

编程真的能促进儿童的个体发展吗? *

——基于 28 项实验和准实验研究的元分析

孙立会 胡琳琳

(天津大学教育学院, 天津 300350)

摘要: 编程作为儿童面向未来、适应未来并改造未来的基本素养, 不断得到各界的关注与重视, 但其究竟到底有多少教育价值, 众多学者开展了大量的实验及准实验研究, 但研究结果却不尽相同。“编程能否真正促进儿童的个体发展”这一问题仍处在左右摇摆的学术争议之中。基于此, 本研究采用元分析方法, 对国外 28 项编程影响儿童个体发展的实验与准实验研究进行量化分析。研究发现: (1) 整体而言, 纳入研究的合并效应量为 0.61, 处于中等偏上水平, 表明编程可以积极正向地促进儿童的个体发展; (2) 具体来看, 编程能够显著地积极促进学生的认知发展、行为发展以及情感发展; (3) 与小学及学前相比, 编程在中学阶段对儿童个体发展的作用效果更突出; (4) 与中期和长期编程活动相比, 短期编程活动对儿童个体发展的促进作用更为显著; (5) 编程教育在小规模样本中对儿童个体发展的作用效果相较于中、大规模样本更大; (6) 不同编程形式的作用效果不同, 其中不插电编程对儿童个体发展的作用更为显著。期望上述研究发现能为我国未来儿童编程的研究与实践提供借鉴。

关键词: 儿童编程; 个体发展; 元分析; 效应量

一、引言

面向智能时代, 日本、美国、新加坡、丹麦、芬兰等世界主要国家均率先出台相关政策普及编程教育, 推进基础教育中的编程跨学科整合(森秀樹, 杉澤学, 張海, 前迫孝憲, 2011; 石晋阳, 2018)。虽然国内儿童编程教育起步较晚, 但极具后发优势。国务院在 2017 年印发了《新一代人工智能发展规划的通知》(中华人民共和国中央人民政府, 2017), 提出“在中小学阶段设置人工智能相关课程, 逐步推广编程教育, 鼓励社会力量参与寓教于乐的编程教学软件、游戏的开发和推广”, 极大地推动了儿童编程教育市场的迅猛发展。2018 年的“两会”再次强调了人工智能给中国带来的历史机遇, 教育部也逐步将儿童编程纳入中小学基础必修课体系之中。随后, 浙江、山东、江苏、山西、广东、上海、天津等省市相继出台政策文件, 陆续实施课程改革, 让编程教育走进中小学课堂。但是, 政策在落地过程中的实际表现, 与初衷有所偏差。就目前形势来看, 编程教育或多或少地忽视了其本质的教育功能与目的, 反而被附加上了众多“目的与功利性”色彩: 从自主招生到高考再到出国乃至未来就业, 掌握编程技能使得儿童在竞争中更具优势。儿童编程终究是否符合教育规律? 它能否真正促进儿童的个体发展? 这既是众多家长及教育者焦虑和担忧的地方, 也是现阶段推广儿童编程教育亟待回答的重要问题。

为此, 众多学者进行了大量的实证研究, 但研究结果却不尽相同, 这导致儿童编程对个体发展的效果仍难以阐释。一部分研究指出, 儿童在编程活动中可获得知识的迁移(Clements & Meredith, 1993;

* 基金项目: 全国教育科学规划“十三五”国家青年课题“非计算机化与计算机化儿童编程教育的理论与实践研究”(CCA190261)。

Clements & Sarama, 1997; Palumbo, 1990)、思维的提升(Johnsongentile, Clements, & Battista, 1994; Lye & Koh, 2014; Moreno-León, Robles, & Román-González, 2016)、问题解决能力的培养(Bernardo & Morris, 1994; Falloon, 2016; Fessakis, Gouli, & Mavroudi, 2013)以及社交协作行为的习得(Popat & Starkey, 2019; Resnick et al., 2009)等诸多方面的发展。同时, Psycharis & Kallia(2017)通过对 66 名高中生进行分组对照实验研究的结果显示,参与程序设计课程的学生与未参与程序设计课程的学生在推理能力及自我效能感评价指标上存在显著差异; Bernardo(1994)和 Palumbo(1990)的早期研究测量了问题解决能力的发展,结果发现编程对数学问题解决能力的改善显著高于没有接受特定编程指导的控制组;在学前儿童的一项编程活动实验中发现,编程使得学前儿童解决问题变得更容易(Papadakis, Kalogiannakis, & Zaranis, 2016);将机器人技术和计算机编程融入儿童早期学习,采用故事排序任务进行干预并对参与者前后的排序技能进行评估后,发现儿童的排序能力显著增强(Kazakoff, Sullivan, & Bers, 2013)。然而,与上述研究结果相反的是,另一部分研究发现编程并没有显著促进儿童个体各个方面的发展。Falloon(2016)、Kalelioğlu & Gülbahar(2014)发现,在小学中进行 Scratch 教学的干预前后,学生的问题解决能力并无显著差异; Psycharis et al.(2017)虽在研究中发现将数学教学与计算机编程结合起来可以提高数学问题解决能力,但在统计学上这种改善并不比没有接受编程数学学习的控制组有明显的提高;同样,一项研究发现,参与编程活动的实验组女孩与未参与编程活动的对照组女孩的解决问题的成绩之间没有显著差异(Miller et al., 1988);甚至, Hayes & Stewart(2016)的研究比较了使用 Scratch 和使用派生关系响应(Derived Relational Responding, DRR)培训计划的学生的认知变化,并得出结论认为,使用 Scratch 的学生进步更小。

基于此,本研究采用元分析(Meta-analysis)的方法,从“儿童编程是否促进儿童的个体发展”这一视角出发,梳理了有关编程对个体发展影响的实证研究,并对其研究结果进行整理、分析、评估,得出编程对儿童个体发展的影响程度。并且,研究进一步分析了不同学段、编程形式、样本规模及实验周期等调节变量对编程影响儿童个体发展的差异,以期得到科学合理的结论,为后续推广儿童编程教育研究与实践提供可行性参考。

二、研究方法过程

(一) 研究方法

元分析(Meta-analysis)是一种依据特定主题,通过整合已有研究结果,运用统计学对多个独立研究的结果进行再分析的研究方法(Glass, 1976)。由于它在一定程度上可防止和纠正传统研究方法中存在的偏见与不足,能够得到更具普适性、规律性和科学性的研究结论(陈鹤阳,朝乐门,2015),因此逐渐成为一种被广泛认可和使用的量化研究方法。近年来元分析方法在教育领域也被广泛用以分析技术在教育教学应用中的有效性问题,且研究结论都具有一定的参考性(顾小清,胡梦华,2018;王翠如,胡永斌,2018;王雪,徐文文,高泽红,王志军,2019)。本文采用元分析方法对 28 项关于儿童编程对个体发展作用效果的实验及准实验研究进行整合、分析,旨在评估编程能否或是在多大程度上促进儿童个体发展。本研究以标准化平均差(Standardized Mean Difference, SMD)作为效应量,用以表征儿童编程对个体发展的影响程度。

