2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #51655

An Engineering TA training Lesson on Problem Solving: Design and Evaluation

Katherine Braught, University of Illinois at Urbana - Champaign
Dr. Blake Everett Johnson, University of Illinois at Urbana - Champaign

Dr. Blake Everett Johnson is a Teaching Associate Professor and instructional laboratory manager in the Department of Mechanical Science and Engineering at the University of Illinois Urbana-Champaign. His research interests include experimental fluid mechanics, measurement science, engineering education, engineering leadership, and professional identity development.

Yael Gertner, University of Illinois at Urbana - Champaign

Dr Gertner joined the Computer Science Department at the University of Illinois in 2020 as a Teaching Assistant Professor. She received her B.S. and MEng in Electrical Engineering and Computer Science from MIT, and Ph.D. in Computer and Information Science at the University of Pennsylvania. She was a Beckman Fellow at the University of Illinois Urbana-Champaign. Her current focus is on broadening participation in Computer Science and Computer Science Education She has been developing materials and teaching for iCAN, a new program for broadening participation in CS for students who have a bachelor's degree in a field other than computer science.

An Engineering TA Training Lesson on Problem Solving: Design and Evaluation

1 Introduction

During the past few decades, engineering programs have experienced substantial growth, increasing demands on instructional resources [1–3]. Universities often need to use graduate and undergraduate teaching assistants (TAs) to manage this growth and provide instruction support. TAs are often used to support instructors through recitations, grading, laboratories, and office hours [4].

Office hours, which are attended primarily by students struggling in a class [5], are a key point of contact between the TAs and these students. TAs are uniquely positioned to influence how students engage with the course material [4]. They are often the ones guiding students through solutions, responding to misunderstanding, and modeling problem solving. As a result, the ways that TAs frame problems and support students' reasoning can have a meaningful impact on the student.

Many engineering programs provide training to TAs. While some TA training focuses on content specific pedagogy or problem solving [6], trainings typically focus on more general content [7]. Most training programs focus on institution policies, grading practices, communication, professionalism, and inclusive teaching [7]. While this training is valuable, TAs are often expected to support students in challenging problem solving engineering courses without any formal preparation.

At the same time, decades of research in mathematics, engineering, and science education have produced well-established frameworks for supporting problem solving [8, 9]. Despite their relevance to TAs' job, these frameworks are rarely incorporated into TA training programs.

In this work, we address this gap by designing and implementing a TA training lesson that explicitly prepares TAs to teach problem solving. Our lesson uses the DHT (Definitions, Heuristics, and Trial and Error) framework, which is a simplified version of Schoenfeld's problem solving behaviors [9]. By grounding TA preparation in established problem-solving frameworks and pedagogical principles, we aim to better align TA training with the instructional role TAs play during office hours and help sessions.

We taught our lesson and framework as an elective in a TA training course in a large R1

University in the Midwest. We also taught the lesson as part of a required graduate engineering TA training course for three semesters. We evaluated our lesson and framework by asking the following questions:

Q1: Are TAs spontaneously using any parts of the DHT framework?

Q2: Can TAs apply the DHT framework?

Q3: What are TAs' perceptions of the quality of the lesson?

Q4: Do TAs from many engineering backgrounds believe the DHT framework could be helpful and applicable?

2 Related Work

2.1 Problem Solving

Teaching problem solving is a central goal of engineering education. Foundational work by Polya [8] established a framework for problem solving. He breaks problems solving down into a four step process:

1. Understanding the Problem
2. Devising a Plan
3. Carrying out a Plan
4. Reflecting on the Solution

Polya's framework is helpful for both teaching problem solving and helping instructors teach problem solving [10].

Beyond Polya's foundational steps, Schoenfeld [9] describes students' successful problem solving behaviors. He considers four dimensions of problem solving:

1. **Resources** or knowledge and facts students may use to solve a problem;
2. **Heuristics** or the strategies that can be used to solve problems;
3. **Control** or students' monitoring and regulation of their problem-solving process; and
4. **Beliefs**, including students' views about mathematics, engineering, and their own abilities.

