




Promoting the AI teaching competency of K-12 computer science teachers: A TPACK-based professional development approach

Junmei Sun¹ · Hongliang Ma¹  · Yu Zeng¹ · Dong Han¹ · Yunbo Jin¹

Received: 2 March 2022 / Accepted: 19 July 2022 / Published online: 30 July 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

With the rapid development of artificial intelligence (AI), the demand for K-12 computer science (CS) education continues to grow. However, there has long been a lack of trained CS teachers. To promote the AI teaching competency of CS teachers, a professional development (PD) program based on the technological pedagogical content knowledge (TPACK) framework was intentionally designed in this research. A quasi-experimental design with a 25-day (75-h) intervention was conducted among 40 in-service CS teachers to examine its impact on AI teaching competency, including AI knowledge, AI teaching skills, and AI teaching self-efficacy. The quantitative data were collected via a pretest and posttest, and qualitative data were collected via artifact analysis and semistructured interviews. The results indicated that the TPACK-based PD program a) significantly improved CS teachers' AI knowledge, especially in representation and reasoning, interaction, and social impact; b) developed CS teachers' AI teaching skills, including their AI lesson plan ability and AI programming skills; and c) significantly improved CS teachers' AI teaching self-efficacy, both in AI teaching efficacy beliefs and AI teaching outcome expectancy. These findings revealed the effectiveness of the TPACK-based PD program in improving the AI teaching competency of K-12 CS teachers and could help to expand the design of effective PD for CS teachers.

Keywords AI teaching competency · Computer science teachers · Teacher professional development · TPACK · In-service teachers

✉ Hongliang Ma
mahl@snnu.edu.cn

¹ School of Education, Shaanxi Normal University, Xi'an 710062, Shaanxi, China

1 Introduction

With the rapid development of artificial intelligence (AI), the demand for K-12 computer science (CS) education continues to rise (El-Hamamsy et al., 2021). However, there has long been a lack of trained CS teachers due to reduced budgets (Margolis et al., 2011). As a result, school administrators have increasingly been forced to ask teachers with little formal training in CS to teach CS courses (Moin et al., 2005). In China, the national government advanced the development and implementation of an AI and programming curriculum for K-12 education in the national policy “*The Development Plan of the New Generation Artificial Intelligence*”, but due to a lack of AI subject knowledge, teaching skills, and attitudes, large numbers of in-service CS teachers are not competent in teaching AI courses or modules in schools.

Reflecting the potential for effective teacher behavior in specific teaching situations, teaching competency is necessary to enhance in-class professionalism (Kim & Kim, 2016), and it involves the combination of psychological qualities such as the knowledge, skills, and dispositions that teachers need to complete teaching tasks (Caena, 2014). In K-12 AI education, although researchers have conducted some studies and implemented interventions among primary and secondary school students directly (Goel & Joyne, 2017; Henry et al., 2021), few studies have targeted in-service CS teachers (Ng et al., 2021). To fill this gap, a professional development (PD) program based on the technological pedagogical content knowledge (TPACK) framework combining online and offline activities was designed and implemented to promote CS in-service teachers’ AI teaching competency in the current research, and the effectiveness of the program was also examined with quantitative data and qualitative data. To some extent, the specific TPACK-based framework oriented toward AI teaching competency (TPACK_{AI}) and its corresponding teacher learning activities developed with some effective elements of PD in this study may theoretically contribute to the literature and practically make up for the gap in CS teacher PD initiatives in China.

The paper is organized as follows: Section 2 provides a literature review, Section 3 describes the research methodology, Section 4 shows the results, Section 5 discusses the findings, and the last section presents the conclusion.

2 Literature review

2.1 AI teaching competency

Teaching competency is defined as an integration of the knowledge, skills, and attitudes required for the successful implementation of subject matter education (Koster et al., 2005; Prajugjit & Kaewkuekool, 2020), and it is necessary to enhance professionalism in class (Doodewaard, 2020). Among various models of teacher competency, the COACTIV model of teachers’ professional competency

by Baumert and Kunter (2013) is highly relevant to this study. This model assumes a diverse set of capacities, including cognitive (i.e., knowledge and skills) and noncognitive characteristics (belief), that constitute the key determinants of successful teaching (Kunter & Baumert, 2013). Drawing on the COACTIV model, we conceptualize three aspects of AI teaching competency that are particularly linked to teachers' AI teaching practice: AI teaching knowledge, AI teaching skills and AI teaching belief (self-efficacy).

Regarding the cognitive area, the TPACK framework (Mishra & Koehler's, 2006) is usually adopted to identify teaching knowledge and skills. Kim et al. (2021) used the TPACK framework to determine which teaching abilities are necessary for CS teachers to improve AI teaching in the K-12 stage. In terms of pedagogical knowledge, teachers are required to implement problem-based learning (PBL) and game-based learning with appropriate AI ethics. Regarding content knowledge, teachers are required to reach the knowledge level of undergraduate students majoring in AI. In terms of technological knowledge, teachers are required to master programming technology and choose appropriate AI platforms.

Regarding the noncognitive area, teaching self-efficacy is one of the most important constructs in teaching competency (Lauermaun & König, 2016). Based on Bandura's (1997) work, teachers' self-efficacy denotes teachers' beliefs about their abilities to succeed in specific situations. For teachers' professional development (TPD), teaching self-efficacy is a potential construct that affects teachers' actual teaching knowledge, ability, behavior, and thinking (Orakcı et al., 2020; Tschannen-Moran & Hoy, 2007). Burić and Kim (2020) reported that teaching self-efficacy is one of the most significant motivational features that affects teachers' teaching quality and students' motivational beliefs. Other studies have also indicated that there is a strong correlation between teachers' teaching self-efficacy and students' achievements (Tschannen-Moran & Hoy, 2007). The extent to which teachers perceive such efficacy may influence whether they will take action, whether they invest effort in an action, and how long they may sustain possible challenges (Tschannen-Moran & Hoy, 2001). Therefore, this study considers teachers' AI self-efficacy as a decisive element for CS teachers' success in AI teaching.

2.2 Professional development for CS teachers

TPD is mainly regarded as the PD of in-service teachers and is defined as a broad activity that develops teachers' skills, knowledge, expertise, and other characteristics (OECD, 2009). Effective PD can facilitate the transformation of teaching practice and ultimately improve students' learning (Desimone, 2009). Even for qualified in-service teachers in CS, PD is necessary, as it allows them to enhance their disciplinary knowledge, acquire innovative pedagogical approaches, and deepen their content pedagogical skills (Brandes & Armoni, 2019; Tokmak et al., 2013).

In terms of PD approaches for CS teachers, face-to-face, online or blended activities have been designed and implemented in previous research (Armoni, 2017; Lazarinis et al., 2019; Murai & Muramatsu, 2020). For instance, Rich et al. (2021) conducted a face-to-face PD program to train CS teachers to teach coding to K–6

students. The year-long PD program included open-ended, hands-on, model lessons, and other activities, and the results showed that there were statistically significant increases in computing teaching efficacy and teaching values. Considering the cost and distance, Goode et al. (2020) transformed face-to-face PD activities into fully online activities to make K-12 CS teachers competent in teaching new and equitable CS courses. To improve K-12 CS teachers' Scratch programming skills, Lazarinis et al. (2019) implemented a blended PD with the Moodle platform and found that it promoted the development of teachers' computing thinking.

