



A Practical Strategy for Training Graduate CS Teaching Assistants to Provide Effective Feedback

Alina Zaman
University of Memphis
Memphis, TN, USA
azaman@memphis.edu

Vinhthuy Phan
University of Memphis
Memphis, TN, USA
vphan@memphis.edu

Amy Cook
University of Memphis
Memphis, TN, USA
ascook@memphis.edu

Alistair Windsor
University of Memphis
Memphis, TN, USA
awindsor@memphis.edu

ABSTRACT

Computer science (CS) relies heavily on teaching assistants (TAs) who are often untrained in CS pedagogy. Existing research on CS TA training typically studies American undergraduate TAs at high-resource universities, ignoring the many universities that use graduate TAs, who are often international students, and that don't have the resources to implement the training strategies discussed in the literature. We describe our approach to implement graduate TA training in a high-diversity, low-resource context. We present a needs assessment, design, pilot test, and deployment of our training course, and discuss implications for other similar departments hoping to train their TAs.

CCS CONCEPTS

• **Applied computing** → **Interactive learning environments; Computer-assisted instruction.**

KEYWORDS

teaching assistant training, feedback, CS1

ACM Reference Format:

Alina Zaman, Amy Cook, Vinhthuy Phan, and Alistair Windsor. 2023. A Practical Strategy for Training Graduate CS Teaching Assistants to Provide Effective Feedback. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2023)*, July 8–12, 2023, Turku, Finland. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3587102.3588776>

1 INTRODUCTION

Computer science (CS) relies heavily on teaching assistants (TAs) to provide feedback on students' code. Duties include grading, helping with homework in office hours, providing feedback during lab sections, and assisting with in-class coding exercises. Low-quality

feedback can negatively impact student success and their willingness to continue in computing.

One way to improve TA feedback quality could be TA training. Many TAs are untrained and have little to no prior teaching experience. Current practices for TA training are unstandardized and often focus on classroom management or course content, rather than pedagogy [5].

Computer science has unique demographics that exacerbate the need for TA training. In 2016-2017, 64% of US graduate degrees in Computer Science were awarded to non-resident aliens [26]. At our institution, 88% of CS graduate students are non-resident aliens, while only 1% of undergraduate students are international students. With many CS TAs coming from educational and cultural backgrounds that are different from students they serve, it is crucial to equip TAs with culturally responsive teaching methods. Addressing issues like implicit bias, growth mindset, and stereotype threat can further enhance the quality of feedback they provide to students.

Providing TA training can be challenging. Professors often lack the time, incentives, or motivation to develop, implement, and evaluate training materials [7, 8, 19]. While a university's teaching and learning center might provide generic resources, such as English language assistance for international TAs, such resources are typically not CS-specific. Further, many departments, including ours, lack non-research resources such as a center, or even a designated faculty member devoted to TA training. Most often, each individual faculty member is responsible for providing training (or not) to TAs assigned to their courses.

In this paper, we investigated how to design TA training for CS that could 1) teach TAs effective strategies for providing feedback on code, 2) address cultural differences between international graduate TAs and domestic undergraduate students, and 3) be delivered in a low-resource context (e.g. no teaching and learning center, or other departmental support). We present our design for a TA Training course developed as Canvas modules. We report the results of a Spring pilot test, changes to our materials, and a Fall deployment of the training. We conclude with implications for other low-resource, high-diversity computing departments hoping to train their TAs.

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 the author(s) 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.

ITiCSE 2023, July 8–12, 2023, Turku, Finland

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0138-2/23/07...\$15.00
<https://doi.org/10.1145/3587102.3588776>

2 RELATED WORK

2.1 Benefits and Challenges of TAs in CS

2.1.1 Why Does CS Rely on TAs? The number of students who want to study CS is continually increasing, but few want to teach Computer Science as a career. CS departments often struggle to hire new teaching-focused faculty, leading to a high student to teacher ratio and large class sizes, particularly for introductory courses. CS class sizes are growing significantly faster than faculty hiring [4], so the reliance on TAs is expected to continue.

2.1.2 Who are CS TAs? Three different categories of students typically work as CS TAs: graduate TAs (GTA), undergraduate TAs (UTA), and international TAs (ITA). Most often, TAs (of any category) have no teaching experience before working as a TA [41]. Whether a department uses GTAs or UTAs usually depends on the university culture [33], but they often perform similar functions [41] in undergraduate courses.

These TAs have different goals. For example, a GTA might not want to be a computer science teacher; they are a TA to get funding for their degree. TA work is often required for graduate students, and their TA duties might feel like a burden that competes with research. In contrast, an undergraduate is not required to work as a TA. A UTA likely volunteered for the position because of an interest in teaching. Researchers in other fields have found that UTAs were as effective as GTAs [17].

An ITA is frequently a non-native English speaker at a university where English is the primary language. ITAs in computing are usually graduate students. They may face challenges beyond language barriers, such as different cultural norms for student-teacher interactions or unfamiliarity with the role of a teaching assistant.