(二) 研究过程

1. 文献检索

本研究选取 Web of Science、EI、EBSCOhost、ERIC、Elsevier、Springer、Wiley、Taylor 等大型数据库,检索类型设定为“期刊”检索,在高级检索入口下选择主题检索模式,且其他检索条件设为默认,检索框中输入检索公式“(students or child* or teenager)and(program* or code or LOGO or Scratch or unplugged)and(thinking or ability or influen*)and(research or study)”运行检索,并将所得文献导出汇总。在此基础上,依据第一轮检索所获取文献的参考文献采用“滚雪球”式的检索方式进行第二轮检索,以

期获得更多主题相关文献。我们将两轮文献检索所得数据汇总至 Excel 进行数据清洗，删除重复的文献，共计产生 322 篇可供进一步筛选分析的候选文献。

2. 文献遴选

文献纳入排除过程如图 1 所示。鉴于检索到的文献并不都符合此次元分析的要求，因此本研究制订了如下遴选标准：(1)文献主题应为编程对儿童个体发展的影响研究。“儿童”指的是联合国《儿童权利公约》所界定的 18 岁以下的每一个人(UNICEF, 1989)。因此，纳入文献研究对象年龄范围划定为 0~18 岁。(2)文献必须是一个实验或准实验研究。其中，单组实验需要有前后测的对照，双组实验必须包含一个在编程环境下教学或学习的实验组，以及与之相对应的其他环境下教学的对照组。(3)文献中应报告足够的数据信息(样本数量、均值)用以计算效应量 SMD。依据上述遴选标准，初步筛选的文献中共有 28 项独立研究符合要求被纳入分析(见表 1)，纳入的研究中共计产生 74 个可用于元分析的效应量(其中部分独立研究包括多个效应量)。

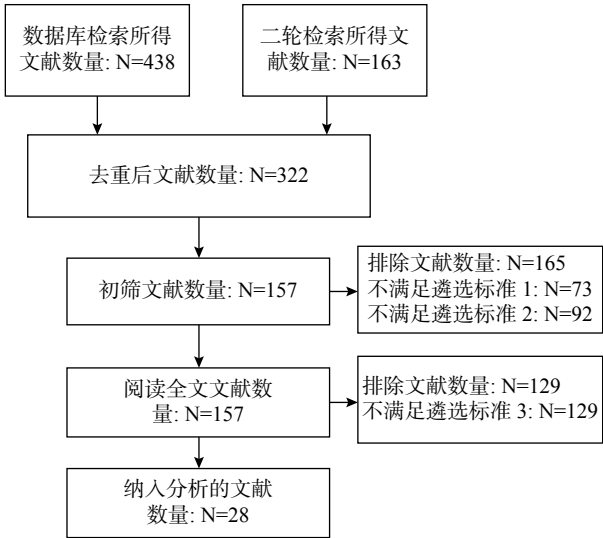


图 1 文献纳入排除流程图

表 1 入选元分析的 28 项独立研究

作者	年度	作者	年度	作者	年度
Fridin, M.	2014	Clements, D. H.	1986	Chou, P.	2019
Hatice, Y. D.	2018	Pea, R. D.	1984	Rose, S. P.	2017
Sáez-López, J.	2016	Stoeger, H.	2013	Jun, S.	2017
Cetin, I.	2016	Grover, S.	2015	Yadav, A.	2014
Berland	2015	Wang, X.	2017	Zhao, W.	2019
Hermans, F.	2017	Kalelioğlu, F.	2015	Papatga, E.	2016
Clements, D. H.	1990	Allsop, Y.	2019	Calao, L. A.	2015
Moreno-León, J.	2015	Kazakoff, E. R.	2014	Kalelioğlu, F.	2014
Corral, J. M. R.	2014	Román-González, M.	2018	Brackmann, C. P.	2017
Browning, S. F.	2017				

3. 特征值编码

在实施元分析前，需对用以分析的研究文献进行特征值编码。不同的研究往往含有不同的特征值，此次研究编码的内容主要包括作者信息、发表年份、实验时间、研究学段、样本规模、实验周期、编程形式以及测试类型等内容，具体编码说明如表 2 所示。综合来看，各研究的发表年份区间为

1984 年至 2019 年, 横跨儿童编程教育发展的各个周期。研究涉及儿童(0~18 岁)学习的各个学段; 实验样本规模在 5~260 人之间; 实验周期短的在 1 周以内, 长的则超过 1 年; 用于实验的编程软件呈现多样化, 涵盖早期的 LOGO、近期的 Scratch 及一些新生编程软件如 Alice。此外, 不插电编程(Unplugged programming)也是一种较受欢迎的形式。

表 2 特征值编码表

特征	标志或说明
实验时间	2005 年以前=1, 2005—2015 年=2, 2015 至今=3
研究学段	学龄前=1, 小学=2, 中学=3
样本规模	小于 20=1, 20~50=2, 大于 50=3
实验周期	1 个月内=1, 2~6 个月=2, 超过 6 个月=3
编程形式	LOGO=1, Scratch=2, Unplugged programming=3, Other=4
测试类型	自行测试=1, 标准化测试=2

4. 元分析框架

依据布鲁姆的教学目标分类原理, 本研究将儿童编程对个体发展的作用效果具体划分为认知发展、行为提高、情感培养三个维度。其中, 认知发展包括学业成果的获得、知识的迁移、计算思维和数学思维的发展等; 行为提高主要涵盖实用层面的技能, 例如问题解决能力、分类规划能力、排序能力、协作创新能力的获得等; 情感培养涉及学习兴趣、学习动机、学习参与及自我效能感的获得等。与此同时, 我们引入编程对儿童个体发展影响的调节变量, 即不同学段、不同实验周期、不同编程形式及样本规模对儿童个体发展作用效果的影响, 以达到综合评价作用效果的目的(如图 2)。

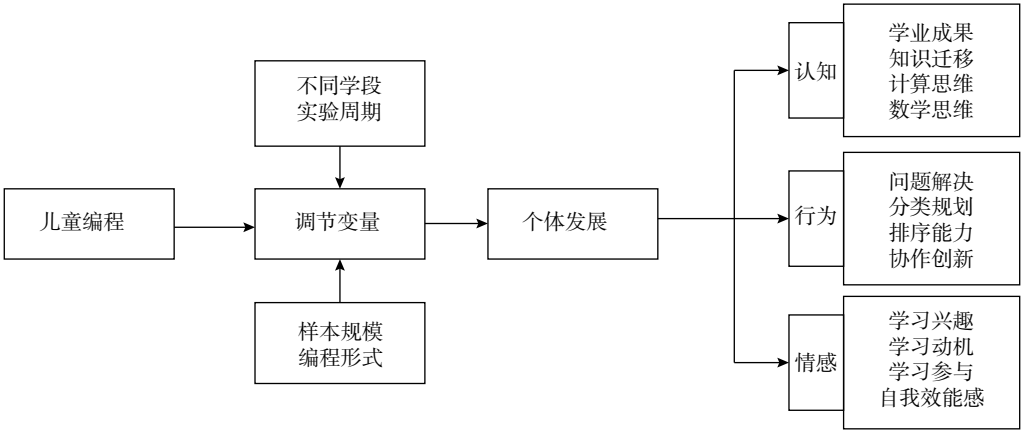


图 2 元分析框架

针对现有研究存在的问题, 本研究依据研究框架展开元分析, 以期回应如下问题:

