

# Intelligent Support for All? A Literature Review of the (In)equitable Design & Evaluation of Adaptive Pedagogical Systems for CS Education

Alexia Charis Martin  
a.martin1@ufl.edu  
University of Florida  
Gainesville, Florida, USA

Kimberly Michelle Ying  
kimying@ufl.edu  
University of Florida  
Gainesville, Florida

Fernando J. Rodríguez  
fernando@eae.utah.edu  
University of Utah  
Salt Lake City, Utah

Christina Suzanne Kahn  
christinakahn@ufl.edu  
University of Florida  
Gainesville, Florida

Kristy Elizabeth Boyer  
keboyer@ufl.edu  
University of Florida  
Gainesville, Florida

## ABSTRACT

The computer science education community has created many adaptive feedback tools and intelligent tutoring systems to improve students' experience in computing-related courses. However, the extent to which these systems—which we collectively refer to as *adaptive pedagogical systems*—support equitable outcomes for learners of all genders and racial identities is not known. We conducted a systematic literature review of SIGCSE, ITiCSE, and ICER publications on adaptive pedagogical systems in computing courses from the last five years. The results reveal that not only is there little to no data on the effectiveness of adaptive pedagogical systems for CS education by gender or race, the vast majority of published papers reporting on these systems do not even include the demographics of their users. Based on these findings, this position paper makes a call to action: we must include the voices of historically marginalized students in the design and evaluation of our software, lest we continue to perpetuate that marginalization. We highlight key ideas that every CS education researcher should consider when designing and evaluating technologies to support learners. We argue that this community must hold ourselves and each other accountable to create technologies that support learners equitably.

## CCS CONCEPTS

• **Social and professional topics** → **Computer science education**; *Race and ethnicity*; *Gender*; • **General and reference** → **Surveys and overviews**.

## KEYWORDS

Intelligent tutoring systems, adaptive feedback tools, equity, race, gender

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
*SIGCSE 2022, March 3–5, 2022, Providence, RI, USA*

© 2022 Association for Computing Machinery.  
ACM ISBN 978-1-4503-9070-5/22/03...\$15.00  
<https://doi.org/10.1145/3478431.3499418>

## ACM Reference Format:

Alexia Charis Martin, Kimberly Michelle Ying, Fernando J. Rodríguez, Christina Suzanne Kahn, and Kristy Elizabeth Boyer. 2022. Intelligent Support for All? A Literature Review of the (In)equitable Design & Evaluation of Adaptive Pedagogical Systems for CS Education. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2022)*, March 3–5, 2022, Providence, RI, USA. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3478431.3499418>

## 1 INTRODUCTION

There has been a significant increase in enrollment in CS majors over the last several years. According to a survey conducted by the Computing Research Association (CRA), CS1 course enrollments increased 158% for majors and 169% for non-majors from 2010 to 2015 [7, 29]. Consistent with this trend, the CRA's 2020 Taulbee Survey found that there was a 15.7% increase of CS Bachelor's degrees awarded from the previous year [41]. This enrollment surge has caused an increasing difficulty for providing individualized attention to each learner, which has contributed to the notorious attrition rates among CS and computing-related majors [2, 33]. To address the need for individualized attention, instructors have introduced adaptive feedback tools and intelligent tutoring systems in their courses. In this paper, we use the term *adaptive pedagogical systems* (APS) to collectively refer to these adaptive feedback tools and intelligent tutoring systems.<sup>1</sup> Integrating these tools into computing classrooms has resulted in mitigating student frustration, improvements to learning gain and course satisfaction [28], as well as greater course retention [27].

In recent years, the CS education research community has come to understand that students' backgrounds can influence their experiences in the classroom [4, 25, 34, 40]. Underrepresentation of minorities in computer science has been a continuing concern, consistent with other STEM fields. Furthermore, the gender gap in computing is a well-known issue with several proposed remedies

<sup>1</sup>In reporting and discussing the individual studies in this literature review, we adopt the same terminology as the authors of each paper to refer to their systems. An intelligent tutoring system is a system that "provides learner-tailored support during the problem solving process, as a human tutor would do" [22]. Intelligent tutoring systems typically use AI-based predictive models to provide feedback for the learner on individual problems as well as present problems according to what concepts the learner still needs to improve on [5]. In contrast, adaptive feedback tools provide feedback based on the most recent student action or answer, but do not adapt future problems based on a model of the student [16].