Together, these components highlight that effective problem solving involves not only applying appropriate content knowledge and strategies, but also monitoring progress and maintaining productive beliefs about the nature of problem solving and learning. Prior work has shown that instructors who explicitly model problem-solving strategies aligned with frameworks Schoenfeld's dimensions can help students develop better strategic reasoning and improve performance [11].

While these frameworks are helpful to instructors teaching problem solving, to the best of our knowledge they have not been translated into a simple training for TAs. Our paper fills this gap by sharing these key ideas with TAs.

2.2 TA training

TAs play a key role in undergraduate engineering education, especially in large course where they support students through office hours, labs, and discussions [4]. Because of their important role, many universities provide training for their TAs. These programs vary widely in their implementation, but typically cover topics like communication, and grading [6].Some trainings cover topics like inclusive teaching [12]. Short workshops, multi-week courses, or mentorship programs are common methods for training TAs [7].

Literature reviews have found that most TA trainings emphasize general knowledge and policies and have little pedagogical training [7, 13]. These reviews also show that TAs receive guidance on classroom management and grading, but only a few programs provide formal training for students conceptual knowledge of course topics [14].

Research has begun to explore alternative models for TA preparation that move beyond basic logistical training. Recent studies examine professional development approaches that emphasize TA identity, inclusive teaching practices, and leadership or facilitation skills within engineering classrooms [12, 15, 16]. Other work investigates structured pedagogical training, mentoring, and reflective practices as ways to support TAs' instructional development and improve the quality of their interactions with students [6]. Collectively, this body of work reflects growing recognition that TA preparation should address not only administrative responsibilities, but also instructional practice.

Collectively, existing research indicates that TA training is widely used as a means of supporting student learning in engineering courses. At the same time, programs' depth and emphasis vary, with many prioritizing broad instructional duties rather than discipline-specific teaching practices. Notably, few studies have directly investigated how TA training can be designed to explicitly foster the teaching of problem solving, even though problem solving is central to engineering education. This gap highlights the need for further research on TA preparation that aligns instructional support with research-based frameworks for problem solving.

3 Our Course

We gave our training in two different contexts. First, we gave our training to graduate computer science TAs in an elective that was part of a required training. Then, we gave our training to a large group of engineering TAs in a required course.

Our university requires new TAs to attend a pre-semester training. The training is made up of required sessions and sessions students elect to take. Our first training was given during this pre-semester training as an elective option for Computer Science (CS) TAs only.

Then, we developed the training to apply to all engineering areas. Graduate Engineering TAs are required to take a semester long training course as new TAs. Our lesson took place during one 50-minute class session. We gave this training the last three semester to over 200 TAs per semester.

4 The DHT Framework

Our TAs start with a solution and must explain how to reach that solution to students. We use definitions, heuristics, and trial and error to help TAs remember key points of the problem solving process. **Definitions** are the basic facts students need to solve the problem, **heuristics** are the steps they need, and **trial and error** is a reminder that students can choose a different heuristic.

The DHT framework mirrors and simplifies Schoenfeld's problem solving behaviors [9]. While his behaviors describe how to explain human problem solving, they can work as a guide for teaching TAs how to explain solutions.

4.1 Definitions

In our training, we encourage TAs to first consider the student's current knowledge level. We call this step definitions because we suggest that TAs check if students have an understanding of all the definitions needed to solve the problem. TAs must know the required knowledge for the problem and we encourage them to think of this before office hours. Similar to Shoenfeld's [9] resources, definitions help TAs confirm that students have the knowledge required to solve the problem.

4.2 Heuristics

After a shared understanding is established in the definitions phase, TAs must demonstrate how to get started solving a problem. Schoenfeld's [9] heuristics suggest that TAs can find a set of tools or approaches to solve a problem. We also call these heuristics. When working with students, TAs should describe the heuristics for solving the problem and explain why they choose this heuristic. TAs can frame problem solving as developing tools for a toolkit and demonstrate to students how they pick their tools. This stage demystifies the process to students and teaches them that practicing helps with developing intuitions about which heuristic to pick. It allows them to see that sometimes it is not known in advance which to pick, but one has to start with something.