- 问题 1: 整体而言, 编程能否真正促进儿童的个体发展? 如若编程对于儿童的个体成长起到促进作用, 其作用效果的程度是多少?
- 问题 2: 具体来看, 编程能否促进儿童个体发展中的认知、行为及情感等方面的提升? 且对于这三个方面所包含的维度的影响如何?
- 问题 3: 从学习者角度来看, 不同学段的儿童在编程教育中所获得个体发展的提升是否相同? 如有差异, 哪个学段的儿童能在编程活动中获得更多发展?
- 问题 4: 从实验周期来看, 编程周期的长短是否会影响儿童个体发展的作用效果? 哪种时长的编程活动能更加显著地促进儿童的个体发展?
- 问题 5: 从样本规模来看, 不同样本规模是否会影响编程个体发展的作用效果? 哪种规模的儿童样本最有利于促进儿童的个体发展?

问题6:从编程形式来看,不同的编程形式对儿童个体发展的作用效果是否有差异?哪种编程产品更有利于促进儿童的个体发展?

5. 数据分析

本研究采用元分析软件 Review Manager 5.3(RevMan)作为主要的数据处理与分析工具,将来自不同独立研究的原始数据包括样本数量(N),实验组、对照组前后测的均值(Mean),标准差(SD)录入 RevMan 软件,进行数据合并计算。首先依据 Hedges & Olkin(1985)提出的标准化均差 SMD 计算方法生成每个样本的效应量(公式1);其次根据漏斗图对样本数据进行发表偏倚检验并使用元分析统计公式2、3进行异质性检验,基于分析结果确定采用何种效应模型;最后依次计算合并后的平均效应量及分组效应量,并对效应量与特征编码进行统计分析,以了解研究特征对效应量的具体影响。

$$\text{效应值标准化均差(SMD)}d_i = \frac{\bar{x}_{1i} - \bar{x}_{2i}}{\sqrt{\frac{(n_{1i}-1)S_{1i}^2 + (n_{2i}-1)S_{2i}^2}{n_{1i} + n_{2i} - 2}}}, i = 1, 2, 3 \dots k \quad (1)$$

$$\text{检验统计量 } CHI^2 = \frac{2(\sum w_i d_i^2 - \bar{d}^2 \sum w_i)}{8 + \bar{d}^2}, i = 1, 2, 3 \dots k \quad (2)$$

$$\text{异质性的定量分析 } I^2 = \frac{CHI^2 - df}{CHI^2} \times 100\% \quad (3)$$

其中,第*i*项研究的实验组和对照组的样本量分别为 n_{1i} 和 n_{2i} , 均值分别为 \bar{x}_{1i} 和 \bar{x}_{2i} , 标准差分别为 S_{1i} 和 S_{2i} , 自由度为 df 。权重系数 w_i 表示样本量之和 ($w_i = n_{1i} + n_{2i}$), 加权均数 $\bar{d} = \frac{\sum w_i d_i}{\sum w_i}$ 用以反映各数值出现的次数对其在平均数中的影响。

三、研究结果与分析

(一) 发表偏倚检验

发表偏倚(Publication Bias)是指由于研究者、审稿人及期刊编辑在筛选发表论文时过于依赖研究结果的方向与强度而导致的偏差(杨扬, 沈志超, 靳纯桥, 2003), 导致元分析所纳入的独立研究可能无法系统地代表该领域已完成的研究总体而造成元分析结果存在偏倚(孔博丹, 许惠芳, 孔博鉴, 2014)。因此, 为保障元分析结果的科学有效性, 需对纳入元分析的独立研究样本进行发表偏倚的评估和检验。本研究使用漏斗图作为发表偏倚检验的依据, 其中, 横轴表示平均标准差的效应量, 纵轴表示效应量的标准误差。如图3所示, 大部分纳入元分析的研究样本效应量处于漏斗图中上部的有效区域内, 且相对均匀地对称散布于平均效应量的两侧, 不存在明显的异常值, 由此说明纳入元分析的研究文献发表偏倚现象可能性较小, 所得的分析结果可靠性较高。

(二) 异质性检验

研究样本的异质性检验结果显示, Chi^2 的值为 466.11 ($p < 0.001$), 这表明研究样本之间存在异质性, I^2 (研究过度分散占总分散的比率) 值为 84%, 表明大约 84% 的观察变异由效应量的真实差异导致, 而仅有 16% 的变异是由随机误差造成的(见表3)。Higgins et al.(2003)指出, 当 $I^2 = 0\%$ 时, 表明研究间不存在异质性; 当 $I^2 < 25\%$ 时, 则认为存在轻度异质性; 当 $25\% \leq I^2 \leq 50\%$ 时, 则认为存在中度异质性; 当 $I^2 > 50\%$ 时将被认为存在高度异质性。本研究异质性检验 $I^2 = 84\%$, 表明研究存在高度异质性, 产生异质性可能是由区域、发表时间、研究对象学段及实验周期的差异等诸多因素导致的。因此, 为保证研究的可靠性, 此次元分析在整体效应层面进行数据处理时采用随机效应模型(Random Effects Models, REM)消除异质性, 并进行效应量合并, 效应量的置信区间(CI)为 95%。

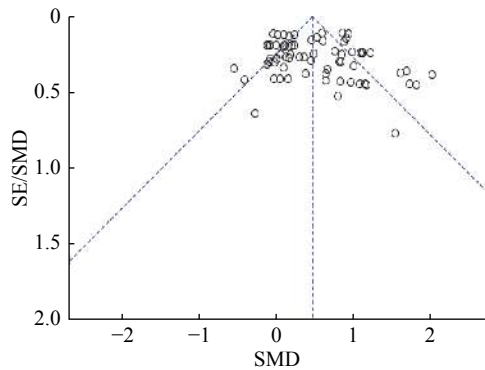


图 3 样本发表偏倚检测漏斗图

表 3 异质性检验结果

效应模型	研究数量	效应量SMD	95%置信区间		异质性检验		
			下限	上限	I ²	df	p
固定效应模型 (FEM)	74	0.45	0.40	0.50	84%	73	0.000
随机效应模型 (REM)	74	0.61	0.47	0.74			