2.1.3 What Do CS TAs Do? TAs play a key role in supporting both instructors and students. TAs generally participate in grading, but might also lead lab sections [3, 12, 21, 22], support group projects [6, 14, 25], develop course activities [3, 9, 50], give feedback on student's code and assist with in-class coding exercises [31].

TAs have a significant positive impact on student learning [24, 36]. TAs impact classroom climate, course grades, and student's knowledge of career options in the discipline [35]. Low-quality feedback can confuse and demotivate students and lead to increased failure and dropout rates [18, 27]. High-quality feedback helps students learn to correct misconceptions, reflect on their work, and self-regulate their learning [2, 18].

2.1.4 Where Do CS TAs Struggle? TAs typically do not receive pedagogical training [23] and are often unsupervised [34]. Researchers have identified common issues when TAs teach. TAs mistakenly think that lecturing and teaching are equivalent [10, 13]. TAs do not have the desired skill set to effectively communicate with students [23, 25, 34]. There are significant differences in amount, length, and quality between instructor and TA feedback on student code [18]. When helping students in office hours, TAs feel 'success' is fixing the students' code, not helping the student learn to fix their own code [30]. TAs tend to guide students through the TA's own solution of the problem, providing explicit direction, rather than building an understanding of the student's code [30]. Given their

lack of pedagogical training, it is not surprising that TAs struggle with effective feedback provision.

2.1.5 Where Do ITAs Struggle? Miscommunication and linguistic differences can prevent ITAs from performing their best in pedagogical duties [28]. Although undergraduate students sometimes report trouble understanding ITAs' accent, English pronunciation is often not the primary communication barrier [32, 37, 45]. Often, a student can understand their TA, but feel they cannot relate to their TA. These communication breakdowns are often linguistic in nature [28]. For example, the difference between "WHY DID YOU TRY THAT!" and "Why did you try that?" is prominence (word stress) and rising intonation (to signal a question).

2.2 TA Training Methods in Computer Science

2.2.1 Who Offers Training, and How. When training is offered, there are multiple models for how TA training is delivered. A university's HR department might offer online modules for general topics like FERPA training. A university's teaching and learning center might offer general TA training focusing on widely applicable pedagogical strategies. These might be delivered as in person lectures, online modules, or even micro-teaching workshops with personalized feedback. A course instructor might provide content-focused training for their specific course's TAs as a one-time event. A CS department might offer a semester-long course for department-wide training for their TAs that focuses on CS-specific pedagogy. In this case, experienced TAs [21, 43] and, to a lesser extent, faculty members [50] can be involved in training new TAs. Some programs involve TA mentorship from peers and instructors [49]. Often, TAs are trained through formal training courses [12, 39, 43, 48] or workshops [44, 44]. Such training can include self reflection exercises [39] and student-TA role playing [48].

2.2.2 What is Taught during Training. Training content might include learning and teaching styles [20, 50], communication skills [50], grading [43, 44], professionalism [3, 48, 50], proper teaching techniques [3, 11, 50], and how to work with difficult students [11, 40]. ITA training typically focused more on English language pronunciation [1, 29, 38] than understanding cultural differences.

2.3 Problems with Existing TA Training Practices

Most studies on undergraduate TA training were conducted at high-resource universities and have focused on American students [25, 30, 33, 42, 46, 47]. This line of work has a few significant gaps.

2.3.1 Lack of Training for Graduate Students. Existing research on CS TA training typically studies UTAs, rather than GTAs with many competing demands on their time. Such training doesn't account for the goal misalignment of GTAs who don't see their TA duties as relevant to their career trajectory.

2.3.2 Lack of Training for International TAs. Existing research on CS TA training typically studies American TAs rather than ITAs. When training was provided to ITAs, it typically focused on English language pronunciation only. Such training didn't include strategies for navigating cultural differences between TAs and undergraduate students.

2.3.3 Lack of Training Strategies for Low-Resource Environments. Prior research was often conducted at schools with resources to design and implement a department-level CS-specific TA training programs, overlooking the majority of schools lacking such resources. Most CS departments do not have a teaching and learning center, a designated faculty member for TA training, or CS-specific teaching materials. In many cases, no TA training is provided [23], making it hard to adopt the methods described in the literature.

3 NEEDS ASSESSMENT: DECIDING WHAT TO INCLUDE IN TRAINING

We discovered three gaps in prior work: 1) a lack of TA training for computer science *graduate* students, 2) a lack of training that addresses cultural competency for international TAs teaching diverse undergraduates, and 3) a lack of research on training strategies that can be delivered in low-resource contexts. We surveyed our faculty and CS1 students to further inform TA training.

3.1 Assessing Faculty Needs

The faculty survey was conducted through Qualtrics. 14 out of 22 CS faculty responded (labeled F1-14); all work with international student TAs. Only three faculty members had previously worked with their assigned TA. We asked multiple choice and open-ended questions about typical expectations for TA duties and the areas their TAs might need more support. The instructor survey asked questions such as: What roles and responsibilities do TAs have in your course? What problems or challenges do your TAs have when trying to complete these duties? What support or resources do you provide to TAs to help them complete their duties? How often do you meet with your TA outside of class? Please describe what happens in your TA meetings. What additional duties do you wish a future TA could perform for your course that you currently don't ask for?