over the years. Women have gendered experiences based on society and cultural norms which foster “different knowledge and ways of knowing” [3]. According to the 2020 Taulbee Survey, only 20.6% of computer science and 16.6% of computer engineering Bachelor’s degrees were awarded to women [41]. There is a wealth of publications focusing on increasing the participation of women in computing going back nearly half a century [10], and more recent but robust discussions of improving racial diversity within the field [9, 12, 15]. As of 2020, the racial/ethnic distribution of CS bachelor’s degree graduates was still highly skewed, with 40.7% White, 28.8% Asian, 8.5% Hispanic, 3.1% Black/African American, and 3.8% other race/ethnicity. This lack of diversity propagates into every aspect of computer science, from academia to the workforce. A central goal of this paper is to inform the CS Ed community on how overlooking minorities in CS classrooms is perpetuating marginalization in CS. Neglecting diverse perspectives during a system’s feature development can cause adverse effects for marginalized communities and perpetuate inequities [30]. For instance, the filter bubbles and algorithms supporting Google’s query functionality were exposed to be wildly racist, characterizing images of black people with a “gorillas” label. This finding, among many, can be credited to the racial homogeneity and insularity of AI developers [26].

Today’s CS classrooms include students with disabilities, students from racial minority groups, students with English as a second language, and other identities. By exploring these identities with an aim to specifically understand the perspective and needs of diverse students, researchers can make informed decisions for the creation of systems. We set out to determine the extent to which this has been done by CS Ed researchers who develop APS. This paper reports on a systematic literature review that investigated the following research questions:

- RQ1:** To what extent are researchers reporting the demographics of the participants who evaluated their adaptive pedagogical systems?
- RQ2:** To what extent are student demographics considered during system design?
- RQ3:** In what ways are student demographics considered during system evaluation?

We conducted a systematic literature review of papers discussing APS published between January 2015 and March 2021 at the following conferences, which are central to the CS Ed community: **SIGCSE:** Special Interest Group on CS Education, **ITiCSE:** Innovation and Technology in CS Education, and **ICER:** International Computing Education Research. We categorized each paper based on the extent to which participant demographics were considered. The analysis revealed that the majority of the publications did not report gender or racial demographics. Of those that did report participant demographics, only one included analyses to evaluate whether the system provided equitable experiences. This literature review reveals a misalignment between our community’s principles and practices when it comes to improving racial and gender diversity in computer science, and it emphasizes the importance of not only providing participant demographic information, but also evaluating whether these systems are equitable.

## 2 METHODS

This section describes our systematic literature review process, including the search string we used and the inclusion/exclusion criteria. This position paper and its literature review are, to the best of the authors’ knowledge, the first literature review to examine APS through the lens of equitable design processes and outcomes. In 2016, Keuning et al. [16] conducted a related literature review that examined 102 publications reporting on 69 automated feedback tools for programming exercises. Their review evaluated the type of feedback provided to students, intelligent tutoring system technique, feedback adaptability, and the quality/correctness of the feedback for the programming exercises. This paper provides an updated survey of publications on intelligent tutoring systems and adaptive feedback tools for CS Ed.

On March 30th, 2021, one researcher searched for full papers in the Association for Computing Machinery (ACM) Digital Library using the following search string:

(adaptive OR smart OR intelligent) AND (system OR tool)

The following proceedings were searched:

- 2015-2021 *Proceedings of the ACM Technical Symposium on Computer Science Education* (SIGCSE TS)
- 2015-2020 *Proceedings of the International Conference on Innovation and Technology in Computer Science Education* (ITiCSE)
- 2015-2020 *Proceedings of the ACM Conference on International Computing Education Research* (ICER)

The initial search resulted in 43 conference publications. The first author reviewed the publications through an iterative process to remove publications that did not meet the inclusion criteria. First, one researcher read the titles and abstracts for all publications and eliminated those publications that were unrelated to our research questions; these topics include automated assessment tools, intelligent plagiarism detectors, curriculum module design, and practitioner-focused practices and frameworks. They then read through the methods and results sections of the remaining publications to understand the authors’ study design and findings from students’ APS intervention. At this stage, we removed papers that did not investigate the design or use of APS for student instruction. Lastly, we read through the full text of remaining publications. After reading the full text of each publication and eliminating all false positives, our review resulted in 14 publications that are discussed in detail throughout the remainder of the paper. We also reviewed the references of the collected publications to check and analyze previous publications written by the authors that could include evaluations of earlier versions of the APS being reported.

## 3 RESULTS

The final 14 publications were categorized based on their inclusion of participant demographic information in student evaluations of their tool. The three categories include publications which report (1) no gender or racial demographics, (2) gender demographics of the participants, and (3) both gender and racial demographics of the participants, as shown in Table 1. The following section details the 14 publications, which include the design and/or evaluation of 13 APS.<sup>2</sup> These publications reported on a range of topics, including

<sup>2</sup>Two publications are published by the same authors who are reporting study outcomes from the same tool but have different findings.

theoretical approaches behind learner-adaptive actions and user experience design for various APS-assisted instruction strategies.