（三）儿童编程赋能思维整体效应检验结果

为考察编程对儿童个体发展的整体影响,计算所得纳入研究的合并效应量 SMD 为 0.61(95% CI, 0.47~0.74), 合并效应量检验 $Z=8.76(p<0.001)$, 达到统计显著水平。根据 Cohen(1992)提出的效应量统计理论,当效应量为 0.2 时,表明影响很小;效应量处于 0.2~0.5 时为中等程度影响;当效应量 ≥ 0.8 则为显著影响。在本研究中,编程对儿童个体发展的整体影响的合并效应量为 $SMD=0.61>0.5$, 这表明编程对儿童的个体发展具有中等偏上的积极正向的促进作用。

本研究主要从三个维度(认知、行为、情感)来衡量编程对儿童个体发展的具体作用。其中,认知层面主要评估编程对儿童学业成果、知识迁移、计算思维及数学思维的作用效果;行为层面主要评估对儿童问题解决能力、分类规划能力、排序能力、协作创新能力的作用效果;而情感层面主要涉及编程对儿童学习兴趣、学习动机、学习参与及自我效能感的作用效果。如表 4 所示,在认知层面,编程对儿童个体影响的合并效应量是($SMD=0.86, p<0.001$), 达到统计学意义上的显著水平, 表明编程能够显著积极地促进学生的认知发展。其中编程对数学思维($SMD=2.68, p<0.001$)的作用效用值超过认知层面的整体水平, 表明编程对儿童数学思维的作用效果十分显著。而学业成果($SMD=0.61, p<0.001$)、知识迁移($SMD=0.82, p<0.001$)和计算思维($SMD=0.50, p<0.001$)的作用效应量略低于认知层面的整体水平, 说明编程在提高学业成果、知识迁移技能、培养儿童的计算思维方面的作用还有提升空间。在行为层面,编程对儿童个体发展作用的合并效应量为 ($SMD=0.34, p<0.001$), 表明编程对儿童的行为发展具有显著的中等作用效果。其中,排序能力($SMD=0.65, p<0.001$)作用效果最为显著, 其他方面如分类规划($SMD=0.31, p<0.05$)、问题解决($SMD=0.25, p<0.05$)和协作创新($SMD=0.40, p<0.01$)的作用效果也较为明显。在情感层面,编程对儿童个体发展作用的合并效应量为($SMD=0.31, p<0.001$), 表明编程对于儿童的情感发展具有显著的中等作用效果。其中作用效果最显著的是学习兴趣($SMD=0.51, p<0.01$), 此外,编程对儿童自我效能感的获得($SMD=0.22, p<0.05$)也具有较为显著的中等作用效果。但研究发现,编程对于儿童的学习动机($SMD=0.19, p>0.05$)和学习参与($SMD=0.20, p>0.05$)的作用效果并不显著,表明编程在这些方面的促进作用还有较大的提升空间。当然,这可能与相关研究设计和研究样本数量不足有关,因此后续还需在这些维度上开展更多的应用研究。

综合认知层面、行为层面和情感层面来看,编程对儿童的认知发展、行为发展和情感发展都具有显著的积极正向作用,且在认知层面的作用效果最为突出。研究结果表明,编程对于儿童个体发展的

促进作用是多方面、多层次的，编程不仅可以促进儿童的认知发展，提升学习成果、增强各项思维能力，也能促进儿童的行为发展，增强问题解决等实践性技能。此外，编程还能促进儿童的情感发展，因此，在编程活动中应加强儿童的学习兴趣和自我效能感等情感体验的获得。

表 4 编程对儿童个体发展的作用效果分析

维度	分类	数量	效应量SMD	95%的置信区间		Z值	合并效应量SMD	95%的置信区间	
				下限	上限			下限	上限
认知层面	学业成果	13	0.61	0.36	0.86	4.73***	7.98***	0.86	0.65 1.07
	知识迁移	6	0.82	0.57	1.08	6.35***			
	计算思维	15	0.50	0.30	0.80	3.28***			
	数学思维	6	2.68	2.07	3.29	8.62***			
行为层面	问题解决	6	0.25	-0.09	0.59	1.45*	3.52***	0.34	0.15 0.53
	分类规划	6	0.31	-0.05	0.67	1.67*			
	排序能力	2	0.65	0.30	1.00	3.61***			
	协作创新	6	0.40	-0.01	0.82	1.89**			
情感层面	学习兴趣	6	0.51	0.14	0.89	2.71**	3.12***	0.31	0.12 0.51
	学习动机	3	0.19	-0.20	0.57	0.95			
	学习参与	1	0.20	-0.18	0.57	1.04			
	自我效能感	4	0.22	-0.30	0.74	0.84*			

注：*=p<0.05，**=p<0.01，***=p<0.001。下同。

如前所述，编程有助于促进儿童个体发展的诸多方面，然而，针对不同的学段、实验规模、样本规模及编程形式，编程对儿童个体发展的促进作用是否存在差异？本研究做了进一步分析。

（四）不同学段对编程个体发展作用效果的差异

从不同学习阶段来看，当前儿童编程在学前、小学以及中学教育中都有相关的研究和应用，但主要聚焦中小学阶段。如表 5 所示，编程对不同学段学生个体发展的作用效果按照合并效应量大小排列依次是中学(SMD= 0.77, p< 0.001)、学前(SMD= 0.56, p< 0.001)及小学(SMD= 0.52, p< 0.001)；合并效应量检验 Z= 8.76(p< 0.001)，达到显著水平，表明编程对学前儿童、小学生及中学生的个体发展均具有显著的中等偏上的促进作用。由组间效应检验结果可知，效应量为($\text{Chi}^2= 2.67, p> 0.05$)，可见编程对不同学段学生个体发展的促进作用不存在显著差异。具体来看，编程对中学生个体发展的影响最为显著，而对学前及小学阶段的作用效果相差不大，这可以解释为中学生正处于个体发展的关键时期，不具备较为稳定成熟的能力和习惯，易受学习环境变化的影响；且较学前儿童和小学生而言，他们又具备较为成熟的认知能力，容易接受外来的知识。总体而言，编程对各学段学生的个体发展都存在积极、正向的促进作用，其中对中学生作用效果尤其突出。

表 5 不同学段对编程个体发展作用效果的差异

学段	数量	效应量SMD	权重	95%置信区间		异质性检验 (I^2) (%)	组间效应量	Z值
				下限	上限			
学前	14	0.56***	15.0	0.28	0.83	39	$\text{Chi}^2=2.67$ $p=0.25>0.05$	8.76***
小学	35	0.52***	49.6	0.32	0.71	84		
中学	25	0.77***	35.4	0.53	1.02	90		

（五）实验周期对编程个体发展作用效果的差异

为考察研究实验周期的长短对儿童个体发展的影响差异，本研究将实验周期划分为三个等级，分别为短期(1 个月以内)，中期(2~6 个月)及长期(超过 6 个月)。从表 6 可见，合并效应量检验