3.1.1 Duties. All faculty reported the primary TA duty as grading homework assignments. 8 out of 14 faculty asked their TAs to grade tests and exams. Helping students in office hours was also a common TA duty (8 out of 14). Only three had TA-led lab sessions. Only one indicated their TA led an occasional lecture.

Some faculty indicated they wished their TAs could do additional duties that other TAs do, like answering students questions during class (F 1,3,5) and leading recitation or tutoring sessions (F 6,7). Two wanted TAs to do more pedagogically intensive duties like leading a lecture (F13) or writing lab assignments (F12), but both expressed significant reservations: "I'd be hesitant to put that responsibility on a non-native English speaker" (F12), and "I am not confident that a TA could independently do this" (F13).

One faculty noted that our TAs are so overwhelmed with just grading that it's unreasonable to expect them to do more: "It would be crazy to ask for more duties. They can barely keep up" (F11). Our department expectations for how TAs spend their contracted hours are almost solely focused on grading. This includes usually not expecting TAs to attend class, and not regularly meeting with TAs. Only 3 of the instructors met weekly with their TAs. 8 reported monthly meetings, and 3 reported meeting only once or twice a semester. There was no relationship between frequency of TA meetings and intensity of TA duties.

3.1.2 Challenges. When asked about challenges TAs face, three themes emerged. Unsurprisingly, the first challenge related to grading (F 1,2,3,6,12). TAs make "errors in grading" (F6), they "sometimes give full credit for wrong answers or take off points for correct answers" (F2), and they struggle to give "effective feedback when grading" (F12). One faculty noted that "They don't know how to use rubrics to grade fairly, or other best practices for grading" (F1). Second, faculty lamented that TAs often do not have sufficient background knowledge in the subject matter and often have not taken the course (F 1,3,8,10,14). They know it is "hard to find qualified TAs on specific subjects" (F14), but struggle when TAs "do not understand the material well to grade assignments. This is especially true if they don't attend lectures" (F3). Third, faculty noted challenges related to larger course sizes (F 5,6,11,13). Our department had a 400% increase in MS student enrollment this semester, which necessitated an increase in our student to TA hours ratio, among a host of other challenges. Faculty commented that TAs are "not responding to students in time" (F6), TAs "find it hard to give individual attention to students" (F13), TAs have "too many students to grade" (F11), and that the "rate of 30 [students] to 10 hours [of TA work per week] is dismal" (F11). These challenges are interconnected. Without sufficient background knowledge, class attendance, or strategies to grade efficiently and effectively, how could TAs possibly grade well?

3.2 Assessing Student Needs

We surveyed students in CS1, who have the most TA interaction, about their experiences with TAs in our department. We asked the instructor to share the survey with all CS1 students. We received 40 responses, referred to as S1-40. The survey asked students to identify specific TA behaviors that they found most and least beneficial to their learning. The student survey asked Likert-scale and open-ended questions such as: How useful were the TAs in providing feedback on in-class exercises? outside of class? How clearly did your TAs answer questions? What did your TAs do that most benefited you? What did your TAs do that you found least beneficial? We did not collect demographic data on this survey to protect student anonymity and prevent possible re-identification of students from underrepresented groups.

3.2.1 Beneficial Behaviors. 26 students identified TA behaviors they considered beneficial to their learning. Students appreciated that TAs would find the bugs in their code (S 14,33,35,36,39,40), saying TAs would "catch a lot of my mistakes" (S33), "point out what is wrong with my code" (S35), and "show me that my code was correct" (S14). Students appreciated that TAs would "help" by fixing their code for them (S 2,9,20,21,27,29,30). Some referred to administrative behavior, such as flexibility in scheduling and fast replies to emails. Only a few students described TA behavior that we would consider effective teaching. One student recalled when their TA "explained some concepts that I had to use in the lab when I didn't understand it from the class or zybooks" (S28), and another student said the TA "helped me on one of the lab questions by drawing an illustration of how one loop works" (S34). Other researchers have also found that when students ask TAs for help, they expect their TAs to find and fix errors in their homework [30], even though that behavior does not promote student learning.

3.2.2 Adverse Behaviors. 16 students reported that their TAs did not negatively impact their learning; 10 left this question blank. One student never communicated with their TA, and another did not know who their TA was, even after attending lab sessions led by the TA. The remaining 12 students described behaviors they did not consider beneficial. Common complaints include confusion about code review meetings with CS1 TAs (S 3,13,27,31,37), such as "this is my first experience with code reviewing and I'm not sure how it's supposed to go. Any amount of feedback from my TA would make it a lot less uncomfortable" (S37). Three students (S 2,7,33) had difficulty understanding their TAs' accent. Four expressed frustration that their TA would not fix their code for them (S 3,16,35,36). Students felt the TA was "a bit vague on what needs to be fixed" (S35) and that the TA "only said what I did wrong" (S16) instead of also providing a solution. Another student described how "when I asked questions in lab [...] was told to think about the problem. This was very frustrating" (S3).