Adaptive pedagogical systems are evaluated to measure their impact on students' learning gain and satisfaction. These evaluations may come in the form of controlled intervention studies, or surveys and assessments which students complete after engaging with the APS. Evaluations may provide educators confidence in the system's abilities to yield beneficial learning outcomes. Ultimately, these evaluation results guide future improvements for the system and can inform general design principles for pedagogical tools in computer science.

### 3.1 No Gender or Racial Demographics Reported

Of the 14 final publications, nine included student evaluations of their system but did not report the demographics of participants. Researchers utilized pre- and post-tests, satisfaction surveys, and comparative analysis to measure the effects of their tools. They found their tools improved learning gain [14, 19, 32], CS major retention [23], and motivation to understand the curriculum [11].

**RedBlackTree Tutor.** Xhakaj and Liew [38] assessed their RedBlackTree Tutor, a web-based intelligent tutoring system that assisted students during laboratory exercises following a granularity approach to red-black tree instruction. The authors used the phrase 'granularity approach' to describe an instructional approach in which students were given several micro-exercises to show step-by-step changes as they worked towards their final solution. To evaluate the RedBlackTree Tutor, 12 students participated in a one-hour laboratory session using the tool. They completed identical pre- and post-tests, which measured the student's ability to identify the current node at a particular step, select the rule for an exercise, and apply the rule. The average pre-test score was 40.83 (out of 75), which increased to 61.08 on the post-test. Additionally, eight of the 12 participants had a post-test score increase of 30% or more. Their findings suggest that RedBlackTree Tutor effectively improved students' learning of red-black trees. We found this publication did not report race or gender demographic information of participants (RQ1) nor conduct any analyses on race or gender in their results

(RQ3). In fact, the researchers solely reported that participants were recruited from a Fall 2014 Data Structures course comprised of 'mostly computer science and computer engineering majors.'

**SIAL.** Maestro-Prieto and Simon-Hurtado developed SIAL to administer obligatory and reinforcement exercises followed by corrective feedback that notified the student if their submission was erroneous. SIAL is an adaptive intelligent tutoring system used to assist students with learning computational logic [21]. The obligatory exercises were given to students as the minimum curriculum while the reinforcement exercises were administered to encourage the students to master certain concepts. SIAL also featured learner-adaptive tutorial actions directly dependent on the student's evolving learning path to dictate the exercise sequencing. The researchers recruited 32 participants from an undergraduate level computational logic course who used the intelligent tutoring system for the duration of one semester, giving them access to a series of 59 reinforcement exercises. The participants completed the exercises, with 22 of the 32 also completing satisfaction surveys. The results indicated that students needed an average of 13.31 reinforcement exercises to understand a concept. Of the 22 survey respondents, 20 found SIAL to be "helpful" or "very helpful" in the understanding of their course content. Overall, this publication did not report any demographic information in their student evaluation (RQ1) nor did they conduct an analysis on how participants' race or gender impacted their results (RQ3). The authors presented details on preliminary testings of SIAL in a previous publication that describes how the pedagogical model would adapt for each student but demographics were also excluded in this publication [20].

**McCartin-Lim et al.** [24] created a software tutoring system that transforms traditional pen and paper discrete proof assignments into puzzle-like exercises with immediate feedback. The tool was offered to students as an optional study aid during two subsequent semesters in an undergraduate algorithms course in Fall 2016 and an undergraduate discrete mathematics course in Spring 2017. Those who chose to use the tool were part of the experimental group ( $n=59$ ). They watched a brief tutorial video to learn how to use the tool, solved proof problems, completed a survey, and uploaded a video recording of their tool intervention. Those who

**Table 1: List of analyzed publications on adaptive pedagogical systems.**

Venue	Year	Author(s)	APS	Demographics Reported
ITICSE	'15	Xhakaj et al. [38]	RBT Tutor	None
ITICSE	'18	Maestro-Prieto and Simon-Hurtado [21]	SIAL	None
ITICSE	'18	Geck et al. [11]	ILTIS	None
SIGCSE	'20	Khosravi et al. [17]	RIPPLE	None
SIGCSE	'18	McCartin-Lim et al. [24]	tutoring system for proof construction	None
SIGCSE	'18	Kumar [19]	Epplets	None
SIGCSE	'17	Harsley et al. [13]	ChiQat-Tutor	None
SIGCSE	'16	Harsley et al. [14]	ChiQat-Tutor	None
SIGCSE	'17	Price et al. [32]	iSnap	None
ITICSE	'15	Alshammari et al. [1]	adaptive e-learning system	Gender
ITICSE	'19	Singh and Meyer [35]	annotation tool for Python programming	Gender
ICER	'19	YeckehZaare et al. [39]	adaptive retrieval practice tool	Gender & Race
ICER	'20	Marwan et al. [23]	adaptive immediate feedback system	Gender & Race
SIGCSE	'21	Wiggins et al. [37]	PRIME	Gender & Race

chose not to use the tool, the control group ( $n=58$ ), were given identical proof problems to solve without the tool and they submitted their answers as PDF files. The results from the student evaluation indicated that their system did not result in any significant results compared to the traditional proof assignments for three of the four proof problems. Students had an improved understanding of proof concepts, with 88% of participants in the experimental group scoring a perfect score compared to 27% from the control group. Additionally, 46.9% of the participants in the experimental group were satisfied with the system and found it helpful for their learning experience.