In terms of PD content for CS teachers, with few studies on AI knowledge and related pedagogy, most research has mainly focused on CS concepts (El-Hamamsy et al., 2021), knowledge (Chai et al., 2020), programming (El-Hamamsy et al., 2021; Martinez et al., 2016), computational thinking (Monjelat & Lantz-Andersson, 2019; Reimer & Blank, 2018), and pedagogical skills (Qian et al., 2018; Sentance et al., 2018). For instance, Martinez et al. (2016) carried out an introductory CS PD course for K-12 teachers, which was integrated with pedagogical content knowledge and teacher classroom practice. The results showed that the PD program effectively improved inquiry-based CS teaching. However, the PD course did not include AI-related TPACK, with only a focus on teachers' fundamental programming concepts. Kandlhofer et al. (2019) presented an educational project for training and certifying teachers and school students in AI and robotics and described the four stages and the detailed content of the training modules and curricula in this project. Although the appropriateness of the teaching methods and teaching materials of the developing system were explored among 16 teachers, the impact on teachers' learning outcomes has not yet been examined. Vazhayil et al. (2019) conducted a PD workshop, with only two learning activities on the technical application of text and image recognition to train CS teachers to introduce AI in their schools; however, the PD program lasted for only 2 days, and qualitative data were collected from teachers to examine the effectiveness of the workshop.

2.3 In-service teacher professional development based on the TPACK framework

Initially, the TPACK framework was utilized to conceptualize teachers' technology integration competency in teaching (Rosenberg & Koehler, 2015). Mishra and Koehler (2006) summarized this complex intersection of domain knowledge as TPACK, expanding the original pedagogical and content knowledge (PCK) model (Shulman, 1986). In teacher education, the TPACK framework has been widely used for in-service teacher PD. Research has shown that TPACK-based PD programs improve in-service teachers' knowledge, self-efficacy, and technology integration skills (Baran, 2010; Blonder et al., 2013; Hong & Stonier, 2015; Koehler et al., 2007; Oda et al., 2020). For instance, Oda et al. (2020) organized 24 school teachers to learn about the integration of GIS in science or social science classes, and the TPACK framework was used to understand the properties and impacts of PD; the authors found that the framework helped teachers introduce and use GIS technologies in their class more often.

However, for in-service CS teachers, the literature on TPACK-based PD is limited, with a focus mainly on computational thinking (CT). Angeli et al. (2016) discussed the TPACK that teachers needed to teach a K-6 CS curriculum based on a CT framework and developed a corresponding PD course, but they provided little empirical evidence in terms of the effectiveness of the CT-based TPACK framework. Kong et al. (2020) implemented a TPACK-based PD program with two 39-h courses for 76 in-service primary school teachers that mainly focused on CT concepts and practice. Although knowledge tests, surveys and reflections were used to examine the teachers' understanding of TPACK-related CT knowledge and perception of PD, AI knowledge was not included, nor were the related teaching skills and self-efficacy examined in this PD program. Chen and Cao (2022) examined a virtual PD program among 43 in-service school teachers, including CS, mathematics, science, and other subjects. The program aimed to improve K-12 teachers' knowledge, attitudes and beliefs in maker-centered instruction instead of their AI teaching competency. The researchers reported that the study relied heavily on self-reports for assessing teachers' capabilities and attitudes, which may influence the reliability and validity of the research. For AI-related PD, Kim et al. (2021) used the TPACK framework to determine which teaching abilities are necessary for CS teachers to enact AI teaching in K-12 education; however, this study did not extend the framework into the PD program and examine the effectiveness on teachers' AI competency.

In sum, general TPACK-based frameworks for CS teachers' PD have been proposed, and their effectiveness has been examined in prior studies, but few studies have designed specific TPACK-based PD programs to improve CS teachers' AI teaching competency, namely, their AI knowledge, skills, and teaching self-efficacy. To address this gap, the current research intentionally designed a TPACK-based PD program incorporating some effective elements of TPD in a blended environment (as described in detail in Sect. 3.3) and then examined the effectiveness of this PD program.

The following research questions (RQs) were investigated in particular in this research:

RQ1: Will the AI knowledge of K-12 CS teachers be significantly improved by the TPACK-based PD program?

RQ2: Will the AI teaching skills of K-12 CS teachers be developed by the TPACK-based PD program?

RQ3: Will the AI teaching self-efficacy of K-12 CS teachers be significantly improved by the TPACK-based PD program?

3 Methods

3.1 Participants

Forty K-12 CS teachers from a northwestern province in China participated in this research voluntarily, and the demographic information is described in Table 1. Among these participants, 22 (55.0%) teachers were male, and 18 (45.0%) were

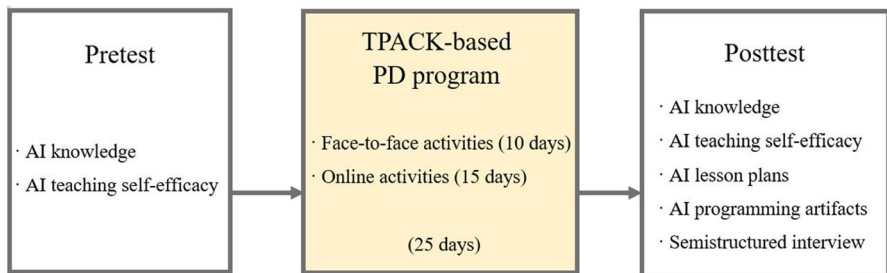
Table 1 Participants' demographic information

Category	Item	Frequency	Percent
Gender	Male	22	55.0%
	Female	18	45.0%
Age	26–30 years	7	17.5%
	31–35 years	10	25.0%
	36–40 years	16	40.0%
	41–45 years	7	17.5%
Teaching experience	1–5 years	6	15.0%
	5–10 years	8	20.0%
	More than 10 years	26	65.0%
Geographic location	Urban	33	82.5%
	Rural	7	17.5%
Educational level taught	Primary school	14	35.0%
	Middle school	9	22.5%
	High school	17	42.5%

female; 16 (40.0%) teachers were aged between 36–40 years old; 26 (65.0%) teachers had been teaching more than 10 years, and 33 (82.5%) teachers had been teaching in urban schools. Notably, none of the participants attended any other PD programs during this TPACK-based PD.

3.2 Research design

This study adopted a single group pretest and posttest quasi-experimental design, as shown in Fig. 1. The PD activities consisted of two phases: face-to-face and online activities. Before the PD program, teachers were invited to participate in a pretest to evaluate their AI knowledge and AI teaching self-efficacy. After 25 days of blended PD activities, their AI knowledge and AI teaching efficacy were tested again, individual teachers' AI programming work and group AI lesson plans were collected, and semistructured interviews were conducted.

**Fig. 1** The quasi-experimental design

3.3 Intervention: TPACK-based PD

In this research, a TPACK-based PD program was designed to promote the AI teaching competency of CS teachers, especially emphasizing content knowledge of AI (CK_{AI}), technological content knowledge of AI (TCK_{AI}), pedagogical content knowledge of AI (PCK_{AI}), technological pedagogical knowledge of AI (TPK_{AI}), and technological pedagogical content knowledge of AI ($TPACK_{AI}$), as shown in Fig. 2. CK_{AI} concerns the introduction of AI, including the “five big ideas about AI” (Touretzky et al., 2019) and the application of AI. TCK_{AI} concerns block-based programming for AI, including the use of digital software and physical hardware to learn AI. PCK_{AI} consists of two parts, including teaching strategies and instructional design for AI. TPK_{AI} focuses on tools for teaching AI, including related apps, websites, and tools. $TPACK_{AI}$ emphasizes the integration of programming, technologies, and pedagogy for teaching AI, for example, the development of a school-based AI textbook. The specific PD modules and content are described in Table 2.

The whole TPACK-based PD program combined 45 h of face-to-face workshops and a 30-h online course. Considering the effective teacher PD elements—“active learning” and “cooperation”—proposed by Darling-Hammond et al. (2017), most of the face-to-face PD activities followed project-based learning pedagogy, which included hands-on assembly and programming of electronic blocks, design and sharing of lesson plans with peers, observation of classroom teaching practice, and on-site visits, as shown in Fig. 3, with more detailed activities shown in Table 3.