3.3 Focus Areas for CS TA Training

Based on existing ITA training research [28, 32, 37, 45] and our needs assessment, our TA training focuses on two areas: cultural competency tips for international TAs, and providing effective feedback on code.

To accomplish this instruction in our low-resource context, we designed Canvas modules that TAs can complete online. TAs can complete 1 module per week for 4 weeks, each taking 1 hour or less. The total time of training is quite small, but this was the maximum amount of time we could add to our graduate TAs workload without violating their existing contract. The next section describes our TA training modules in detail.

4 DESCRIPTION OF TRAINING MODULES

We developed Canvas modules (videos, quizzes, and reflection posts) to introduce topics related to effective feedback provision.

4.1 Week 1: Context

4.1.1 Scaffolding Feedback. This 5 minute video served as an introduction to the overall training course. It emphasized how important TA feedback is to student success. We introduced the concept of scaffolding feedback, or giving feedback in bite-sized chunks, to help a student move forward on their own rather than giving away the answer and preventing a student from learning.

4.1.2 ITA Tips. First, we define the role of a TA in our department. We emphasize that a TA is a guide and ally to the students, not a disciplinarian, and that we define success for a TA as helping students understand course material. We also discuss contextual differences between the US education system and universities in other countries.

Second, we provide 4 tips to help TAs seem more approachable to students, and therefore more likely to ask them for help. The first two tips are focused on WHAT to say. We suggest TAs introduce themselves, call out differences between their native language and English, encourage students to ask for clarification if they have difficulty understanding them because of an accent. Next, we provide strategies for how to buffer negative feedback. The last two

tips are about HOW to say things, providing linguistic tips for non-native English speakers. Experts in this field indicated that rising intonation and word stress are the two most impactful tips for ITAs.

The reflection post asks ITAs to share differences between classrooms in their home country and at our university, and differences between English and their native language.

4.2 Week 2: Mindset

Week 2 focused on growth mindset and stereotype threat. There is growing evidence that the mindset of STEM instructional staff affect student performance, and that this effect is particularly strong for students from under-represented groups [15, 16]

The two modules are primarily informational, focusing on research papers. The growth mindset module offers vignettes from current teaching assistants reflecting on their own mindsets. They reflect on the effects this has had on their academic careers and offer explicit advice for how growth mindset research indicates that feedback should be delivered. The module on stereotype threat includes advice on how to select exemplars and frame questions in a way that best minimizes stereotype threat and offers information about the current makeup of both undergraduate and graduate computer science programs (our graduate program has near gender parity). The reflection post asks TAs about their personal experiences with stereotypes in computing.

4.3 Week 3: Feedback Structure

We taught three recommendations for feedback structure: use full sentences, ask questions, and how to use code snippets wisely. We called out how feedback that provides code a student can copy takes away a learning opportunity from the student. Instead, we encouraged TAs to use code snippets to provide examples or test cases that help students identify their own misconceptions.

We devoted an entire module to the concept of providing feedback that is actionable but not prescriptive. In other words, how to give a useful hint without giving away the answer. We provided multiple examples of feedback that was not actionable (e.g. "good job" or "this line is wrong"), feedback that was actionable and prescriptive that gives away the answer or tells students exactly how to change their code, and feedback that was able to provide an actionable next step without prescribing an answer.

4.4 Week 4: Feedback Strategies

First, we teach TAs how to identify the gap between what a program is supposed to do and what it actually does. In the first video, we walk TAs through specific examples that show coding mistakes together with hypothetical feedback that can be improved upon. We ask TAs to articulate the gap between what a program does and what it needs to accomplish. We also discuss different ways to provide feedback to an example program that helps students understand the difference between "printing" and "returning".

Second, we discuss the benefits of using well-chosen examples in feedback. At times, providing students with tangible examples can be more effective in clarifying complex concepts and identifying the gaps for students. We emphasize that examples should not give away answers and that such examples might be inadequate on their own.

5 PILOT TEST

5.1 Participants

At the end of the Spring semester, we emailed all current graduate students who had worked as a department TA and invited them to be a pilot test participant. Thirteen students agreed to participate (referred to as PT 1-13). Each student earned \$80 (\$20 per 1-hour module) upon completion of the pilot test.

5.2 Procedure

Pilot testers could complete the Canvas modules at their own pace during the month of May. We added an evaluation survey at the end of each module so pilot testers could give feedback. The evaluation survey asked TAs if each learning objective was new to them, and if they felt it was important. TAs reported their completion time, and if they had any suggestions for improving the module.

5.3 Pilot Test Results

5.3.1 Familiarity. TAs were mostly unfamiliar with the content of our training (Figure 1). Least familiar topics were growth mindset, stereotype threat, and being actionable without being prescriptive.