**ILTIS.** Geck et al. [11] introduced students to a web-based interactive system that developed the students' ability to translate natural language statements into propositional logic formulas. ILTIS provided feedback on the student's proposed formula with an explanation of why certain variables in the solution were (in)correct. To evaluate the impact of ILTIS, a controlled study with three experimental groups was conducted within a sophomore-level introductory logic course. The first experimental group received feedback from ILTIS ( $n=43$ ), the second group was administered a video demo on how to model statements using propositional logic ( $n=98$ ), the third group was shown the demo and used ILTIS ( $n=51$ ), and the control group received no feedback ( $n=57$ ). The results showed that the error rate for the most commonly made error dropped from .42 to .02 for a particular exercise for students in the condition that received ILTIS. Additionally, 74.6% of participants found the system to be "good" or "very good". The lack of racial and gender reporting and analysis makes these findings less generalizable to diverse learners.

**Epplets.** Kumar [19] presented an experience report where students used a Java web-based adaptive application to solve Parsons puzzle problems with single and nested if-else statements. The application provided feedback to students when their solutions were incorrect by highlighting incorrect lines of code and explaining why their solution was erroneous. Although the number of participating students was not reported in this work, the researchers stated the data analyzed was "from students who had given permission for their data to be used for research purposes" from the classrooms of 24 instructors from 2015 to 2017. The researchers reported their tool allowed students to solve Parsons puzzle problems in less time with better accuracy.

**Ripple.** Khosravi et al. [17] developed an adaptive learning system that recommended personalized activities and instructional resources to learners. A few notable features include 1) the representation of the student's knowledge state compared against the domain model, 2) the student's ability to rate the usefulness of certain activities and resources, 3) learner-specific content recommendations based on expected learning outcomes for the student, and 4) gamification features such as badges and leader boards. The researchers presented a pilot study in which students from a database principles course interacted with Ripple and answered a survey. The primary focus of these findings is the presentation of lessons learned from 56 responses to a Likert statement survey. They found that the open learner model increased engagement for a majority of participants yet led to others not completing certain activities in fear of lowering their knowledge rating. Additionally, the incorporation of gamification improved engagement and learning.

**ChiQat-Tutor.** Harsley et al. [14] detailed the system design of their tool, ChiQat-Tutor, which aids the learner in understanding data structures and algorithms through analogies and worked out examples (WOE). This intelligent tutoring system had three unique implementations with each having its own experimental group: WOE ( $n=23$ ), analogy-based WOE ( $n=21$ ), and a combination of WOE and analogy-based WOE ( $n=22$ ). The 40-minute tool intervention was preceded by a 10-minute pre-test and concluded with an identical post-test. The evaluation indicated that students who engaged with the standard implementation of WOE had greater learning gains compared to the students who engaged with implementations that included analogies. In another work, researchers conducted a comparative analysis between individual programmers and pair programmers using the same intelligent tool. Learning gain, engagement, time spent on problems, and system efficacy were measured to assess the students' code efficiency. Over 53,000 interactions were analyzed from 116 participants. Yet, there was no significant difference on learning gain between individual programmers and pair programmers [13].

**iSNAP.** Price et al. [32] reported on a pilot study of their adaptive programming environment extension, iSnap, which provided students automatically generated hints to guide their next action for block-based coding assignments. Previous correct student submissions were used to generate hints for students during their single lab session. Participants were recruited from a Spring 2016 introductory CS course for non-majors ( $n=62$ ). Of the 62 students, 33 made at least one hint request provided by iSnap, and 23 students made at least three. Additionally, 13 participants that acted on at least two of the requested hints achieved seven of the nine assignment objectives. This indicated that students are generally willing to use hints and that hints can create positive scaffolding outcomes.

### 3.2 Gender Demographics Reported

Of the 14 publications, two reported only on gender demographics—no racial demographics—of the study participants.

**Alshammari et al.** [1] described the learning style adaptivity, user interface, and experimental outcomes of their e-learning system. The learning style adaptivity of their system was derived from the Felder-Silverman model which classifies a student's information perception style as sensory or intuitive. The adaptation model for the system was based on the theory that the way one receives instructional material should match one's learning style. They conducted a study with 60 male participants to measure learning gain and student satisfaction from using the system. There were 29 participants in the matched group and 31 in the unmatched group. Of the participants, 72% had sensory characteristics while 28% had intuitive characteristics. The pre- and post-test administered to the participants in the matched group had an improved score of 33, while the unmatched group had an improved score of 20. Furthermore, a Likert-scale questionnaire was given to participants to measure their satisfaction with the learner interface, learning content, and personalization. Participants in the matched group had higher overall satisfaction with the tool than the unmatched group.