$Z=22.36(p<0.001)$, 达到显著水平, 这说明无论实验周期多长, 编程对儿童个体发展都具有显著的促进作用。从组间效应量来看, $\text{Chi}^2=24.68, p<0.001$, 达到统计学显著水平, 可见不同长短的编程实验周期对儿童个体发展的影响存在显著差异。由具体效应可知, 短期实验的合并效应量最高($\text{SMD}=0.43, p<0.001$), 长期实验次之($\text{SMD}=0.26, p<0.001$), 而中期实验的合并效应量最低($\text{SMD}=0.24, p<0.001$), 由此可知短期的编程实验比中期和长期的编程实验对儿童个体发展的影响更为显著。上述研究结果表明, 实验周期的长短会影响到编程对儿童个体发展的作用效果。短期实验个体发展促进效果最为突出, 出现这种现象的原因可能是短期实验儿童对编程抱有强烈的好奇心及高涨的热情和激情, 学习投入较多, 因而作用效果较为突出; 而随着实验周期的延长, 儿童对编程的熟悉度和适应度越来越好, 可能出现倦怠心理, 造成儿童的个体发展促进效果没有初始阶段那么显著。

表6 实验周期对编程个体发展作用效果的差异

实验周期	数量	效应量SMD	权重	95%置信区间		异质性检验 (I^2) (%)	组间效应量	Z值
				下限	上限			
短期	26	0.43***	71.3	0.39	0.47	91	$\text{Chi}^2=24.68$ $p<0.001$	22.36***
中期	37	0.24***	20.9	0.17	0.31	93		
长期	11	0.26***	7.8	0.14	0.37	91		

(六) 不同样本规模对编程个体发展作用效果的差异

为考察不同样本规模的编程对儿童个体发展的影响, 本研究分别从小(小于50人)、中(50~100人)、大(大于100人)三种规模分析编程对个体发展的促进作用。

由表7可知, 小规模、中等规模及大规模样本的个体发展合并效应量均为正值, 合并效应量检验 $Z=21.53(p<0.001)$, 达到显著水平。这说明无论样本规模为多大, 编程均对儿童个体发展具有中等偏下程度的积极影响。组间效应量显示 $\text{Chi}^2=23.53, p<0.001$, 达到显著水平, 表明编程对大、中、小规模样本学生个体发展的影响存在显著差异。从具体效应量来看, 小规模的合并效应量($\text{SMD}=0.41, p<0.001$)高于中等规模($\text{SMD}=0.37, p<0.001$)且高于大规模的合并效应量($\text{SMD}=0.17, p<0.001$), 这说明小规模编程活动比中等规模和大规模的编程活动对儿童个体发展的影响更为显著。而且, 样本规模与作用效果二者间存在负相关, 随着样本规模的增大, 作用效果逐渐降低, 此项研究发现可为后续编程活动的设计与开展提供借鉴参考。

表7 不同样本规模对编程个体发展作用效果的差异

样本规模	数量	效应量SMD	权重	95%置信区间		异质性检验 (I^2) (%)	组间效应量	Z值
				下限	上限			
小规模	53	0.41***	81.5	0.37	0.45	92	$\text{Chi}^2=23.53$ $p<0.001$	21.53***
中等规模	11	0.37***	5.1	0.22	0.51	70		
大规模	10	0.17***	13.4	0.08	0.26	96		

(七) 不同编程形式对编程个体发展作用效果的差异

当前, 儿童编程产品市场呈现百家争鸣的景象。多样的编程产品在编程教育中被研究和应用。作为最早的儿童编程语言——LOGO 经过时代的转化, 在当今焕发出了不同的生命活力。在现有研究中, LOGO 语言和 Scratch 被广泛使用, 同时, 不插电编程也逐渐成为低龄儿童群体极受欢迎的形式。此外, 随着技术的发展, 逐渐衍生出众多新型的编程形式, 例如 Alice、Code.org 等。为了评估编程产品对儿童个体发展的影响, 本研究分别分析了不同编程形式对个体发展的促进作用。

如表8所示, 合并效应量检验 $Z=25.76(p<0.001)$, 达到显著水平, 表明不同的编程形式对儿童的个体发展均具有显著的积极效应。从组间效应量来看, $\text{Chi}^2=32.72, p<0.001$, 达到显著水平, 表明不同的

编程形式在促进儿童个体发展的作用上存在显著差异。其中，不插电编程的合并效应量最高($SMD=1.47, p<0.001$)，其次为 LOGO 编程($SMD=0.78, p<0.05$)和其他编程形式($SMD=0.41, p<0.001$)，而 Scratch 的合并效应量最低($SMD=0.23, p<0.001$)，这表明不插电编程在促进儿童个体发展方面显著优于其他编程形式，但这并不排除是由几种编程形式的使用群体间的差异所造成的。

表 8 不同编程形式对编程个体发展作用效果的差异

编程形式	数量	效应量SMD	权重	95%置信区间		异质性检验 (I^2) (%)	组间效应量	Z值
				下限	上限			
LOGO	15	0.78*	0.2	0.03	1.54	74	Chi ² =32.72 p<0.001	25.76***
Scratch	19	0.23***	19.7	0.15	0.30	97		
Unplugged	4	1.47***	0.3	0.88	2.05	52		
Other	36	0.41***	79.8	0.37	0.44	88		

四、研究结论与启示

(一) 研究结论

本研究通过元分析方法对 28 项关于编程对儿童个体发展影响的实验或准实验研究进行了整体性的梳理与分析，并且依据元分析结果对编程的个体发展促进作用进行了客观分析与评估。在分析整体作用效果的基础之上，又将作用效果具体划分为认知、行为、情感三个层面，并通过引入学段、实验周期、样本规模及编程形式作为调节变量分析讨论编程对儿童个体发展的作用效果及其差异。