During project-based learning, teachers were required to solve challenges or complete projects in groups of 4 to 5 people, and coaches or experts walked around the lab to assist teachers and provide dynamic scaffolds. For instance,

Fig. 2 TPACK-based PD framework

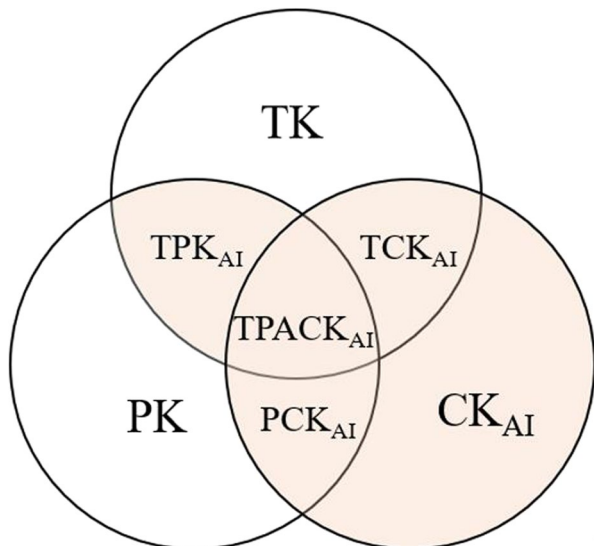


Table 2 The PD modules and content based on the TPACK framework

	Module	Content	Hours
CK _{AI}	Introduction of AI (F2F + Online)	<ul style="list-style-type: none"> • Five big ideas about AI • Application of AI • AI in the Beidou Navigation System • AI ethics 	13
TCK _{AI}	Block-based Programming for AI (F2F + Online)	<ul style="list-style-type: none"> • Use of digital software to learn AI (Scratch, mixly, mDesigner) • Use of physical hardware to learn AI (Arduino, mCookie) 	26
PCK _{AI}	Teaching Strategies for AI (F2F + Online)	<ul style="list-style-type: none"> • Project-based learning in AI education • Problem-based learning in AI education • Game-based learning in AI education • Unplugged activities in AI education (role-playing, storytelling) • Hands-on learning for teachers 	13
	Instructional Design for AI (F2F)	<ul style="list-style-type: none"> • (Robot constructions, training AI models, programming) • Methods and cases for AI lesson plans 	6
TPK _{AI}	Tools for Teaching AI (F2F)	<ul style="list-style-type: none"> • Image recognition apps • Facial recognition apps • Supervised learning websites • Unsupervised learning tools 	5
TPACK _{AI}	Integration of Programming, Technologies, and Pedagogy for Teaching AI (F2F)	<ul style="list-style-type: none"> • Development of a school-based AI textbook 	6



Fig. 3 Active learning and cooperation in face-to-face activities

Table 3 Face-to-face workshop schedule

Activities	
Day 1	· Registration and tour of campus. (5 h)
Day 2	· Opening ceremony. (1 h) · Lectures: (4 h) a) Introduction to AI b) AI in Beidou Navigation c) AI education based on open-source hardware
Day 3	· Ice-breaker activity. (0.5 h) · Block-based AI programming activity. (4.5 h)
Day 4	· Field trips: (5 h) a) AI laboratory of a middle school b) AI laboratory of a university
Day 5	· Break
Day 6	· Hands-on programming activity with hardware: AI in Beidou Navigation. (5 h)
Day 7	· Hands-on programming activity with hardware: AI and Intelligent Internet of Things. (5 h)
Day 8	· Hands-on programming activity with hardware: AI-based artifacts. (5 h)
Day 9	· Lectures: Teaching strategies for AI education in K-12 schools. (1 h) · Workshop: Collaborative lesson design, presentations and sharing. (4 h)
Day 10	· Lectures: (2.5 h) a) Development of school-based AI teaching materials b) Case study of TPD in AI education · Closing ceremony. (2.5 h)



Fig. 4 Online learning tools and activities. Note: (a) table of contents of the AI programming course, including the introduction of AI, the AI expert system, and block-based programming for AI, (b) the community of practice in WeChat, where the participants were discussing how to design a rotated symmetric figure in Python, and (c) two examples of block-based AI programming, which are image recognition and data processing

during the lesson plan activity, 40 teachers were divided into 8 groups, with 5 teachers in each group, and they collaborated together to design an AI lesson plan within groups and finally shared their lesson plan in the whole class.

The online phase for CS teachers' PD lasted 15 days, and it included a self-paced online course and community of practice (CoP) in WeChat. Within the online course, there were 11 video lectures about AI and its programming, as well as 3 assignments for block-based AI programming projects, which ranged from easy to difficult and which focused on emotion recognition, speech recognition, and augmented reality. In the online CoP, participants communicated academic content with each other, shared their programming artifacts and asked for help from their facilitators (Fig. 4).

3.4 Instruments

In this research, an AI knowledge test was developed to measure the change in teachers' AI knowledge before and after the intervention. The evaluation criteria for AI lesson plans and AI programming artifacts were used to measure teachers' AI teaching skills. An AI teaching self-efficacy scale was adopted to assess the improvement of teachers' AI teaching self-efficacy before and after the intervention. Finally, a semistructured interview was used to support and supplement the quantitative data.

3.4.1 AI knowledge tests

The Artificial Intelligence for K-12 Initiative in the United States proposed “5 big ideas” for AI education: perception, representation and reasoning, machine learning, interaction, and social impact (Touretzky et al., 2019). Based on these ideas, an AI knowledge test with two homogeneous versions was developed for the pre- and posttest by a team comprising an AI expert and two researchers, whose opinions were used to support the content validity of this test. Then, an educational technology expert reviewed the test for face validity, including its accuracy and PD appropriateness. For the items in each version of the AI knowledge test, there were 15 multiple choice questions, with 3 items for each of the “5 big ideas”. The test included 10 questions with a single answer, with 1 point for correct answers and 0 for wrong answers, and 5 questions with multiple answers, with 2 points for selecting all the correct answers, 1 point for selecting some of the correct answers and 0 for selecting or including wrong answers. Some examples of the pre- and posttest items are shown in Table 4. The overall Cronbach’s alpha of the AI knowledge tests was 0.608. Although this value is not ideal, since AI knowledge has not been accurately defined in previous research and since no relevant instrument has been released yet, a slightly lower internal consistency is acceptable (Chai et al., 2016). The discrimination of each version of the AI knowledge test ranged from 0.32 to 0.41, and the difficulty coefficient ranged from 0.487 to 0.623.

Table 4 Example items of the AI knowledge test

Tests	Description
Pretest	<ul style="list-style-type: none"> • There is a research field in AI that mainly studies how computers automatically acquire knowledge and skills to achieve self-improvement. This research field is called () • Which one is not part of an expert system () • The uncertainty of AI development has brought many new challenges, including (). *
Posttest	<ul style="list-style-type: none"> • The first step in machine learning is () • Natural language understanding is an important application field of AI. Which one is not its goal () • What safety and ethical problems will arise with the development of AI (). *

Note: * means a multiple-answer choice

Table 5 AI lesson plan evaluation criteria

Category	Item	Sources
Learning objectives	• Be clear and diversified	Kim et al., 2015
	• Combine with students' learning situation	Ye & Wu, 2003
	• Reflect the core literacy of CS	Ministry of education, 2017
Learning content	• Be accurately and specifically described	Kim et al., 2015
	• Embody AI knowledge	Ministry of education, 2017
	• Relate to real life	Ye & Wu, 2003
Learning activities	• Be open and flexible	Ye & Wu, 2003
	• Be consistent with the learning objectives	Kim et al., 2015
	• Reflect project-based learning	Ministry of education, 2017

3.4.2 Evaluation criteria for AI lesson plans

To measure CS teachers' AI lesson plan ability, the researchers developed a rubric of AI lesson plans, including 3 categories and 9 items (as shown in Table 5). The categories included learning objectives, learning content and learning activities, which were adapted from the Lesson Plan Evaluation Criteria for K-12 Education originally developed by Ye and Wu (2003). The specific items were integrated from 3 related rubrics, including the STEM Lesson Plan Evaluation Criteria (Kim et al., 2015), the Information Technology Curriculum Standard for Senior High Schools (Ministry of education, 2017) and the Lesson Plan Evaluation Criteria for K-12 Education. To improve the structural validity of the rubric, the researchers of this study discussed it together, conducted a trial study and then revised it. In addition, the structure and level of the rubric were reviewed for content validity by an educational technology expert, and then researchers refined the final version of the rubric based on expert feedback. The rubric was organized into three sections: learning objectives, learning content and learning activities. The total score of the evaluation criteria is 9, and each item is scored as 1 or 0.