**Singh and Meyer.** [35] investigated whether a social annotations tool assisted students in learning course material and if the

annotations had a significant impact on student participation and performance. The study included 57 participants, 37 belonging to the experimental group and 20 in the control group. The gender composition of the participant pool was 70% male and 30% female. The tool included six categories of annotations for student use: comment, question, errata, important, confusing, and interesting. The categories ‘important’ and ‘comment’ comprised 77% of the annotations created by the participants. There were significant differences in participation and performance between the experimental group and control group. Just 35% of students from the control group interacted with the tool by spending more than four hours reading instructional material, while 73% of the experimental group interacted with the annotation tool for more than four hours. Participants in the experimental group were also more likely to have completed all of the course readings, with 57% having completed all of the course readings compared to just 36% of the control group.

### 3.3 Racial & Gender Demographics Reported

Surprisingly, only three of the 14 publications reported both racial and gender demographics of their study participants, which are described in this section.

**YeckehZaare et. al** [39] detailed the design features and learning model of a practice tool used to assist students in a large introductory Python course during Spring 2018. These researchers did report an analysis of their data stratified by ethnicity and gender amongst other metrics. The researchers reported that they collected the following demographic information from the registrar’s office: the course had 85 male students and 108 female students; 62.18% of participants reported their race as White, 24.87% as Asian, and 12.95% as other. Their findings indicated that male participants performed better with their tool compared to female participants. However, they did not find any significant results when focusing on ethnicity. They evaluated the tool for its impact on student motivation, and found that 62 of 193 participants used the tool for longer than the required 45 days. These results show that their intelligent tutoring system successfully encouraged student practice.

**PRIME.** Wiggins et al. [37] presented an experience report assessing an intelligent hinting system incorporated into a block-based coding environment. They focused on understanding students’ help-seeking behaviors during their block-based programming activities to determine patterns in their actions preceding a hint request. They conducted a study of 174 undergraduate students, recruited from an introductory engineering course on computing principles. Participants had an average age of 18.76, with 34.29% female and 65.71% male. Further, 71.26% reported their race/ethnicity as White/Caucasian, 14.94% as Asian, 4.02% as African American, 4.02% as Hispanic and 3.35% as other while the remaining 2.41% did not report their race. Researchers categorized the patterns of help-seeking behaviors into five clusters describing their hint requests, which were defined by the student’s code completeness and time elapsed for the activity at the moment they requested the hint. The clusters observed from the students’ hint-requesting patterns follow in descending order of total hints requested: *Where Do I Start?*, *Does This Look Right?*, *I Think I’m Missing Something*, *This Time For Sure...Right?*, and *I’m Out of Ideas. Help, Please*. Many hints were

requested with code completeness being 4%, suggesting students requested hints because they were completely unaware of how to start the activity. Additionally, both the *Does This Look Right?* and *This Time For Sure...Right?* clusters had a code completeness of greater than 70% and much greater elapsed time indicating that students needed affirmation or problem-specific feedback to complete the activity. These findings demonstrate how researchers can prioritize understanding the actions that indicate a learner needs assistance.

**Marwan et al.** [23] aimed to improve student engagement and commitment to a future in CS by developing an adaptive immediate feedback (AIF) system that administered encouraging messages during block-programming exercises. They conducted a pilot study consisting of 25 high school students recruited from CS summer camps. Participants had a mean age of 14. Nineteen students identified as female, six identified as male, and one indicated they preferred not to report their gender. Fourteen participants reported their race/ethnicity as White, seven as Black or African American, one as Native American, two as Asian, and one as other. With the integration of this adaptive feedback tool into block-based programming, researchers looked at how students received feedback messages during their activity.

## 4 DISCUSSION

From reviewing the last five years of SIGCSE community publications on APS, we found that demographic data was overwhelmingly underreported in discussions of study implications. In this section, we detail why these findings are concerning and identify the actions we need to take as researchers, peer reviewers, and educators to ensure our findings are properly contextualized.

The SIGCSE community strives to include and represent diverse groups of people within the computer science field, yet among many of our publications, the demographic information on study participants has not been reported. According to Upadhyaya et al. [36], when demographic data is underreported, the study’s generalizability is limited and it becomes difficult to replicate and compare studies. The key to reproducibility is the availability of information, and without important information like demographic data, it becomes much more difficult to advance as a field. Additionally, the decisions being made based on the reporting could have detrimental effects if the findings are not equitable. Even when this important demographic data is being reported, in most cases the authors do not address participant demographics during analyses or discuss what implications they may have on their findings.