1. 编程对儿童的个体发展具有正向促进作用，显著体现在认知、行为及情感三个层面

元分析结果显示，编程对儿童个体发展的合并效应量为 0.61，处于中等偏上水平，表明编程可以积极正向地促进儿童的个体发展。具体来看，编程能显著地促进学生认知发展($SMD=0.86$)中的学业成果、知识迁移、计算思维及数学思维；且对于儿童的行为发展也具有显著的中等作用效果($SMD=0.34$)，有助于培养儿童问题解决能力、分类规划能力、排序能力及协作创新能力；编程对于儿童的情感发展具有显著的中等作用效果($SMD=0.31$)，主要体现在儿童学习兴趣及自我效能感的获得。数据表明，编程不仅可以提升儿童的编程技能和计算机科学知识，还可以提升儿童的其他技能和素养，以实现儿童的个体发展。这与 Popat et al.(2019)开发的描述主题总体模型显示研究结果一致，即学生在学习编程的过程中可以学习或实践一系列其他教育成果，例如数学问题解决、批判性思维、社会技能、自我管理和学术技能等。无独有偶，Nouri et al.(2020)指出，编程不仅能使儿童掌握编程概念，而且还可以让他们获得计算思维数字能力及 21 世纪技能的发展。上述研究结果集中体现了儿童编程的创立图景。从西蒙·派珀特(Seymour Papert)LOGO 的创立，倡导“要让儿童成为计算机的主人，利用计算机改变他们学习其他知识的方式”(Papert, 1980)，到米切尔·雷斯尼克(Mitchel Resnick)的 Scratch 打造的“4P”学习法，即由项目(Project)切入，从热情(Passion)出发，同伴(Peers)协作互助，游戏(Play)启动内驱，通过创造性学习螺旋的多次迭代培养儿童创造性的思考、系统的推理及协作的能力(Resnick, 1998)，再到玛蒂娜·U·贝斯(Marina U. Bers)的 ScratchJr 所主张的“编码和其他任何语言一样，可以用来表达自己的概念、技能和思维习惯，帮助我们用新的方式思考和交流”(Bers, Gonzalez-Gonzalez, & Torres, 2019)，无不展现出儿童编程的强大教育作用。

2. 编程对儿童个体发展的促进作用受学段、实验周期、样本规模及编程形式等调节变量的影响

调节变量效应检验结果表明，编程对儿童个体发展的作用效果存在边界条件，主要受不同的学段、实验周期、样本规模及编程形式的影响。从学段来看，编程对学前儿童、小学生及中学生的个体发展均具有显著的促进作用。其中，编程对中学生个体发展的作用效果最为显著($SMD=0.77$)，而在学前($SMD=0.56$)及小学($SMD=0.52$)阶段编程的影响效果相差不大；组间效应检验结果显示，编程对不

同学段儿童的个体发展的促进作用不存在显著差异。在实验周期方面,不同周期的编程活动对儿童个体发展都具有中等偏下程度的促进作用,但不同长短的编程实验周期对儿童个体发展的影响存在显著差异,且短期实验周期($SMD=0.43$)的作用效果更为显著。从样本规模来看,编程对不同规模样本学生个体发展的影响存在显著差异。相较于中等规模($SMD=0.37$)和大规模($SMD=0.17$),编程在小规模样本($SMD=0.41$)中的作用效果更为显著。在编程形式方面,合并效应量检验达到显著水平,表明不同编程形式对儿童的个体发展均具有显著的积极效应,且组间效应量表明不同的编程形式在促进儿童个体发展的作用上存在显著差异。其中,不插电编程($SMD=1.47$)在促进儿童个体发展方面要显著优于LOGO编程($SMD=0.78$)、其他编程形式($SMD=0.41$)及Scratch编程($SMD=0.23$)。

(二) 研究启示

综合来看,元分析结果肯定了编程对儿童成长的重要性及其对个体发展的积极促进作用。但通过研究结果也可以看出,编程教育在面对不同的学习者及学习情境时,会呈现不同的作用效果。因此编程教学实践不能盲目开展,而应根据教学系统各要素的不同特征,进行科学合理的设计与规划。

1. 编程对儿童个体发展的作用效果还有较大提升空间

虽编程在一定程度上可以积极促进儿童的个体发展,但这与儿童编程的创立图景及政府部门所提出的推动信息化进程、加快信息革命、实现教育现代化的愿景仍有一定距离。研究表明,编程对儿童的学习动机和学习参与的作用效果并不显著,且其对于儿童分类规划、问题解决及协作创新等方面的作用效果也均未达到整体影响水平。面向未来的教育需要最大可能地将人工智能(如编程)与人类的社交和情感技能、态度和价值观结合起来(施莱希尔, 2020)。因此,在进行编程教育推广实践时,教育者应清楚地认识到,教授编码是编程教育的一个重要任务,但编程的教育价值远比编码广泛得多,因此在教学过程中既应发挥编程课程的主体性能,培养儿童的编程能力;又应注重编程对教育的赋能价值,关注其对于儿童个体综合发展的促进作用,通过设计科学合理的教学方案保障编程赋能作用的发挥。此外,政府部门也需继续推动各项相关政策的合理制定与贯彻实施,以确保儿童编程教育真为儿童发展“谋福利”。未来还需政府、学校、社会多方面联动协调,逐步增强编程对儿童个体发展的促进效果和影响水平。

2. 加速推广编程在中小学中的应用,积极推进编程教育学段下移

前述研究数据显示,当前编程在中小学阶段开展较多,且作用效果较为显著,基于此,本研究建议应加速推广编程在中小学教育中的应用。全国各地中小学都应积极开展“编程教育进校园”活动,不仅增设独立的编程课程,更应将其作为一种教学或学习的方式融入中小学各门学科之中。同时,通过政策制度把关儿童编程教师的行业准入标准,及通过相关培训不断提升教师的编程教学技能。此外,研究还发现学前教育阶段的编程教育研究成果较少,因此,在后续研究中,应积极推进在学前教育阶段开展编程教育实践活动,为编程能否促进学前阶段儿童的个体发展提供充分而强有力的证据。贝斯一直强调,年龄较小的儿童也具备学习编程的能力(Bers, Gonzalez-Gonzalez, & Torres, 2019),而皮亚杰(1981)在其S(A)R的公式中,也指出一个个体要想对刺激做出反应,首先需要具有同化这个刺激的格局。因此,“编程要从娃娃抓起”,实现编程教育的“幼小初高”基础教育阶段的无缝衔接,目的并非是为了培养一批未来的“码农”“程序员”,而是为了使儿童尽快地适应未来的人工智能社会,在编程中发展自我、成就自我,成为数字产品的生产者而不仅仅是消费者,赋予每一个儿童“改变世界”的可能。

3. 寻找编程对儿童个体发展促进作用持续性的平衡点

实验周期的差异检验结果表明,编程对儿童个体发展的作用受实验周期的影响,短期实验的作用效果最为显著,远高于中期实验与长期实验。而编程对儿童个体发展的作用效果随着实验周期的延长出现逐渐降低的情况,足以表明编程对儿童个体发展促进作用的持续性不足,无法发挥持久稳定的作用。编程是一个试误的过程,随着学习时间的延长,知识的不断深入,学生极有可能产生倦怠心理,进

而导致编程作用效果降低。因此,编程对儿童个体发展促进作用的持续性需进一步加强,这个过程离不开各方面的支持与辅助。一方面,教师应给予适当鼓励和支持,赋予儿童继续学习的动力;另一方面,通过设计个性化、多样化趣味合理的教学活动,不断激发学生的求知欲,寻求儿童的认知因素和动机因素调整的最佳水平,寻找编程学习时长和个体发展作用效果的平衡点,进而增进编程对个体发展作用的持续性,实现最佳的编程教育效果。

