/g4.2022.03.007.
- 李锋,程亮 & 顾小清.(2022).计算思维学业评价的内容构建与方法设计——文献比较研究的视角.中国远程教育,(02),65-75+77.doi:10.13541/j.cnki.chinade.2022.02.008.
- 刘邦奇,聂小林,王亚飞,袁婷婷,赵子琪 & 张国强.(2025).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.中国电化教育,(03),61-70.
- 陆霞 & 董永权.(2024).面向计算思维培养的儿童编程教育研究综述.中国教育信息化,30(10),109-118.
- 孟鸿伟.(2024).面向数字化未来的“计算思维”.中国教育信息化,30(02),3-12.
- 任友群,隋丰蔚 & 李锋.(2016).数字土著何以可能?——也谈计算思维进入中小学信息技术教育的必要性和可能性.中国电化教育,(01),2-8.
- 孙丹 & 李艳.(2019).国内外青少年编程教育的发展现状、研究热点及启示——兼论智能时代我国编程教育的实施策略.远程教育杂志,37(03),47-60.doi:10.15881/j.cnki.cn33-1304/g4.2019.03.005.
- 王靖,马志强,刘亚琴 & 杜鸿羽.(2022).面向计算思维的可视化编程活动设计与应用.现代教育技术,32(09),55-63.
- 王佑镁,南希烜,李宁宇,尹以晴 & 柳晨晨.(2023).编程韧性: 数字时代计算思维培养的新议题.现代教育技术,33(02),14-23.

- 郁晓华 & 王美玲.(2020).计算思维培养之路还有多远?——基于计算思维测评视角.开放教育研究,26(01),60-71.doi:10.13966/j.cnki.kfjyyj.2020.01.007.
- 岳彦龙,张学军 & 梁屿藩.(2022).人工智能教学如何培养高中生的计算思维?——基于人工智能案例驱动的 Python 编程教学的实证研究.基础教育,19(01),74-84.
- 中华人民共和国教育部.义务教育信息科技课程标准(2022 年版)[M].北京:北京师范大学出版社,2022.4-11.
- 张进宝, (2020).国际计算思维挑战赛试题集锦[M].北京: 电子工业出版社.
- 翟雪松,张丽洁,夏亮亮,徐鑫 & 朱强.(2024).基于 GAI 的逆向工程教学思维在人机协作中的应用研究 —— 以 编 程 教 育 为 例 . 电 化 教 育 研 究 ,45(09),61-68.doi:10.13811/j.cnki.eer.2024.09.008.
- Aljanabi, M. (2023). ChatGPT: Future directions and open possibilities. *Mesopotamian journal of Cybersecurity*, 2023, 16-17.
- Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).
- Choi, K. S. (2015). A comparative analysis of different gender pair combinations in pair programming. *Behaviour & Information Technology*, 34(8), 825-837.
- Demir, Ö., & Seferoglu, S. S. (2021). The effect of determining pair programming groups according to various individual difference variables on group compatibility, flow, and coding performance. *Journal of Educational Computing Research*, 59(1), 41-70.
- Durak,H,Y,Yilmaz,F.G,K.,&Yilmaz,R.(2019). Computational thinking,programming self-efficacy,problem solving and experiences in the programming process conducted withrobotic activities.*Contemporary Educational Technology*,10(2),173-197.
- Fang, J. W., He, L. Y., Hwang, G. J., Zhu, X. W., Bian, C. N., & Fu, Q. K. (2023). A concept mapping-based self-regulated learning approach to promoting students' learning achievement and self-regulation in STEM activities. *Interactive Learning Environments*, 31(10), 7159-7181.
- Fan, G., Liu, D., Zhang, R., & Pan, L. (2025). The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: a comparative study with traditional pair programming and individual approaches. *International Journal of STEM Education*, 12(1), 16.
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in human behavior*, 72, 558-569.
- Kukul, V., Gökçearslan, Ş., & Günbatar, M. S. (2017). Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation and reliability. *Eğitim Teknolojisi Kuram ve Uygulama*, 7(1), 158-179.
- Piccolo, S. R., Denny, P., Luxton-Reilly, A., Payne, S., & Ridge, P. G. (2023). Many bioinformatics programming tasks can be automated with ChatGPT. *arXiv preprint arXiv:2303.13528*.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Yilmaz, R., & Yilmaz, F. G. K. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence*, 4, 100147.

Yu, H. (2023). Reflection on whether Chat GPT should be banned by academia from the perspective of education and teaching. *Frontiers in Psychology, 14*, 1181712.