Pilot Test: TA Familiarity with Training Content

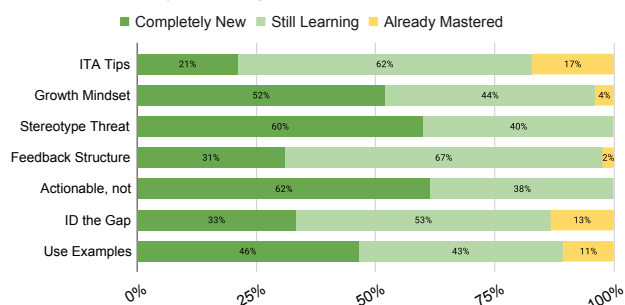


Figure 1: The training covered topics that TAs were unfamiliar with.

5.3.2 Importance. TAs felt the topics we covered were important (see Figure 2). Note that Growth Mindset and Stereotype Threat had lower ratings of importance than the other modules. The learning goals in this module included knowing the names of authors of research in this field, which TAs did not think was valuable. TAs rated the other learning goals in these modules similarly to the other training content.

Pilot Test: TA Perspective of Importance of Training Content

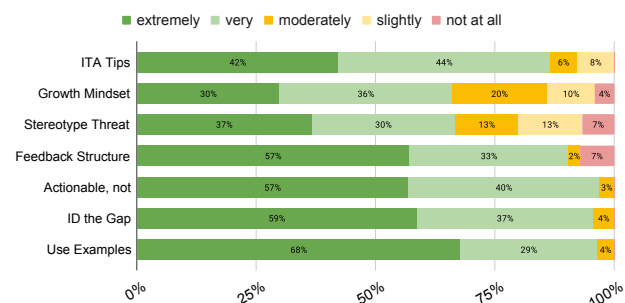


Figure 2: TAs felt the training covered important topics.

5.3.3 Completion Time. We were hoping the completion time for each module would be around 30 minutes per topic, as students would typically do two topics per week for one hour of training. Almost all the modules met this goal (see Figure 3), but the Use Examples module was too long. TAs indicated that they felt this quiz was much more difficult.

Completion Time (in Minutes) for Each Module

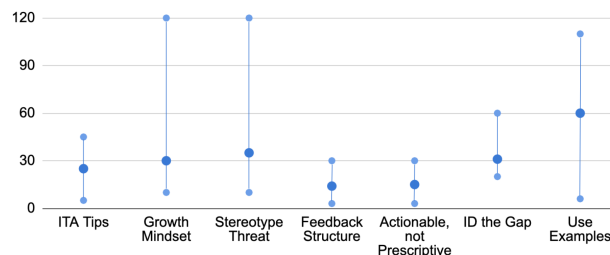


Figure 3: TAs reported most modules took about half an hour to complete.

5.3.4 Changes to Modules. To align our material with what TAs value, we removed quiz questions on research paper authorship. We also adjusted the Use Examples module and quiz to be shorter. We updated the code examples in the quiz questions to use Python instead of Java, in line with the upcoming changes for our CS1 course. The remainder of our changes were bug fixes. The researchers were unfamiliar with Canvas and had a few setup issues. After making necessary adjustments, we felt the training was ready for an authentic deployment. We successfully convinced the department chair to require training for all CS1 TAs in the fall semester.

6 DEPLOYMENT

6.1 Method

6.1.1 Participants. All 15 CS1 computer science teaching assistants participated in the feedback training course, although 2 dropped out before completing all the modules. 6 were graduate TAs, and 9 were undergraduate graders. Graduate TAs lead lab sections, help students one-on-one during lab, and grade lab assignments. They also conduct code reviews meetings with each student on each lab assignment. Undergraduate graders provide written feedback on in-class exercises. Undergraduate graders also occasionally help the instructor with grading quizzes and exams.

6.1.2 Procedure. We expected TAs to complete one module per week, for 4 weeks of training at the start of the semester. However, training took 6 weeks in practice. We gave one extra week so that TAs would not be required to do a module over Fall Break. We gave a second extra week because TAs frequently asked via email for a second chance on the quizzes. Second attempts were not mandatory, but many students wanted to improve their score. The TAs' performance drastically improved in the second attempt.

6.2 Results and Discussion

6.2.1 Training Completion. There are a total of 7 videos throughout the 4 modules. Canvas tracks how much of the video each student

watched. We report the students' average quiz scores from their first and second attempts.

Topic	Average % of video seen	Average score in quiz (1st attempt)	# of students who took a 2nd attempt	Average score in quiz (2nd attempt)
Tips for International TAs	84%	97%	2	100%
Growth Mindset	93%	77%	6	97%
Stereotype Threat	91%	91% <i>1 non-participant</i>	2	84%
Feedback Structure	87% <i>1 non-viewer</i>	73%	4	98%
Actionable not Prescriptive	100% <i>2 non-viewers</i>	80% <i>2 non-participants</i>	5	100%
Identifying the Gap	97%	93%	2	100%
Using Examples in Feedback	82%	71%	2	97%

Figure 4: TAs' training completion.