The papers we collected reported on tools that aimed to provide learners with effective pedagogical support, and the writeups offered insight for instructors and researchers interested in adopting their own APS. However, the results of the systematic literature review suggest that the community should take great care when making claims about the effectiveness of their tools for various learner populations. Attempting to inform instructors and CS Ed researchers on the effective outcomes of APS without considering the demographics of participants carves a gaping path to biased system design. Worse, it could lead practitioners to make incorrect assumptions about their students’ experiences with these systems. Because racism is a socially-constructed component of society, people of

color face discrimination embedded in every facet of cultural, economical, political, and socio-environmental spheres [8], including education. We cannot neglect race when evaluating these learning technologies. Ogbonnaya-Ogburu et al. [31] presented nine stories of racial inequity in education communities, and found that areas of intense empirical study have been dictated by the curiosities and motives of a privileged population. Meanwhile, people of color have vastly different experiences in computing compared to their peers, but researchers who wish to engage in rigorous, novel, minority-focused research topics may face great challenges in conveying the importance of that work to others. The SIGCSE community must strive to not only welcome this research, but also insist that demographics are considered in the analyses of all research studies.

Regarding our first research question, this systematic literature review found that the majority of papers reporting student evaluations solely report the number of participants or the recruitment source. They report no demographics. Gender demographics were included in only five of the 14 publications. Singh and Meyer [35] reported that 70% of their participants were male and 30% were female; this distribution is reflective of the prevalent gender disparities in computing. On the other hand, Alshammari et al.'s tool evaluation study participants were comprised of 60 male students, with no representation from students of other genders. This homogeneous sample makes it difficult to generalize about the effectiveness of the tool for a gender-diverse group of learners. Findings from publications that do not include participant demographics should be interpreted with caution.

The second research question driving this literature review sought to determine if racial and gender minorities were considered within the development of APS's student models and whether researchers investigated equity in their system design. None of the collected publications featured an investigation on student demographics during system design. Participatory design combined with intentionally reporting results from historically marginalized groups is one way researchers can preemptively confront bias in their study or system design. For example, Coenraad et al. [6] demonstrated how including diverse participants in the design of a system empowered students and improved system design. This work presented a case study of 12 Black female middle school students who engaged in STEM-centered participatory design activities and provided insight into the significance of participants seeing themselves as designers rather than just users. Lastly, the majority of publications did not consider race or gender in their analyses (RQ3), with just one publication [39] stratifying their data to check whether their outcomes were equitable.

We encourage all CS Ed researchers to adhere to the following minimal guidelines, and we call upon SIGCSE and related communities to demand these practices from submitted papers unless there is a clear reason why these practices could not be followed:

- **Collect the data.** Collect demographic data for all human subjects studies, whether research-focused or with the aim of presenting experience reports. Timing matters: do not collect demographics before students complete an activity, because doing so can trigger stereotype threat where learners from historically marginalized groups are impacted in a way that skews their performance negatively [18].

- **Include all minority students.** Typical approaches of random selection of participants, or first-come-first-served scheduling, will propagate structures of marginalization. For example, if a researcher needs 20 research participants and the call for volunteers receives interest from 5 women and 30 men, **include all the women**. Then, fill in the remaining slots with men. Women's voices have been marginalized for too long, but we have an opportunity to make them heard. Follow a similar approach that includes *all* of the minority learners who volunteer.
- **Report the data.** Report on the demographics of your participants, including at minimum gender and race.

## 5 CONCLUSION AND FUTURE WORK

When designing, refining, and evaluating adaptive pedagogical systems for CS Ed, the learners whose voices are included in that process shape the ultimate product. In turn, the product may be more effective for those learners with whom it was designed and tested, and less effective with others. This systematic literature review has examined the extent to which CS Ed papers reporting on adaptive pedagogical systems have considered demographics such as gender and race/ethnicity. The results demonstrate that as a community, we must do better in designing for diverse learners and reporting our work in a way that allows the community to move in this direction. From an AI fairness perspective, the models underlying intelligent tutoring system and adaptive feedback systems need to be trained on data from diverse learners to mediate bias.

While this paper has emphasized the need for reporting basic demographics, it has long been recognized in the diversity and equity research communities that simply reporting demographics is not enough. **Reporting demographics of participants involved in system design processes and evaluations should be recognized as a necessary component of CS Ed publications, but we must not stop there.** We must analyze the extent to which our systems support diverse learners. When we have enough numbers to do so, we can do this quantitatively. When we do not have the numbers to support quantitative analysis, we can gain insights into the experiences of diverse learners by conducting qualitative analyses including interviews, focus groups, think-alouds, and post-hoc process studies of system logs and videos, all with proper consent and compensation for our participants. Is a system supporting significantly greater learning gains for men than women? Are Black students choosing to use a tool less than White students? To move toward equitably supporting all students, we must investigate these questions rather than turn away from them. We as a CS Ed community can either continue to perpetuate marginalization by our inaction, or we can hold ourselves to a higher standard. Let us go for the higher standard.