3.4.3 Evaluation criteria for AI programming artifacts

To assess teachers' teaching skills, this study also used block-based programming artifact evaluation criteria to analyze teachers' individual AI programming artifacts; these criteria were originally developed by Moreno-León et al. (2015) and include seven dimensions: flow control, synchronization, parallelism, user interactivity, logical thinking, data representation, abstraction and problem decomposition. The total score is 21, which is divided into three levels: basic (0–7 points), developing (8–14 points) and proficient (15–21 points).

3.4.4 AI teaching self-efficacy scale

Riggs and Enochs (1990) developed a 5-point Likert scale to assess teachers' science teaching efficacy belief; it includes two core dimensions: science teaching efficacy

belief and science teaching outcome expectancy. There are 25 items in this scale, each scored on a scale ranging from 1 (strongly disagree) to 5 (strongly agree). To measure the AI teaching self-efficacy of CS teachers, the scale was adapted with some modifications and deletions in the current research, as shown in Table 6. There are 10 items in total, which reflect the two core dimensions: AI teaching efficacy beliefs and AI teaching outcome expectancy. Among the 10 items, item 4, 6 and item 9 are reversed. It should be noted that the reversed encoding was converted into forward encoding during data analysis. Then, the validity and reliability of the adapted scale were tested with 111 K-12 CS teachers. The KMO value was $0.791 > 0.7$, indicating that there was a certain correlation between the items. The loading values ranged from 0.52 to $0.79 > 0.45$. The Cronbach's alpha coefficient was $0.822 > 0.7$, indicating that the scale had an acceptable stability and consistency.

3.4.5 Semistructured interview outline

For AI teaching competency and the teachers' PD, the semistructured interview focused on the following questions: (1) What are your gains and feelings from these PD activities? (2) Do you feel more confident in carrying out AI-related courses/activities or guiding students in the future? Why? (3) What challenges do you think you may face in carrying out AI-related courses/activities after these PD activities?

3.5 Data collection and analysis

To efficiently collect quantitative data, the AI knowledge test and the AI teaching self-efficacy scale were integrated into one online questionnaire. Meanwhile, the pairwise deletion method was used to screen valid questionnaires (Graham, 2009). After invalid responses were eliminated, 32 valid questionnaires remained, and the valid rate reached 80%. The Shapiro-Wilk test was used to test the normality of change scores since the sample size was less than 50. The results showed that the change scores between the pretest and posttest of AI knowledge ($w = 1.934$, $p = 0.469 > 0.05$) and AI teaching efficacy ($w = 0.972$, $p = 0.561 > 0.05$) showed a normal distribution. Therefore, a paired samples t test was used to analyze the data. To avoid the effect of sample size, Cohen's d was used to indicate the standard difference between the pre- and posttest means. Effect sizes with Cohen's $d > 0.5$, 0.8, 1.2, and 2.0 were considered medium, large, very large, and huge, respectively (Sawilowsky, 2009).

In terms of AI lesson plans, works from 8 groups were collected. To assess these 8 works, 2 lesson plans were first randomly selected and assessed by two evaluators to conduct a trial evaluation. The Cronbach's alpha of the trial evaluation was $0.809 > 0.8$. Then, the two evaluators discussed the differences together and reached a consensus. Finally, all the lesson plans were scored separately by the two evaluators, and the Cronbach's alpha coefficient was 0.904.

In terms of AI programming projects, there were 30 teachers who submitted all three online assignments, up to 90 artifacts in total. To improve the reliability of the assessment, the researchers initially discussed the evaluation criteria together to

Table 6 Factor loading matrix of the AI teaching self-efficacy scale

Category	Item	Description	Component	
			1	2
AI teaching efficacy beliefs	Item 2	I understand AI concepts well enough to be effective in teaching elementary AI	0.895	
	Item 3	I have the necessary skills to teach AI	0.880	
	Item 1	I know the steps necessary to teach AI concepts effectively	0.816	
	Item 6	I don't know what to do to turn students on to AI	0.767	
	Item 5	I am typically able to answer students' AI questions	0.731	
	Item 4	I find it difficult to explain to students why AI works	0.532	
AI teaching outcome expectancy	Item 8	The inadequacy of a student's AI knowledge can be overcome by good teaching		0.855
	Item 10	Teachers with good AI teaching abilities can help some kids learn AI		0.804
	Item 7	When the AI skills of students improve, it is most often due to their teacher having found a more effective teaching approach		0.786
	Item 9	Effectiveness in AI teaching has little influence on the achievement of students with low motivation		0.745

ensure that the two evaluators could completely understand the evaluation criteria. Then, four AI programming artifacts were randomly selected from each project and assessed by the two evaluators to conduct a trial evaluation. The Cronbach's alpha of the trial evaluation was $0.922 > 0.8$, indicating high internal consistency. Then, the two evaluators discussed the results and reached a consensus on the differences. Finally, all 90 programming artifacts were randomly divided into two halves, and 15 teachers' three works were rated by the two evaluators separately.

To conduct semistructured interviews, researchers purposefully selected 4 participants, namely, teacher T1 with high AI teaching self-efficacy and low AI knowledge, teacher T2 with high AI teaching self-efficacy and high AI knowledge, teacher T3 with low AI teaching self-efficacy and low AI knowledge, and teacher T4 with low AI teaching self-efficacy and high AI knowledge. The qualitative data were analyzed following four steps: transcribing the interviews, reading the transcripts, coding the data and interpreting the data (Ruona, 2005). Prior research-driven coding nodes were developed based on AI knowledge, AI teaching skills and AI teaching self-efficacy. Initially, some samples from the interview data were coded by two researchers individually. Then, the two researchers discussed the results together and reached a consensus regarding the codes and subthemes, and finally, they coded all interview data individually again.

4 Results

4.1 RQ1: CS teachers' AI knowledge

The results of the paired samples *t* test of AI knowledge are shown in Table 7. The average score of CS teachers' AI knowledge was 10.88 before the intervention and increased to 13.56 after the intervention. There was a statistically significant difference in the average score of AI knowledge before and after the PD program ($t=5.241$, $p=0.000$, Cohen's $d=1.033$), especially in representation and reasoning ($t=6.723$, $p=0.000$, Cohen's $d=1.107$), interaction ($t=4.274$, $p=0.000$, Cohen's $d=0.701$), and social impact ($t=3.388$, $p=0.002$, Cohen's $d=0.554$).