4.3 Trial and Error

In the previous subsection, we explained that TAs must explain what heuristic they use and why. However, TAs and students may choose the wrong heuristic. It is important that TAs demonstrate that they can recover from mistakes by trying a different heuristic. TAs should emphasize that this is a natural part of problem solving and experts can get stuck too.

Write down a script of what you would say and how you might normally explain this in your Office Hours or Lab Section.

Solution: $25 \times 2^{-3\log_2(5)} = 25 \times (2^{\log_2(5)})^{-3} = 25 \times 5^{-3} = 25 \times \frac{1}{125} = \frac{1}{5}$

Figure 1: Example Logarithm Solution

Trial and error and the TAs response to mistakes can help students with the control and beliefs systems parts of Schoenfeld's problem solving behaviors. The ability to adjust approaches and accept that mistakes are natural are keys to effective problem solving.

5 Our Lesson

Our lesson introduces TAs to the DHT framework, gives them practice opportunities, and places TAs in the novice student role. We use a combination of examples and active learning to accomplish our learning objectives:

1. Explain how learners benefit from different methods at different points in their learning
2. Take a written solution and explain the thought process of how you get there in terms of definitions, heuristics, and "trying it out"
3. Practice how to provide sufficient details in problem solving that can help students with studying/understanding what they're learning in technical classes.

We gave our training first to a small group of computer science graduate TAs. After analyzing our training, we updated our training to apply to all engineering student with new, generalized examples.

5.1 Small CS Training

We demonstrated the DHT framework through several examples and active learning. First, we asked students to explain the solution to a logarithm problem. Then, we used a recipe metaphor to demonstrate to TAs that students may need different amounts of detail. Next, we explained the DHT framework, using a graph isomorphism problem as an example. Finally, we had TAs practice using the DHT framework by explaining a graph color problem. This training relied on CS graduate students having knowledge of basic concepts.

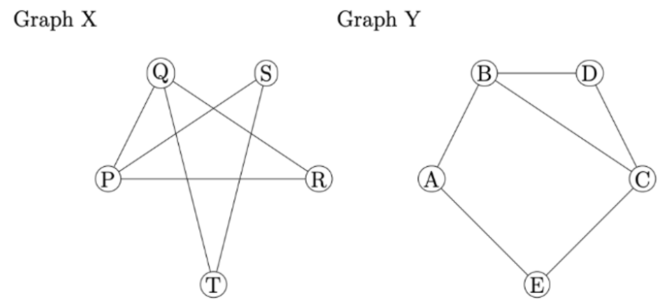
Activity 1: Write a Log Solution

We used a logarithm solution as shown in Figure 1. At the start of the training, we asked TAs to write down how they would explain this solution to a confused student.

Example 1: The Recipe Metaphor

The first example in our lesson is the Recipe Metaphor. When a TA explains a solution, they must carefully choose the level of detail to provide. Too little detail can leave students confused, while too much detail can overwhelm or disengage them. To illustrate this, we

1. (10 points) Are graphs X and Y (below) isomorphic? Justify your answer.



Solution: Yes, they are isomorphic. For example, use the map f, where f(R) = D, f(Q) = B, f(P) = C, f(S) = E, and f(T) = A.

Figure 2: Example 2: A short solution for a Graph Isomorphism problem provided by the instructor

compare problem-solving explanations to baking biscuits. An experienced baker needs only a simple recipe, listing the ingredients and basic steps, because they already know how to perform key techniques. In contrast, a novice baker benefits from a more detailed recipe, with precise measurements and step-by-step instructions, so they can successfully complete each part of the process. For instance, an experienced baker already understands how to “cut” the dough or “rest” it, whereas a novice needs guidance on how to perform these steps correctly.

Activity 2: Reflect

After students wrote their answer to Activity 1, we presented the ”recipe metaphor” in Example 1. Then, we presented a slide with Activity 2, which asked TAs to reflect on what they wrote and whether they had a student in mind when creating their narrative.

Example 2: DHT with Graph Isomorphism

In this example, we used a problem taken from Graph Theory called Graph Isomorphism, see Figure 2. The instructor provided a brief solution to the students. Our example demonstrates how to use the DHT Framework to create a narrative for this solution and provide students with transferable thinking skills that go beyond solving this specific problem.