## ACKNOWLEDGMENTS

Thanks to the LearnDialogue Group and to the anonymous reviewers for providing suggestions for improvement. This material is based upon work supported by the National Science Foundation under grant CNS-1622438. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

- [1] Mohammad Alshammari, Rachid Anane, and Robert J. Hendley. 2015. The Impact of Learning Style Adaptivity in Teaching Computer Security. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITICSE '15)*. 135–140.
- [2] Theresa Beaubouef and John Mason. 2005. Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations. *SIGCSE Bull.* 37, 2 (June 2005), 103–106.
- [3] Kacey Beddoes and Maura Borrego. 2011. Feminist theory in three engineering education journals: 1995–2008. , 281–303 pages.
- [4] B. Blaser, R. E. Ladner, and S. Burgstahler. 2018. Including Disability in Diversity. In *2018 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE Computer Society, 1–4.
- [5] Peter Brusilovsky and Christoph Peylo. 2003. Adaptive and Intelligent Web-Based Educational Systems. *Int. J. Artif. Intell. Ed.* 13, 2–4 (April 2003), 159–172.
- [6] Merijke Coenraad, Jen Palmer, Diana Franklin, and David Weintrop. 2019. Enacting Identities: Participatory Design as a Context for Youth to Reflect, Project, and Apply Their Emerging Identities. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children (Boise, ID, USA) (IDC '19)*. Association for Computing Machinery, New York, NY, USA, 185–196. <https://doi.org/10.1145/3311927.3323148>
- [7] Computing Research Association. 2017. Generation CS: Computer Science Undergraduate Enrollments Surge Since 2006. <https://cra.org/data/Generation-CS/> Accessed on 01.28.2021.
- [8] Richard Delgado, Jean Stefancic, and ANGELA HARRIS. 2012. *Critical Race Theory: An Introduction, Second Edition*. NYU Press. <http://www.jstor.org/stable/j.ctt9qg9h2>
- [9] J. Denner and S. Bean. 2018. The Role of Relationships in Engaging Latino/a High School Students in Computer Science. In *2018 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE Computer Society, 1–5.
- [10] W. M. DuBow and L. James-Hawkins. 2015. What influences female interest and persistence in computing? Preliminary findings from a multi-year study. In *2015 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE Computer Society, 1–5.
- [11] Gaetano Geck, Artur Ljuljin, Sebastian Peter, Jonas Schmidt, Fabian Vehlken, and Thomas Zeume. 2018. Introduction to Iltis: An Interactive, Web-Based System for Teaching Logic. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITICSE 2018)*. 141–146.
- [12] David Hankerson, Andrea R. Marshall, Jennifer Booker, Houada El Mimouni, Imani Walker, and Jennifer A. Rode. 2016. Does Technology Have Race?. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. 473–486.
- [13] Rachel Harsley, Davide Fossati, Barbara Di Eugenio, and Nick Green. 2017. Interactions of Individual and Pair Programmers with an Intelligent Tutoring System for Computer Science. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. 285–290.
- [14] Rachel Harsley, Nick Green, Mehrdad Alizadeh, Sabita Acharya, Davide Fossati, Barbara Di Eugenio, and Omar AlZoubi. 2016. Incorporating Analogies and Worked Out Examples as Pedagogical Strategies in a Computer Science Tutoring System. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. 675–680.
- [15] D. James and L. Hampton. 2020. Using Black Music as a bridge to understanding introductory programming concepts. In *2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE Computer Society, 1–4.
- [16] Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. 2016. Towards a Systematic Review of Automated Feedback Generation for Programming Exercises. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITICSE '16)*. Association for Computing Machinery, 41–46.
- [17] Hassan Khosravi, Shazia Sadiq, and Dragan Gasevic. 2020. Development and Adoption of an Adaptive Learning System: Reflections and Lessons Learned. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. 58–64.
- [18] Amruth N. Kumar. 2012. A Study of Stereotype Threat in Computer Science. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education (ITICSE '12)*. Association for Computing Machinery, New York, NY, USA, 273–278. <https://doi.org/10.1145/2325296.2325361>
- [19] Amruth N. Kumar. 2018. Epplets: A Tool for Solving Parsons Puzzles. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. 527–532.
- [20] J. A. Maestro-Prieto and María Aránzazu Simón Hurtado. 2015. Learner-Adaptive Pedagogical Model in SIAL, an Open-Ended Intelligent Tutoring System for First Order Logic. In *AIED*.