Table 7 Paired samples *t* test of AI knowledge ($n=32$)

Category	Pretest M±SD	Posttest M±SD	<i>t</i>	<i>p</i>	Cohen's <i>d</i>
Overall	10.88±2.37	13.56±2.06	5.241**	0.000	1.033
Perception	2.09±0.73	1.84±1.02	1.161	0.255	0.220
Representation and reasoning	1.63±0.91	3.00±1.02	6.723**	0.000	1.107
Machine learning	1.56±0.95	1.50±0.95	0.232	0.818	0.044
Interaction	2.88±0.98	3.84±0.81	4.274**	0.000	0.701
Social impact	2.72±0.96	3.38±0.71	3.388**	0.002	0.554

Note. ** $p < 0.01$

Table 8 Descriptive statistics for AI lesson plan scores

Category	Item	Number of lessons that met the criterion among 8	M	SD
Learning objectives	Be clear and diversified	7	0.88	0.35
	Combine with students' learning situation	4	0.50	0.53
	Reflect the core literacy of CS	7	0.88	0.35
Learning content	Be accurately and specifically described	3	0.38	0.52
	Embody AI knowledge	7	0.88	0.35
	Relate to real life	7	0.88	0.35
Learning activities	Be open and flexible	5	0.63	0.52
	Be consistent with the learning objectives	7	0.88	0.35
	Reflect project-based learning	1	0.13	0.35
Total			6.04	3.67

However, there was no significant change in the other two dimensions, including perception ($t = 1.161$, $p = 0.255$, Cohen's $d = 0.220$) and machine learning ($t = 0.232$, $p = 0.818$, Cohen's $d = 0.044$).

4.2 RQ2: CS teachers' AI teaching skills

4.2.1 AI lesson plan

Descriptive statistics of the AI lesson plan scores are shown in Table 8. The mean value of the overall score was 6.04 ($SD = 3.67$). For each item, when the mean value was above 0.5, it indicated that more than half of the lesson plans met the standard. For most groups, the learning objectives were clear and diversified ($M = 0.88$), and most of these objectives reflected the core literacy of CS ($M = 0.88$). The learning content of most groups embodied AI knowledge ($M = 0.88$) and related it to real life ($M = 0.88$). In terms of learning activity design, more than half of the groups' teaching design activities were open and flexible ($M = 0.63$), which was consistent with the learning objectives ($M = 0.88$). However, the description of content was not accurate and specific for most groups ($M = 0.38$), and only one group's learning activity reflected project-based learning ($M = 0.13$).

4.2.2 AI programming

Table 9 shows that the mean values of the overall score of each AI programming project increased gradually, from 10.00 (the first project—emotion recognition) to

Table 9 Descriptive statistics for AI programming skills

	M	SD
#1 Emotion recognition	10.00	.000
#2 Speech recognition	12.93	1.033
#3 Augmented reality	16.76	1.455

12.93 (the second project—speech recognition) and to 16.76 (the third project—augmented reality). Furthermore, the average scores of the first and the second projects were at the developing level (8–14), and the average score of the third project was at the proficient level (15–21). This indicated that the AI programming skills of the CS teachers improved constantly, especially considering that the complexity of the three projects increased gradually.

4.3 RQ3: CS teachers' AI teaching self-efficacy

The results of the paired samples t test are shown in Table 10. The average score of CS teachers' AI teaching self-efficacy was 31.66 before PD and increased to 38.19 after PD. This showed that the AI teaching self-efficacy of CS teachers was significantly improved ($t=6.361$, $p=0.000$, Cohen's $d=1.124$) in the two subdimensions of "AI teaching efficacy belief" ($t=3.123$, $p=0.004$, Cohen's $d=1.164$) and "AI teaching outcome expectancy" ($t=6.577$, $p=0.000$, Cohen's $d=0.551$).

4.4 Interview data analysis results

According to the three research questions, the codes of the interview data were categorized into 3 themes and 5 subthemes. The frequency counts of AI teaching competency dimensions are provided in Table 11. All 4 interviewees expressed positive views about "AI lesson plan ability", with these views expressed 12 times in total; "AI teaching efficacy belief", expressed 12 times; "AI programming skills", expressed 8 times; and "AI knowledge", expressed 7 times.

In terms of AI knowledge, teacher T3 (low AI teaching self-efficacy, low AI knowledge) mentioned, "I have learned a lot of AI knowledge this time. For example, I have seen

Table 10 Paired samples t test of AI teaching self-efficacy ($n=32$)

Category	Pretest M ± SD	Posttest M ± SD	t	p	Cohen's d
Overall	31.66 ± 3.73	38.19 ± 5.29	6.361**	0.000	1.124
AI teaching efficacy belief	17.28 ± 3.43	21.94 ± 3.10	3.123**	0.004	1.164
AI teaching outcome expectancy	14.38 ± 2.20	16.25 ± 2.78	6.577**	0.000	0.551

Note. ** $p < 0.01$

Table 11 Frequency counts of the AI teaching competency dimension

Theme	Subtheme	Frequency of codes	Frequency of participants among 4
AI knowledge	AI knowledge	7	3
AI teaching skills	AI lesson plan ability	12	4
	AI programming skills	8	4
AI teaching self-efficacy	AI teaching efficacy belief	12	4
	AI teaching outcome expectancy	5	3

some hardware with my own eyes, and I have learned how to assemble and program them." Teacher T1 (low AI teaching self-efficacy, high AI knowledge) said, "I didn't know what AI was before this training program. Now I have a general understanding of AI, and I feel more confident in teaching AI."

In terms of AI teaching skills, teacher T4 (low AI teaching self-efficacy, high AI knowledge) said, "Our school has a programming club using block-based software such as Scratch, and I only taught students how to program with software since there was a lack of supportive hardware and poor hands-on practice. After this training, I feel I am particularly interested in AI programming with hardware." Teacher T3 (low AI teaching self-efficacy, low AI knowledge) noted, "After training, I know how to apply Beidou navigation to teach AI in my class, and all the lesson plans provided by the experts can be taught to my students. In addition, I learned to carry out AI instructional design based on the core CS literacy, and I felt that I have made great progress."

Regarding AI teaching self-efficacy, teacher T2 (high AI teaching self-efficacy, high AI knowledge) stated, "When I return to school, I will adjust my CS course, add some AI-related content, one or two lessons or a unit into a semester, carry out some related activities to make AI and block-based programming popular among students." Teacher T4 (low AI teaching self-efficacy, high AI knowledge) said, "I will apply (at school) to buy some Arduino (products) and learn to do some creative works by myself. I have a strong expectation to study further and guide my students to combine their ideas with creation."

5 Discussion

Previous research has indicated that TPACK is an effective knowledge framework for teachers' PD programs (Akyuz, 2018; Archambault & Barnett, 2010; Chai & Koh, 2017); however, few studies have used the TPACK model to promote CS teachers' AI teaching competency. To our knowledge, this study may be the first empirical study to promote the AI teaching competency of K-12 in-service CS teachers using a TPACK-based PD approach. Meanwhile, the findings indicated that this approach improved CS teachers' AI teaching competency, including their AI knowledge, AI teaching skills and AI teaching self-efficacy. These findings, as well as practical implications, are discussed in the following sections.

5.1 TPACK-based PD design and its effectiveness on AI teaching competency

To improve CS teachers' AI teaching competency, this TPACK-based PD program focused on multiple content designs embedded in a series of blended learning activities. In this study, CK_{AI} , PCK_{AI} , TCK_{AI} , TPK_{AI} and $TPACK_{AI}$ were intentionally developed among CS in-service teachers. This new content design extended many previous studies on CS teachers' PD, which have mainly focused on basic CS knowledge, concepts, programming, or computational thinking (Angeli et al., 2016; Chen & Cao, 2022; El-Hamamsy et al., 2021; Kong et al., 2020). In terms of activity

design, this TPACK-based program combined various face-to-face activities with an online teacher learning course, especially stressing that the activities reflected effective TPD elements, including active learning, collaboration, and the use of models and modeling (Darling-Hammond et al., 2017; Desimone, 2009). In particular, these activities in this PD program focused on hands-on assembly and programming and lesson plans with peers.

Regarding the effectiveness of this PD program, first, CS teachers' AI knowledge improved significantly after the 25-day intervention, especially in representation and reasoning ($d=1.107$), interaction ($d=0.701$), and social impact ($d=0.554$). This finding is in line with previous research that found that robot programming activities in a PD program promoted teachers' better understanding of the ideas and technologies in AI (El-Hamamsy et al., 2021). To some extent, this TPACK-based PD program was successful in general; however, teachers' perceptions and machine learning of AI knowledge have not yet been improved. The reason probably lies in the PD content of this program not being strengthened enough regarding the knowledge and activities related to these two AI subtopics.