4. 贯彻落实国家政策,聚焦编程课堂的小班制情境

编程对不同规模样本学生的个体发展的影响存在显著差异,编程活动在小规模样本实践中对儿童个体发展的作用效果更为显著,且样本规模与作用效果间存在负相关,即随着样本规模的增大,作用效果逐渐下降。因此,为保障编程教育的作用效果,编程教学情境应聚焦小班教学,班容量最好控制在50人以下。中华人民共和国教育部(2010)《关于“十二五”期间加强学校基本建设规划的意见》对初中段、小学段具体招生办法做了详细规定,明确继续严格控制班容量,原则上普通中学每班学生45~50人,城市小学40~45人。但由于我国儿童人口数量众多,小班制的教育形式在一些欠发达地区仍难以实现,为此政府需加强监管,出台相应的政策支持薄弱地区的教育发展建设,进行合理的资源与师资分配。只有这样,教师才能有更充沛的精力和时间关注每个儿童的学习情况,采取因材施教的方式制定个性化的学习方案,设计科学合理的个性化编程教学活动促进学生的个体发展。

5. 遵循儿童认知发展规律,选择与之匹配的编程形式

虽然不同的编程形式对儿童的个体发展均具有显著的积极作用,但不同的编程形式在促进儿童个体发展的作用上仍存在显著差异。其中,不插电编程的作用效果最为突出,其次为LOGO编程和其他编程形式,而Scratch的作用效果最低。不同的编程形式具有不同的特点,因此,在开展编程教学活动时,应遵循儿童的认知发展规律,结合教育情境的特征,选择与之匹配的编程形式,并通过编订合理的编程教材以及设计科学的教学活动,在最大程度上促进学生个体发展的作用效果。此外,同一种编程形式在不同科目中会有不同的表现,Moreno-León et al.(2016)在研究中发现,计算机编程的使用加速了儿童的学习曲线,但在社会科课堂中的影响大小是数理科课堂中的两倍。因此,在将编程与其他学科结合开展时,也应依据学科特点选择适合的编程形式。

五、结语

本研究借鉴相关研究者有益的经验 and 做法,采用较为科学且行之有效的元分析方法,客观评估了编程对儿童个体发展的作用效果及影响,并通过对已有研究的综合梳理分析,得出编程对儿童个体发展的总体影响的合并效应量,采用量化数据结果试图解释编程对儿童个体发展的积极、正向的促进作用,进而对当前学术界有争议的话题“编程能否真正促进儿童的个体发展”做出回应。同时,本研究针对不同的学段、实验周期、样本规模及编程形式对编程个体发展促进作用的差异做了进一步探究分析,有效回答了“哪些因素会影响编程对儿童个体发展的促进作用”这一问题。本研究结果是在广泛收集具备代表性样本文献的基础上加以量化分析得出的,具有一定的可靠性和有效性。但由于元分析方法自身仍存在一些局限性,得到的研究结论属于推断性结论而并非析因性结论,因此在研究结果的推广方面还需谨慎。

参考文献

- 陈鹤阳,朝乐门.(2015).元分析方法在信息分析中的应用.《情报资料工作》,36(6),46—50.
- 顾小清,胡梦华.(2018).电子书包的学习作用发生了吗?——基于国内外39篇论文的元分析.《电化教育研究》,39(5),19—25.
- 孔博丹,许惠芳,孔博鉴.(2014).元分析常见问题及解决方法.《心理技术与应用》,2(1),19—22.
- 皮亚杰.(1981).《发生认识论原理》(王宪钊等译,胡世襄等校).北京:商务印书馆.
- 森秀树,杉泽学,张海,前迫孝惠.(2011).Scratchを用いた小学校プログラミング授業の実践.《日本教育工学会論文誌》,34(4),387—394.

- 施莱希尔. (2020). 教育面向学生的未来, 而不是我们的过去 (宋晓凤, 陈钰童译). *华东师范大学学报(教育科学版)*, 38(5), 1—21.
- 石晋阳. (2018). *儿童编程学习体验研究*. 南京: 南京师范大学博士学位论文.
- 王翠如, 胡永斌. (2018). 翻转课堂真的能提升学习成绩吗?——基于 38 项实验和准实验研究的元分析. *开放教育研究*, 24(4), 72—80.
- 王雪, 徐文文, 高泽红, 王志军. (2019). 虚拟现实技术的教学应用能提升学习效果吗?——基于教学设计视角的 38 项实验和准实验的元分析. *远程教育杂志*, 37(6), 61—71.
- 杨扬, 沈志超, 靳纯桥. (2003). 发表偏倚的原因、后果与预防研究. *中国科协 2003 年学术年会*. 中国辽宁沈阳.
- 中华人民共和国教育部. (2010-06-22). 关于“十二五”期间加强学校基本建设规划的意见. 2020-01-11. 取自: <https://www.doc88.com/p-6866673049430.html>.
- 中华人民共和国中央人民政府. (2017-09-15). 国务院关于印发新一代人工智能发展规划的通知. 2020-01-11, 取自: http://www.gov.cn/zhengce/content/2017-07/20/content_5211996.htm.
- Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International Journal of Child-Computer Interaction*, 19, 30—55.
- Berland, M., & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology*, 24(5), 628—647.
- Bernardo, M. A., & Morris, J. D. (1994). Transfer effects of a high school computer programming course on mathematical modeling, procedural comprehension, and verbal problem solution. *Journal of Research on Computing in Education*, 26(4), 523—536.
- Bers, M. U., Gonzalez-Gonzalez, C., & Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, 138(1), 130—145.
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., ... Barone, D. (2017). *Development of computational thinking skills through unplugged activities in primary school*. Paper presented at the Proceedings of the 12th Workshop on Primary and Secondary Computing Education (pp. 65—72), Nijmegen, Netherlands.
- Browning, S. F. (2017). Using Dr. Scratch as a formative feedback tool to assess computational thinking. *All Theses and Dissertations*, 6659, 1—78.
- Calao, L. A., Moreno-León, J., Correa, H. E., Gregorio, & Robles. (2015). Developing mathematical thinking with Scratch. *Design for Teaching and Learning in a Networked World*, 9307, 17—27.
- Cetin, I. (2016). Preservice teachers' introduction to computing: Exploring utilization of Scratch. *Journal of Educational Computing Research*, 54(7), 997—1021.
- Chou, P. (2019). Using ScratchJr to foster young children's computational thinking competence: A case study in a Third-Grade computer class. *Journal of Educational Computing Research*, 58(3), 570—595.
- Clements, D. H. (1986). Effects of Logo and CAI environments on cognition and creativity. *Journal of Educational Psychology*, 78(4), 309—318.
- Clements, D. H. (1990). Metacomponential development in a Logo programming environment. *Journal of Educational Psychology*, 82(1), 141—149.
- Clements, D. H., & Meredith, J. S. (1993). Research on Logo: Effects and efficacy. *Journal of Computing in Childhood Education Archive*, 4(4), 263—290.
- Clements, D. H., & Sarama, J. (1997). Research on Logo: A decade of progress. *Computers in The Schools*, 14(1), 9—46.
- Cohen, J. (1992). A power primer. *Psychological Bulletin*, 112(1), 155—159.
- Corral, J. M. R., Balcells, A. C., Estevez, A. M., Moreno, G. J., & Ramos, M. J. F. (2014). A game-based approach to the teaching of object-oriented programming languages. *Computers & Education*, 73, 83—92.
- Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr on the iPad. *Journal of Computer Assisted Learning*, 32(6), 576—593.
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers in Education*, 63, 87—97.
- Fridin, M. (2014). Storytelling by a kindergarten social assistive robot: A tool for constructive learning in preschool education. *Computers & Education*, 70, 53—64.
- Glass, G. V. (1976). Primary, secondary, and meta-analysis of research. *Educational Researcher*, 5(10), 3—8.
- Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2), 199—237.
- Hatice, Y. D. (2018). The effects of using different tools in programming teaching of secondary school students on engagement, computational