We analyzed TAs' reflection posts using affinity diagrams to learn their perspective on the course content. There were 7 discussion questions, and 14 out of 15 participants (referred to as P 1-14) voiced their opinions. Notable themes are discussed below.

6.2.2 TAs connected training content to their past experiences. TAs compared different expectations for appropriate classroom behavior in their home country and in the United States (P 4,5,12,13). "[...] Here there is lot of freedom for students to express their thoughts and focusing on self study", said P3.

The linguistic tips were beneficial to both US and international TAs (P 1,3,6,7,9,11,12). "It's important to be sure you're using a tone that comes off as helpful or asking a question rather than accusatory" (P11). Some found the tips quite surprising (P 1,6,11). Some TAs "never heard someone say that you should make your voice go slightly higher when asking a question and slightly lower when making a statement (P1)" or that in their native language, they "don't have too much prominence (P6)".

TAs shared their experiences encountering gender stereotypes about computer scientists (P 4,7,13). TAs also encountered stereotypes about computer scientists as smart and antisocial (P 3,5,6,9,12,13). Most TAs believe the media propagates stereotypes (P 1,2,3,4,5,6,8,12).

6.2.3 The Training Contradicted TAs Assumptions about Good Feedback. TAs shared which things they learned in training were surprising to them. Some of their surprise came from a misunderstanding about what feedback should do, and a lack of awareness of feedback strategies. Many TAs expressed surprise that feedback shouldn't give away the answer (P 1,2,8,10,12). P2 stated, "The thing that surprised me was not to tell the students directly how to fix their code and instead motivate them to think for it and give them a push in the right direction." P 3, 4, 5, 6, 7, 8, 10, 11 and 12 also realized that feedback should help students learn to fix code on their own. After completing the training module, the TAs understood that not all feedback is helpful (P 3,4,5,7,8), saying "What surprised me is that some feedback may not be beneficial to students" (P6).

These feedback strategies were unfamiliar, but TAs expressed the importance of providing actionable feedback (P 6,9,10,11) and identifying the gap (P 1,2,11,13). "I also learned that it is important to give feedback that is specific, objective, actionable and tell them what they can do differently to improve" (P3). Through the completion of this training, TAs have also realized TAs are a helper, not a person in charge (P 2,3,4,8,10).

6.2.4 TAs realized writing good feedback is harder than expected. TAs lamented that writing good feedback is harder than they originally thought (P 1,2,3,5,6,8,12). TAs find difficulty in not giving away the answer (P 1,3,4,5,6,8,10,11,13). "[...] to choose examples that will just help the student understand his mistake and not necessarily give away the answer, will be the most difficult to implement. But, I am up for the challenge"(P5) Some other TAs think "Bridging the gap"(P11) and "inspiring students to find out answers by themselves"(P6) are the most challenging parts of giving feedback "because the student may have a different way of thinking through what is happening in their code"(P11) .

TAs picked up new tips for providing feedback (P 2,4,7,8,14). P2 remarked, "Through this module I learned that different types of people should be given different feedbacks depending on how close their answers are and what concepts are seemed to be clear to them and what are not."

6.2.5 TAs were committed to implementing the strategies. TAs felt the feedback strategies would be effective (P 1,5,6,7,10). "The most important thing that I learned from this week modules is that explaining the mistakes with examples, which is better and easy way for the students to understand their mistake and I am excited to implement this strategy in further.", said P9. TAs were committed to give the feedback in such a way that helps students think and solve on their own (P 2,3,4,5,7,10,11). P1 mentioned, "I'm most excited to implement actionable feedback and feedback that helps students arrive at the correct answer on their own." The TAs think that they could accomplish giving effective feedback (P 5,6,8,12,13). P6 said, "But with some practice, it is not hard to achieve it".

TAs felt the training could help them provide more support for international students (P 1,2,7,8,10,14). "I could use these tips to support international students by making sure they understand what I am trying to tell them and make sure to let them know they are more than welcome to ask questions. ", remarked P14.

7 CONCLUSION AND FUTURE WORK

We started small, with only 4 hours of online training for CS1 TAs, as this was all our department was willing to support. However, they are now more receptive to TA training. Next semester, all department TAs are required to complete training, and we are adding an in-person component. TA Orientation Week will include 10 hours of training (4 online and 6 in person) during the first week of class, when TAs are under contract but have not yet received their class assignment. We hope this arrangement will allow TAs to participate in training activities without adding to their workload. We will survey TAs to get their input on the effectiveness of future training. In our future work, we are exploring adapting and expanding this training to CS departments in other universities.

ACKNOWLEDGMENTS

This work is funded by NSF Award 2044279.

REFERENCES

- [1] Comfort Tosin Adebayo and Mike Allen. 2020. The experiences of international teaching assistants in the US classroom: A qualitative study. *Journal of International Students* 10, 1 (2020), 69–83.
- [2] Joe Michael Allen, Frank Vahid, Alex Edgcomb, Kelly Downey, and Kris Miller. 2019. An analysis of using many small programs in cs1. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 585–591.