Secondly, after the intervention, most CS teachers' AI lesson plans met the criteria, especially in the subdimensions of "be clear and diversified", "reflect core literacy of computer science", "embody AI knowledge", "relate to real life", and "be consistent with the learning objectives". This finding indicated that teachers' pedagogical content knowledge related to AI was improved through this TPACK-based PD program. However, project-based learning was not reflected in most groups' lesson plans, which revealed that there was a weakness in this PD program. As a student-centered constructivist approach, project-based learning has often been applied to programming education with positive outcomes (Hsu et al., 2018; Wang & Hwang, 2017). Xue and Wang (2022) also stressed that teachers should reasonably integrate teaching methods and information technology and cultivate students' creative thinking and information literacy in AI courses. Therefore, the weakness of this PD program should be addressed in the future.

Regarding the effect of the program on CS teachers' AI programming skills, the scores of the three programming projects increased gradually, indicating that the TPACK-based PD program had a positive result. Since programming technology is an important element of technological knowledge related to the AI discipline (Kim et al., 2021), it is necessary to purposefully improve teachers' programming skills. Toward this goal, block-based coding sessions, both online and offline, were implemented in this study. This approach is consistent with previous research that used robotics and block-based programming to inspire teachers' creativity and develop their understanding of advanced CT concepts (Kim et al., 2015; El-Hamamsy et al., 2021; Dorotea et al., 2021).

Finally, CS teachers' AI teaching self-efficacy was significantly improved in terms of both "AI teaching outcome expectancy" and "AI teaching efficacy belief" after the intervention. This finding aligns with Kapici and Akcay's (2020) study, which found that lesson planning practice supported by the environment and technical equipment could significantly improve teachers' self-efficacy. In the present study, TPACK-based programs embedded with various professional learning activities, especially group-based lesson planning accompanied by hands-on assembly and programming

electronic fabrics, may play an important role in increasing CS teachers' teaching beliefs. As a result, high AI teaching self-efficacy may have a positive impact on their related teaching practice and students' learning outcomes, according to previous studies on teaching self-efficacy (Burić & Kim, 2020; Orakçı et al., 2020).

5.2 Contributions and implications

To sum up the contributions of this research, we suppose that our PD intervention is one of the first empirical and evidence-based attempts to promote CS teachers' AI competency. The intervention in this study was designed particularly based on the TPACK framework (Rosenberg & Koehler, 2015), and it also integrated certain elements of effective TPD (Darling-Hammond et al., 2017; Desimone, 2009). Another contribution of this study was that we used multiple instruments, including knowledge-based tests, artifact-based assessments (programming artifacts and lesson plans), self-reports, and interview data, to assess CS teachers' AI teaching competency. Compared with previous research that examined the effect of TPACK-based PD programs on CS teachers (Angeli et al., 2016; Chen & Cao, 2022; Kong et al., 2020), the assessment approach in our research was more comprehensive and might improve the reliability and validity of the research.

The main implication of the present study lies in the effective design of CS in-service teachers' PD programs for practitioners. Due to the lack of competent CS teachers to teach AI-related courses in K-12 schools (Akpınar & Bal, 2006), it is urgent to design high-quality PD programs to improve in-service CS teachers' AI teaching competency, including in cognitive and noncognitive aspects. According to effective elements of TPD, first, PD content should focus on "teaching strategies associated with specific curriculum content [that] supports teacher learning within teachers' classroom contexts" (Darling-Hammond et al., 2017). In the present study, this element reflected an intentional focus on AI-specific discipline and constructivist pedagogies, which were mainly categorized into CK_{AI} , TCK_{AI} , PCK_{AI} , TPK_{AI} and $TPACK_{AI}$. The study revealed that this TPACK-based content design had a positive impact on the CS teachers' AI teaching knowledge, skills and self-efficacy in general. Meanwhile, this content design particularly incorporated active teacher learning into the PD program, which engaged CS teachers directly in constructing AI-related artifacts and designing lesson plans based on constructionism theory (Papert, 1991). Furthermore, this PD program created spaces for CS in-service teachers to share ideas and collaborate in their professional learning, which reflected the "collaboration" element of effective PD to some extent. This holistic TPACK-based design may have some practical implications for many CS teachers' PD initiatives in China.

6 Conclusion and limitations

To promote CS teachers' AI competency, this research constructed and implemented a TPACK-based PD program. With a 25-day intervention, including 45 h of offline activities and 30 h of online activities, the TPACK-based PD program

a) significantly improved CS teachers' AI knowledge, especially in representation and reasoning, interaction, and social impact; b) developed CS teachers' AI teaching skills, including their AI lesson plan ability and AI programming skills; and c) significantly improved CS teachers' AI teaching self-efficacy, both in AI teaching efficacy belief and AI teaching outcome expectancy. These findings revealed the positive effectiveness of TPACK-based PD in improving the AI teaching competency of CS teachers in K-12 education, which helps to expand and enrich the existing research and practice on the design of effective PD programs for CS teachers, especially in AI education.

In terms of the limitations of this study, on the one hand, a one-group pretest–posttest quasi-experimental design was employed in the study, which may have made it hard to explain the real effect of the intervention due to the lack of a control group. However, during this TPACK-based intervention, no other PD programs were offered to the participants, which could exclude other confounding variables. In the future, it is suggested that a much more rigorous pretest and posttest (quasi-)experimental design with a control group be conducted to examine the effectiveness of the program. On the other hand, the TPACK-based PD program in this study did not involve all the effective PD elements, such as “coaching and expert support” and “feedback and reflection”, which focus on improving teaching practice in the classroom. In future research, the program could be strengthened through job-embedded PD for CS teachers, and the impact on teaching practice could be examined further. Furthermore, students' AI learning outcomes also need to be assessed to provide much stronger evidence of the PD program's effectiveness.

Acknowledgements The authors would like to express their sincere gratitude to Jialin Yu and Ming Zhao, who provided general help for data coding of this manuscript; and to Hong Wang, who helped to transcribe the interview content of this manuscript.

Author contribution All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Junmei Sun; Hongliang Ma; Yu Zeng; Dong Han; and Yunbo Jin. The first draft of the manuscript was written by Junmei Sun and Hongliang Ma, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

- Conceptualization: Junmei Sun, Hongliang Ma;
- Methodology: Junmei Sun, Hongliang Ma, Yu Zeng, Yunbo Jin;
- Formal analysis and investigation: Junmei Sun, Hongliang Ma, Dong Han, Yunbo Jin;
- Writing—original draft preparation: Junmei Sun, Hongliang Ma;
- Writing—review and editing: Junmei Sun, Hongliang Ma, Yu Zeng, Dong Han

Funding This research was supported by the SNNU' Fundamental Research Funds for the Central Universities (2021TS046) and the key project of basic education research of Xi'an (2021ZB-GG10).