- thinking and reflective thinking skills for problem solving. *Technology Knowledge & Learning*, 25(1), 179—195.
- Hayes, J., & Stewart, I. (2016). Comparing the effects of derived relational training and computer coding on intellectual potential in school-age children. *British Journal of Educational Psychology*, 86(3), 397—411.
- Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. New York: Publisher Academic Press.
- Hermans, F., & Aivaloglou, E. (2017). *To Scratch or not to Scratch?: A controlled experiment comparing plugged first and unplugged first programming lessons*. Paper presented at the Proceedings of the 12th Workshop on Primary and Secondary Computing Education (pp. 49—56), Nijmegen, Netherlands.
- Higgins, J. P. T., Thompson, S. G., Deeks, J. J., & Altman, D. G. (2003). Measuring inconsistency in meta-analyses. *BMJ*, 327(7414), 557—560.
- Johnsongentile, K., Clements, D. H., & Battista, M. T. (1994). Effects of computer and noncomputer environments on students' conceptualizations of geometric motions. *Journal of Educational Computing Research*, 11(2), 121—140.
- Jun, S., Han, S., & Kim, S. (2017). Effect of design-based learning on improving computational thinking. *Behaviour & Information Technology*, 36(1), 43—53.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52(52), 200—210.
- Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, 13(1), 33—50.
- Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 41(4), 245—255.
- Kazakoff, E. R., & Bers, M. U. (2014). Put your robot in, put your robot out: Sequencing through programming robots in early childhood. *Journal of Educational Computing Research*, 50(4), 553—573.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming. *Computers in Human Behavior*, 41(41), 51—61.
- Miller, R. B., Kelly, G. N., & Kelly, J. T. (1988). Effects of Logo computer programming experience on problem solving and spatial relations ability. *Contemporary Educational Psychology*, 13(4), 348—357.
- Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of Scratch projects to assess and foster computational thinking. *Revista de Educación a Distancia*, (46), 1—23.
- Moreno-León, J., Robles, G., & Román-González, M. (2016). Code to learn: Where does it belong in the K-12 curriculum?. *Journal of Information Technology Education: Research*, 15, 283—303.
- Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1—17.
- Palumbo, D. B. (1990). Programming language/problem-solving research: A review of relevant issues. *Review of Educational Research*, 60(1), 65—89.
- Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: A case study. *International Journal of Mobile Learning and Organisation*, 10(3), 187—202.
- Papatga, E., & Ersoy, A. (2016). Improving reading comprehension skills through the Scratch program. *International Electronic Journal of Elementary Education*, 9(1), 124—150.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Pea, R. D., & Kurland, D. Midian. (1984). Logo programming and the development of planning skills. *Technical Report*, 16(1), 3—56.
- Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, 128, 365—376.
- Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, 45(5), 583—602.
- Resnick, M. (1998). Technologies for lifelong kindergarten. *Educational Technology Research and Development*, 46(4), 43—55.
- Resnick, M., Maloney, J., Monroyhernandez, A., Rusk, N., Eastmond, E., Brennan, K.,... Silverman, B. (2009). Scratch: Programming for all. *Communications of The ACM*, 52(11), 60—67.
- Román-González, M., Pérez-González, J., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the computational thinking test. *International Journal of Child-Computer Interaction*, 18, 47—58.
- Rose, S. P., Habgood, M. P. J., & Jay, T. (2017). An exploration of the role of visual programming tools in the development of young children's computational thinking. *Electronic Journal of e-Learning*, 15(4), 297—309.
- Sáez-López, J., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in element-

- ary school: A two-year case study using “Scratch” in five schools. *Computers & Education*, 97, 129—141.
- Stoeger, H., Duan, X., Schirner, S., Greindl, T., & Ziegler, A. (2013). The effectiveness of a one-year online mentoring program for girls in STEM. *Computers & Education*, 69, 408—418.
- UNICEF. (1989). Convention on the rights of the child. 2020-06-17, Retrieved from <https://www.unicef.org/child-rights-convention/convention-text>.
- Wang, X., Hwang, G., Liang, Z., & Wang, H. (2017). Enhancing students’ computer programming performances, critical thinking awareness and attitudes towards programming: An online peer-assessment attempt. *Journal of Educational Technology & Society*, 20(4), 58—68.
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, 14(1), 1—16.
- Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills?. *Computers & Education*, 141, 1—13.

(责任编辑 童想文)

Can Programming Really Promote the Individual Development of Children? A Meta-analysis of 28 Experimental and Quasi-experimental Studies

Sun Lihui Hu Linlin

(College of Education, Tianjin University, Tianjin 300350, China)

Abstract: As a basic accomplishment for children to face, adapt to the future and transform the future, programming has been constantly paid more and more attention to and attached importance to. However, how much educational value it has, for this reason, many scholars have carried out a large number of experiments and quasi-experimental studies based on the individual development of children, but the results are not the same. The question of whether programming can actually promote the individual development of children remains a subject of academic debate. Based on this, this study uses the meta-analysis method to quantitatively analyze 28 foreign experiments and quasi-experimental studies on the influence of programming on the individual development of children. It is found that: (1) overall, the combined effect size of the included studies is 0.61, which is in the upper-middle level, indicating that programming can positively promote the individual development of children; (2) specifically, programming can significantly and positively promote students’ cognitive development, behavioral development and emotional development; (3) compared with primary schools and preschools, programming plays a more prominent role in children’s separate development in middle schools; (4) compared with medium-term and long-term programming activities, short-term programming activities have a more significant promoting effect on children’s individual development; (5) programming has a greater effect on children’s individual development in small-scale samples than in medium and large samples; (6) different programming forms have different effects, among which unplugged programming has a more significant effect on children’s individual development. Based on the findings of the research conclusions, it is expected to provide reference for the future research and practice of programming for children in China.

Keywords: children programming; individual development; meta-analysis; effect size