- [21] Jose A. Maestro-Prieto and Arancha Simon-Hurtado. 2018. The Pedagogical Model of SIAL: An Adaptive and Open-Ended Intelligent Tutoring System for First Order Logic. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITICSE 2018)*. 21–26.
- [22] Ioannis Magnisalis, Stavros Demetriadis, and Anastasios Karakostas. 2011. Adaptive and Intelligent Systems for Collaborative Learning Support: A Review of the Field. *IEEE Transactions on Learning Technologies* 4, 1 (2011), 5–20.
- [23] Samiha Marwan, Ge Gao, Susan Fisk, Thomas W. Price, and Tiffany Barnes. 2020. Adaptive Immediate Feedback Can Improve Novice Programming Engagement and Intention to Persist in Computer Science. In *Proceedings of the 2020 ACM Conference on International Computing Education Research (ICER '20)*. 194–203.
- [24] Mark McCartin-Lim, Beverly Woolf, and Andrew McGregor. 2018. Connect the Dots to Prove It: A Novel Way to Learn Proof Construction. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. 533–538.
- [25] M. Mejias, K. Jean-Pierre, L. Burge, and G. Washington. 2018. Culturally Relevant CS Pedagogy - Theory & Practice. In *2018 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE Computer Society, 1–5.
- [26] Cade Metz. 2021. Who Is Making Sure the A.I. Machines Aren't Racist? *The New York Times* (Mar 2021). <https://www.nytimes.com/2021/03/15/technology/artificial-intelligence-google-bias.html>
- [27] Barbara Moskal, Deborah Lurie, and Stephen Cooper. 2004. Evaluating the Effectiveness of a New Instructional Approach. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE)*. 1053–1059.
- [28] Barbara Moskal, Deborah Lurie, and Stephen Cooper. 2017. iSnap: Towards Intelligent Tutoring in Novice Programming Environments. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE)*. 1053–1059.
- [29] National Academies of Sciences, Engineering, and Medicine. 2018. "Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments". "The National Academies Press". <https://doi.org/10.17226/24926>
- [30] Lunga Ntila. [n.d.]. How The Racism baked Into Technology Hurts Teens. Available at <https://www.theatlantic.com/family/archive/2020/10/algorithmic-bias-especially-dangerous-teens/616793/> (2020/10/24).
- [31] Ihudiya Finda Ogbonnaya-Ogburu, Angela D.R. Smith, Alexandra To, and Kentaro Toyama. 2020. Critical Race Theory for HCI. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. 1–16.
- [32] Thomas W. Price, Yihuan Dong, and Dragan Lipovac. 2017. iSnap: Towards Intelligent Tutoring in Novice Programming Environments. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. 483–488.
- [33] Jaime Raigoza. 2017. A study of students' progress through introductory computer science programming courses. In *2017 IEEE Frontiers in Education Conference (FIE)*. 1–7. <https://doi.org/10.1109/FIE.2017.8190559>
- [34] J. J. Ryooy and K. Tsui. 2020. What Makes a Computer Science Person? Minoritized Students' Sense of Identity in AP CSP Classrooms. In *2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE Computer Society, 1–8.
- [35] Shaveen Singh and Bernd Meyer. 2019. Using Social Annotations to Augment the Learning Space and Learner Experience. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (ITICSE '19)*. 527–533.
- [36] Bishakha Upadhyaya, Monica M. McGill, and Adrienne Decker. 2020. A Longitudinal Analysis of K-12 Computing Education Research in the United States: Implications and Recommendations for Change. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. 605–611.
- [37] Joseph B. Wiggins, Fahmid M. Fahid, Andrew Emerson, Madeline Hinckle, Andy Smith, Kristy Elizabeth Boyer, Bradford Mott, Eric Wiebe, and James Lester. 2021. Exploring Novice Programmers' Hint Requests in an Intelligent Block-Based Coding Environment. 52–58.
- [38] Franceska Xhakaj and Chun W. Liew. 2015. A New Approach To Teaching Red Black Tree. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITICSE '15)*. 278–283.
- [39] Iman YeckehZaare, Paul Resnick, and Barbara Ericson. 2019. A Spaced, Interleaved Retrieval Practice Tool That is Motivating and Effective. In *Proceedings of the 2019 ACM Conference on International Computing Education Research (ICER '19)*. 71–79.
- [40] R. Zarch, S. Dunton, J. Childs, and A. Leftwich. 2020. Through the Looking Glass: Computer Science Education and the Unintended Consequences of Broadening Participation Policy Efforts. In *2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE Computer Society, 1–4.
- [41] Stuart Zweben and Betsy Bizot. 2020. Total Undergraduate CS Enrollment Rises Again but with Fewer New Majors; Doctoral Degree Production Recovers From Last Year's Dip. In *2019 Taulbee Survey, Computing Research Association*. CRA.