Data availability The datasets generated during and analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

- Akpınar, Y., & Bal, V. (2006). Teachers' collaborative task authoring to help students learn a science unit. *Educational Technology & Society*, 9(2), 84–95.
- Akyuz, D. (2018). Measuring technological pedagogical content knowledge (TPACK) through performance assessment. *Computers & Education*, 125, 212–225. <https://doi.org/10.1016/j.compedu.2018.06.012>
- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society*, 19(3), 47–57.
- Archambault, L. M., & Barnett, J. H. (2010). Revisiting technological pedagogical content knowledge: Exploring the TPACK framework. *Computers & Education*, 55(4), 1656–1662. <https://doi.org/10.1016/j.compedu.2010.07.009>
- Armoni, M. (2017). Computing in schools on teaching problem-solving paradigms in K-12 education. *ACM Inroads*, 8(4), 30–32. <https://doi.org/10.1145/3148533>
- Bandura, A. (1997). *Self efficacy: The exercise of control*. Freeman.
- Baran, B. (2010). Facebook as a formal instructional environment. *British Journal of Educational Technology*, 41(6), E146–E149. <https://doi.org/10.1111/j.1467-8535.2010.01115.x>
- Baumert, J., & Kunter, M. (2013). The COACTIV model of teachers' professional competence. In M. Kunter, J. Baumert, W. Blum, U. Klusmann, S. Krauss, & M. Neubrand (Eds.), *Cognitive activation in the mathematics classroom and professional competence of teachers* (pp. 25–48). Springer.
- Blonder, R., Jonatan, M., Bar-Dov, Z., Benny, N., Rap, S., & Sakhnini, S. (2013). Can You Tube it? Providing chemistry teachers with technological tools and enhancing their self-efficacy beliefs. *Chemistry Education Research and Practice*, 14(3), 269–285. <https://doi.org/10.1039/c3rp00001j>
- Brandes, O., & Armoni, M. (2019). The effects of a professional development workshop focusing on action research on the practice of high-school computer science teachers. *Proceedings of the 14th Workshop in Primary and Secondary Computing Education*, 1–10. <https://doi.org/10.1145/3361725.21.3361725>
- Burić, I., & Kim, L. E. (2020). Teacher self-efficacy, instructional quality, and student motivational beliefs: An analysis using multilevel structural equation modeling. *Learning and Instruction*, 66, 101302. <https://doi.org/10.1016/j.learninstruc.2019.101302>
- Caena, F. (2014). Teacher competence frameworks in Europe: Policy-as-discourse and policy-as-practice. *European Journal of Education*, 49(3), 311–331. <https://doi.org/10.1111/ejed.12088>
- Chai, C. S., & Koh, J. H. L. (2017). Changing teachers' TPACK and design beliefs through the Scaffolded TPACK Lesson Design Model (STLDM). *Learning: Research and Practice*, 3(2), 114–129. <https://doi.org/10.1080/23735082.2017.1360506>
- Chai, C. S., Koh, J. H. L., & Tsai, C. C. (2016). A review of the quantitative measures of technological pedagogical content knowledge (TPACK). In M. C. Herring, M. J. Koehler, & P. Mishra (Eds.), *Handbook of technological pedagogical content knowledge (TPACK) for educators* (pp. 131–160). Routledge.
- Chai, C. S., Rahmawati, Y., & Jong, M. S. Y. (2020). Indonesian science, mathematics, and engineering preservice teachers' experiences in STEM-TPACK design-based learning. *Sustainability*, 12(21), 9050. <https://doi.org/10.3390/su12219050>
- Chen, Y., & Cao, L. (2022). Promoting maker-centred instruction through virtual professional development activities for K-12 teachers in low-income rural areas. *British Journal of Educational Technology*, Advance online publication. <https://doi.org/10.1111/bjjet.13183>
- Darling-Hammond, L., Hyler, M. E., Gardner, M. (2017). *Effective teacher professional development*. Palo Alto, CA: Learning Policy Institute. <https://doi.org/10.54300/122.311>
- Desimone, L. M. (2009). Improving impact studies of teachers' professional development: Toward better conceptualizations and measures. *Educational Researcher*, 38(3), 181–199. <https://doi.org/10.3102/0013189X08331140>
- Dorotea, N., Piedade, J., & Pedro, A. (2021). Mapping K-12 computer science teacher's interest, self-confidence, and knowledge about the use of educational robotics to teach. *Education Sciences*, 11(8), 443. <https://doi.org/10.3390/educsci11080443>
- El-Hamamsy, L., Chessel-Lazzarotto, F., Bruno, B., Roy, D., Cahlikova, T., Chevalier, M., Parriaux, G., Pellet, J. P., Lanares, J., Zufferey, J. D., & Mondada, F. (2021). A computer science and robotics integration model for primary school: Evaluation of a large-scale in-service K-4 teacher-training

- program. *Education and Information Technologies*, 26(3), 2445–2475. <https://doi.org/10.1007/s10639-020-10355-5>
- Goel, A. K., & Joyne, D. A. (2017). Using AI to teach AI: Lessons from an online AI class. *Ai Magazine*, 38(2), 48–59. <https://doi.org/10.1609/aimag.v38i2.2732>
- Goode, J., Peterson, K., Malyn-Smith, J., & Chapman, G. (2020). Online professional development for high school computer science teachers: Features that support an equity-based professional learning community. *Computing in Science & Engineering*, 22(5), 51–59. <https://doi.org/10.1109/MCSE.2020.2989622>
- Graham, J. W. (2009). Missing data analysis: Making it work in the real world. *Annual Review of Psychology*, 60, 549–576. <https://doi.org/10.1146/annurev.psych.58.110405.085530>
- Henry, J., Hernalesteen, A., & Collard, A. S. (2021). Teaching artificial intelligence to K-12 through a role-playing game questioning the intelligence concept. *Kunstliche Intelligenz*, 35(2), 171–179. <https://doi.org/10.1007/s13218-021-00733-7>
- Hong, J. E., & Stonier, F. (2015). GIS in-service teacher training based on TPACK. *Journal of Geography*, 114(3), 108–117. <https://doi.org/10.1080/00221341.2014.947381>
- Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296–310. <https://doi.org/10.1016/j.compedu.2018.07.004>
- Kandlhofer, M., Steinbauer, G., Laßnig, J. P., Baumann, W., Plomer, S., Ballagi, A., & Alfoldi, I. (2019). Enabling the creation of intelligent things: Bringing artificial intelligence and robotics to schools. *Proceedings of 2019 IEEE Frontiers in Education Conference (FIE)* (pp. 1–5). <https://doi.org/10.1109/FIE43999.2019.9028537>
- Kapici, H. O., & Akcay, H. (2020). Improving student teachers' TPACK self-efficacy through lesson planning practice in the virtual platform. *Educational Studies*, 1–23. <https://doi.org/10.1080/03055698.2020.1835610>
- Kim, B. H., & Kim, J. (2016). Development and validation of evaluation indicators for teaching competency in steam education in Korea. *Eurasia Journal of Mathematics Science & Technology Education*, 12(7), 1909–1924. <https://doi.org/10.12973/eurasia.2016.1537a>
- Kim, C., Kim, D., Yuan, J., Hill, R. B., Doshi, P., & Thai, C. N. (2015). Robotics to promote elementary education pre-service teachers' STEM engagement, learning, and teaching. *Computers & Education*, 91, 14–31. <https://doi.org/10.1016/j.compedu.2015.08.005>
- Kim, S., Jang, Y., Choi, S., Kim, W., Jung, H., Kim, S., & Kim, H. (2021). Analyzing teacher competency with TPACK for K-12 AI education. *Kunstliche Intelligenz*, 35(2), 139–151. <https://doi.org/10.1007/s13218-021-00731-9>
- Koehler, M. J., Mishra, P., & Yahya, K. (2007). Tracing the development of teacher knowledge in a design seminar: Integrating content, pedagogy and technology. *Computers & Education*, 49(3), 740–762. <https://doi.org/10.1016/j.compedu.2005.11.012>
- Kong, S. C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, 151, 103872. <https://doi.org/10.1016/j.compedu.2020.103872>
- Koster, B., Brekelmans, M., Korthagen, F., & Wubbels, T. (2005). Quality requirements for teacher educators. *Teaching and Teacher Education*, 21(2), 157–176. <https://doi.org/10.1016/j.tate.2004.12.004>
- Kunter, M., & Baumert, J. (2013). The COACTIV research program on teachers' professional competence: Summary and discussion. In M. Kunter, J. Baumert, W. Blum, U. Klusmann, S. Krauss, & M. Neubrand (Eds.), *Cognitive activation in the mathematics classroom and professional competence of teachers* (pp. 345–368). Springer.
- Lauermann, F., & König, J. (2016). Teachers' professional competence and wellbeing: Understanding the links between general pedagogical knowledge, self-efficacy, and burnout. *Learning and Instruction*, 45, 9–19. <https://doi.org/10.1016/j.learninstruc.2016.06.006>
- Lazarinis, F., Karachristos, C. V., Stavropoulos, E. C., & Verykios, V. S. (2019). A blended learning course for playfully teaching programming concepts to school teachers. *Education and Information Technologies*, 24(2), 1237–1249. <https://doi.org/10.1007/s10639-018-9823-2>
- Margolis, J., Goode, J., & Bernier, D. (2011). The need for computer science. *Educational Leadership*, 68(5), 68–72.
- Martinez, M. C., Gomez, M. J., Moresi, M., & Benotti, L. (2016). Lessons learned on computer science teachers professional development. *Proceedings of the 2016 ACM conference on innovation and technology in computer science education* (pp. 77–82). <https://doi.org/10.1145/2899415.2899460>