- [3] Christine Alvarado, Mia Minnes, and Leo Porter. 2017. Micro-classes: A structure for improving student experience in large classes. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. 21–26.
- [4] Computing Research Association et al. 2018. Taulbee survey. *Computing Research News* 14 (2018).
- [5] Ann E Austin. 2002. Preparing the next generation of faculty: Graduate school as socialization to the academic career. *The journal of higher education* 73, 1 (2002), 94–122.
- [6] Xiaoying Bai, Mingjie Li, Dan Pei, Shanshan Li, and Deming Ye. 2018. Continuous delivery of personalized assessment and feedback in agile software engineering projects. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*. 58–67.
- [7] Paul J Baker and Mary Zey-Ferrell. 1984. Local and cosmopolitan orientations of faculty: Implications for teaching. *Teaching Sociology* (1984), 82–106.
- [8] Judith Berman and Kelley M Skeff. 1988. Developing the motivation for improving university teaching. *Innovative Higher Education* 12, 2 (1988), 114–125.
- [9] Maureen Biggers, Tuba Yilmaz, and Monica Sweat. 2009. Using collaborative, modified peer led team learning to improve student success and retention in intro cs. In *Proceedings of the 40th ACM technical symposium on Computer science education*. 9–13.
- [10] Michaela Borg. 2004. The apprenticeship of observation. *ELT journal* 58, 3 (2004), 274–276.
- [11] Rebecca Brent, Jason Maners, Dianne Raubenheimer, and Amy Craig. 2007. Preparing undergraduates to teach computer applications to engineering freshmen. In *2007 37th Annual Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports*. IEEE, FIJ–19.
- [12] Shearon Brown and Xiaohong Yuan. 2014. Experiences with retaining computer science students. *Journal of Computing Sciences in Colleges* 29, 5 (2014), 34–41.
- [13] Sara E Brownell and Kimberly D Tanner. 2012. Barriers to faculty pedagogical change: Lack of training, time, incentives, and... tensions with professional identity? *CBE—Life Sciences Education* 11, 4 (2012), 339–346.
- [14] Andrew J Budd and Heidi JC Ellis. 2008. Spanning the gap between software engineering instructor and student. In *2008 38th Annual Frontiers in Education Conference*. IEEE, S3H–10.
- [15] Elizabeth A. Canning, Katherine Muenks, Dorianne J. Green, and Mary C. Murphy. 2019. STEM faculty who believe ability is fixed have larger racial achievement gaps and inspire less student motivation in their classes. *Science advances* 5, 2 (2019), eaau4734. Publisher: American Association for the Advancement of Science.
- [16] Elizabeth A. Canning, Elise Ozier, Heidi E. Williams, Rashed AlRasheed, and Mary C. Murphy. 2022. Professors who signal a fixed mindset about ability undermine women's performance in STEM. *Social Psychological and Personality Science* 13, 5 (2022), 927–937. Publisher: SAGE Publications Sage CA: Los Angeles, CA.
- [17] Hannah C Chapin, Benjamin L Wiggins, and Linda E Martin-Morris. 2014. Undergraduate science learners show comparable outcomes whether taught by undergraduate or graduate teaching assistants. *Journal of college science teaching* 44, 2 (2014), 90–99.
- [18] Amy Cook, Vinhthuy Phan, and Alistair Windsor. 2022. Improving TA Feedback on In-Class Coding Assignments for Introductory Computer Science. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 1*. 421–427.
- [19] Milton D Cox. 2004. Introduction to faculty learning communities. *New directions for teaching and learning* 2004, 97 (2004), 5–23.
- [20] Holger Danielsiek, Jan Vahrenhold, Peter Hubwieser, Johannes Krugel, Johannes Magenheimer, Laura Ohrndorf, Daniel Ossenschmidt, and Niclas Schaper. 2017. Undergraduate teaching assistants in computer science: Teaching-related beliefs, tasks, and competences. In *2017 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 718–725.
- [21] Adrienne Decker, Phil Ventura, and Christopher Egert. 2006. Through the looking glass: reflections on using undergraduate teaching assistants in CS1. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*. 46–50.
- [22] Paul E Dickson, Toby Dragon, and Adam Lee. 2017. Using undergraduate teaching assistants in small classes. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. 165–170.
- [23] Jessica Ellis, Jessica Deshler, and Natasha Speer. 2016. Supporting institutional change: A two-pronged approach related to graduate teaching assistant professional development. In *Proceedings of the 19th Annual Conference on Research in Undergraduate Mathematics Education*. 729–735.
- [24] Ronald Erdei, John A Springer, and David M Whittinghill. 2017. An impact comparison of two instructional scaffolding strategies employed in our programming laboratories: Employment of a supplemental teaching assistant versus employment of the pair programming methodology. In *2017 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–6.
- [25] Francisco J Estrada and Anya Taffiovi. 2017. Bridging the gap between desired and actual qualifications of teaching assistants: An experience report. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*. 134–139.