- Ministry of Education of the People's Republic of China. (2017). *Information technology standard for high school*. People's Education Press.
- Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6), 1017–1054. <https://doi.org/10.1111/j.1467-9620.2006.00684.x>
- Moin, L. J., Dorfield, J. K., & Schunn, C. D. (2005). Where can we find future K-12 science and math teachers? a search by academic year, discipline, and academic performance level. *Science Education*, 89(6), 980–1006.
- Monjelat, N., & Lantz-Andersson, A. (2019). Teachers' narrative of learning to program in a professional development effort and the relation to the rhetoric of computational thinking. *Education and Information Technologies*, 25(3), 2175–2200. <https://doi.org/10.1007/s10639-019-10048-8>
- Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista De Educación a Distancia*, 46, 1–23.
- Murai, Y., & Muramatsu, H. (2020). Application of creative learning principles within blended teacher professional development on integration of computer programming education into elementary and middle school classrooms. *Information and Learning Sciences*, 121(7/8), 665–675. <https://doi.org/10.1108/ILS-04-2020-0122>
- Ng, D. T. K., Leung, J. K. L., Chu, S. K. W., & Qiao, M. S. (2021). Conceptualizing AI literacy: An exploratory review. *Computers and Education: Artificial Intelligence*, 2, 100041. <https://doi.org/10.1016/j.caeai.2021.100041>
- Oda, K., Herman, T., & Hasan, A. (2020). Properties and impacts of TPACK-based GIS professional development for in-service teachers. *International Research in Geographical and Environmental Education*, 29(1), 40–54. <https://doi.org/10.1080/10382046.2019.1657675>
- OECD. (2009). *Creating effective teaching and learning environments first results from TALIS*. OECD, Paris. Retrieved January 15, 2021, from <http://www.oecd.org/dataoecd/17/51/43023606.pdf>
- Orakçı, Ş., Dilekli, Y., & Erdağ, C. (2020). The structural relationship between accountability felt and responsible teaching in Turkish teachers: The mediating effect of innovative thinking. *Thinking Skills and Creativity*, 36, 100662. <https://doi.org/10.1016/j.tsc.2020.100662>
- Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 1–14). Lawrence Erlbaum Associates.
- Prajugjit, M., & Kaewkuekool, S. (2020). Development of trainer competency model: A case study at Thailand's Department of Skill Development. *International Journal of Innovation and Learning*, 28(2), 206–238. <https://doi.org/10.1504/IJIL.2020.108971>
- Qian, Y. Z., Hambrusch, S., Yadav, A., & Gretter, S. (2018). Who needs what: Recommendations for designing effective online professional development for computer science teachers. *Journal of Research on Technology in Education*, 50(2), 164–181. <https://doi.org/10.1080/15391523.2018.1433565>
- Reimer, Y. J., & Blank, L. (2018). Teacher (as student) driven curriculum development for an introductory computer science course. *Journal of Computing Sciences in Colleges*, 34(1), 216–222. <https://doi.org/10.5555/3280489.3280525>
- Rich, P. J., Larsen, R. A., & Mason, S. L. (2021). Measuring teacher beliefs about coding and computational thinking. *Journal of Research on Technology in Education*, 53(3), 296–316. <https://doi.org/10.1080/15391523.2020.1771232>
- Riggs, I. M., & Enochs, L. G. (1990). Toward the development of an elementary teacher's science teaching efficacy belief instrument. *Science Education*, 74(6), 625–637. <https://doi.org/10.1002/sce.3730740605>
- Rosenberg, J. M., & Koehler, M. J. (2015). Context and technological pedagogical content knowledge (TPACK): A systematic review. *Journal of Research on Technology in Education*, 47(3), 186–210. <https://doi.org/10.1080/15391523.2015.1052663>
- Ruona, W. E. A. (2005). Analyzing qualitative data. In R. A. Swanson & E. F. Holton (Eds.), *Foundations and methods of inquiry* (pp. 223–263). Berrett-Koehler.
- Sawilowsky, S. S. (2009). New effect size rules of thumb. *Journal of Modern Applied Statistical Methods*, 8(2), 597–599.
- Sentance, S., Sinclair, J., Simmons, C., & Csizmadia, A. (2018). Classroom-based research projects for computing teachers: Facilitating professional learning. *ACM Transactions on Computing Education (TOCE)*, 18(3), 1–26. <https://doi.org/10.1145/3171129>

- Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, 15(2), 4–14. <https://doi.org/10.30827/profesorado.v23i3.11230>
- Tokmak, H. S., Incikabi, L., & Ozgelen, S. (2013). An investigation of change in mathematics, science, and literacy education pre-service teachers' TPACK. *The Asia-Pacific Education Researcher*, 22(4), 407–415. <https://doi.org/10.1007/s40299-012-0040-2>
- Touretzky, D. S., Gardner-McCune, C., Martin, F., & Seehorn, D. (2019). K-12 guidelines for artificial intelligence: what students should know. *Paper presented at the 2019 ISTE Conference*.
- Tschannen-Moran, M., & Hoy, A. W. (2001). Teacher efficacy: Capturing an elusive construct. *Teaching and Teacher Education*, 17(7), 783–805. [https://doi.org/10.1016/S0742-051X\(01\)00036-1](https://doi.org/10.1016/S0742-051X(01)00036-1)
- Tschannen-Moran, M., & Hoy, A. W. (2007). The differential antecedents of self-efficacy beliefs of novice and experienced teachers. *Teaching and Teacher Education*, 23(6), 944–956. <https://doi.org/10.1016/j.tate.2006.05.003>
- van Doodewaard, C. (2020). They may be a level one student, but I love them all: Preservice teachers negotiating inclusion and diversity in their classes. *Journal of Teaching in Physical Education*, 40(4), 547–555. <https://doi.org/10.1123/jtpe.2020-0042>
- Vazhayil, A., Shetty, R., Bhavani, R. R., & Akshay, N. (2019). Focusing on teacher education to introduce AI in schools: Perspectives and illustrative findings. *Proceedings of 2019 IEEE Tenth International Conference on Technology for Education (T4E)* (pp. 71–77). <https://doi.org/10.1109/T4E.2019.00021>
- Wang, X. M., & Hwang, G. J. (2017). A problem posing-based practicing strategy for facilitating students' computer programming skills in the team-based learning mode. *Educational Technology Research & Development*, 65(6), 1655–1671. <https://doi.org/10.1007/s11423-017-9551-0>
- Xue, Y., & Wang, Y. (2022). Artificial intelligence for education and teaching. *Wireless Communications and Mobile Computing*. Advance online publication. <https://doi.org/10.1155/2022/4750018>
- Ye, L., & Wu, Y. P. (2003). Reform of classroom teaching and reform of classroom teaching evaluation. *Educational Research*, 08, 42–49.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.