- [26] National Center for Education Statistics, Inter university Consortium for Political, and Social Research. 1991. *Integrated Postsecondary Education Data System (IPEDS): Earned Degrees, 1988–1989*. Vol. 9598. Inter-university Consortium for Political and Social Research.
- [27] John Hattie and Helen Timperley. 2007. The power of feedback. *Review of educational research* 77, 1 (2007), 81–112.
- [28] Asma A Khan and Marla Mallette. 2019. Understanding US Undergraduate Students' Perceptions of International Teaching Assistants. *CATESOL Journal* 31, 1 (2019), 133–149.
- [29] Li Li, Joseph P Mazer, and Ran Ju. 2011. Resolving international teaching assistant language inadequacy through dialogue: Challenges and opportunities for clarity and credibility. *Communication Education* 60, 4 (2011), 461–478.
- [30] Yana Malysheva, John Allen, and Caitlin Kelleher. 2022. How Do Teaching Assistants Teach? Characterizing the Interactions Between Students and TAs in a Computer Science Course. In *2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 1–9.
- [31] Julia M Markel and Philip J Guo. 2021. Inside the Mind of a CS Undergraduate TA: A Firsthand Account of Undergraduate Peer Tutoring in Computer Labs. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 502–508.
- [32] Akbar Marvasti. 2005. US academic institutions and perceived effectiveness of foreign-born faculty. *Journal of Economic Issues* 39, 1 (2005), 151–176.
- [33] Diba Mirza, Phillip T Conrad, Christian Lloyd, Ziad Matni, and Arthur Gatin. 2019. Undergraduate teaching assistants in computer science: a systematic literature review. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*. 31–40.
- [34] David J Nicol and Debra Macfarlane-Dick. 2006. Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in higher education* 31, 2 (2006), 199–218.
- [35] Christopher O'neal, Mary Wright, Constance Cook, Tom Perorazio, and Joel Purkiss. 2007. The impact of teaching assistants on student retention in the sciences: Lessons for TA training. *Journal of College Science Teaching* 36, 5 (2007), 24.
- [36] Inna Pivkina. 2016. Peer learning assistants in undergraduate computer science courses. In *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–4.
- [37] Barbara S Plakans. 1997. Undergraduates' experiences with and attitudes toward international teaching assistants. *TESOL quarterly* 31, 1 (1997), 95–119.
- [38] Vijay Anil Ramjattan. 2019. *Working with an accent: The aesthetic labour of international teaching assistants in Ontario universities*. Ph.D. Dissertation. University of Toronto (Canada).
- [39] Stuart Reges. 2003. Using undergraduates as teaching assistants at a state university. *ACM SIGCSE Bulletin* 35, 1 (2003), 103–107.
- [40] Stuart Reges, John McGrory, and Jeff Smith. 1988. The effective use of undergraduates to staff large introductory CS courses. *ACM Sigse Bulletin* 20, 1 (1988), 22–25.
- [41] Emma Riese and Viggo Kann. 2020. Teaching assistants' experiences of tutoring and assessing in computer science education. In *2020 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–9.
- [42] Emma Riese and Viggo Kann. 2022. Training Teaching Assistants by Offering an Introductory Course. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1*. 745–751.
- [43] Eric Roberts, John Lilly, and Bryan Rollins. 1995. Using undergraduates as teaching assistants in introductory programming courses: An update on the Stanford experience. In *Proceedings of the twenty-sixth SIGCSE technical symposium on Computer science education*. 48–52.
- [44] Guido Rößling and Jacqueline Gözl. 2018. Preparing first-time CS student teaching assistants. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. 376–376.
- [45] Donald L Rubin and Kim A Smith. 1990. Effects of accent, ethnicity, and lecture topic on undergraduates' perceptions of nonnative English-speaking teaching assistants. *International journal of intercultural relations* 14, 3 (1990), 337–353.
- [46] Suzanne M Ruder and Courtney Stanford. 2018. Strategies for training undergraduate teaching assistants to facilitate large active-learning classrooms. *Journal of Chemical Education* 95, 12 (2018), 2126–2133.
- [47] Faria Sana, Matthew V Pachai, and Joseph A Kim. 2011. Training undergraduate teaching assistants in a peer mentor course. *Transformative Dialogues: Teaching and Learning Journal* 4, 3 (2011).
- [48] Andries van Dam. 2018. Reflections on an introductory CS course, CS15, at Brown University. *ACM Inroads* 9, 4 (2018), 58–62.
- [49] Arto Vihavainen, Thomas Vikberg, Matti Luukkainen, and Jaakko Kurhila. 2013. Massive increase in eager TAs: Experiences from extreme apprenticeship-based CS1. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*. 123–128.
- [50] Dee AB Weikle. 2016. More insights on a peer tutoring model for small schools with limited funding and resources. *Journal of Computing Sciences in Colleges* 31, 3 (2016), 101–109.