Definitions The first step in the solution narrative is to provide the necessary definitions. For example, in our graph theory problem, it is key for the TA to state the definition of isomorphism to make sure students understand that it is ”a mapping or a function that is from the vertices of one graph to the vertices of the other that is also edge-preserving”. By naming the concepts explicitly, TAs can help students practice identifying the relevant prior knowledge and understanding the problem goals and constraints.

Heuristics In the second step, TAs need to show students how to get started in solving the problem. We established that to solve this problem, we need to provide a mapping from nodes to nodes. But how do we know which node to start from? Our framework suggests that the way to establish this is through highlighting a set of heuristics that help in solving such problems. In our example of graph isomorphism, the heuristics are matching properties of graphs, such as matching each node to a node of the same degree, or alternatively matching subgraphs. In our example, the nodes Q and P each have degree 3, so they should only be mapped to another node with degree 3, either B or C. At this point the TA should make it clear that we don't know which one to start with but we can try one and see where it leads.

Trial and Error The last step is to highlight the value of trial and error in problem-solving. It is common to revise an approach and even backtrack. Even experienced students have to try multiple approaches. We encourage TAs to normalize this process and create failed solution attempts as part of their narrative. In our example, after mapping node Q to node B we need to map node P to something. We can tell that P is adjacent to Q and therefore might try to map P to A. However, we eventually realize that this is an error. The TA can point out that in retrospect P should not have been mapped to A because they have different degrees.

Activity 3: Write a Graph Coloring Solution

Finally, we asked TAs to practice the DHT framework and write a solution to a graph coloring problem. We chose this problem because it is similar to the graph isomorphism problem, but TAs can practice on a new problem.

5.2 Large Engineering Training

In the large engineering training, we update our activities and examples to not require CS knowledge. First, we used a logarithm example throughout the training, using it as a pre and post activity and to explain the DHT framework. Then, we also added a drawing activity to remind TAs of when they were novices.

Activity 1: Write a Log Solution

We again used the log problem shown in Figure 1 and asked TAs to write down how they would explain this solution to a confused student. Activity 1 is the same in both trainings; however, we used this log example throughout the engineering training.

Example 1: The Recipe Metaphor

We use the same recipe metaphor as in the small CS training.

Example 2: DHT with Log Problem

We also use the log example in Figure 1 to explain the DHT framework. We start by stating that the definitions required for this problem are the logarithm rules. The heuristics are the order in which we apply the rules. Finally, we demonstrate that TAs can model trial and error by allowing students to apply a useless log rule and then work backwards when stuck.

Activity 2: Drawing

An interesting part of our training includes a non-engineering example. Our goal is to put TAs in a novice role to understand how their students may feel. We use a drawing example with active learning.

To put TAs in the novice role, we asked them to draw a cat. Many TAs struggled to draw a cat well. Next, we showed them a "solution," or a drawing of a cat and asked them to draw the cat again. Many TAs could not draw the cat even with the solution. Finally, we showed a step by step process for drawing the cat using shapes and an ordering that builds the shapes into a cap. As a result most TAs could draw the cat. We explained that the definitions in this example are the shapes needed to draw the cat and the heuristics are the step by step directions. TAs were able to draw the cat when given help structured like the DHT framework.

Activity 3: Rewrite a Log Solution

Finally, we asked the TAs to rewrite their solution explanation from the beginning of the lesson. We used the same log problem throughout the training, shown in Figure 1.

6 Methods

To answer our research questions, we used surveys with both quantitative and qualitative data. We also used students active learning responses to understand how they write narratives. We inductively coded students' survey and active learning responses [17, 18]. One author coded each student's response and generated a set of codes. Then, another author independently coded all student responses using the created codebook. The coders discussed and resolved all the differences in coding, generating one set of shared codes for each student's answer.

Q1: Spontaneous Use of the DHT framework

We generated a codebook and coded the initial logarithm narratives. The codes we created are seen in Tables 1 and example responses for each code are seen in Figure 3. We developed these codes because we wanted to see whether TAs would independently, before our lesson, focus on the elements of our framework. For each element, we created a code: student's background (S), include definitions (D), explicitly mention where to get started (H), and how to recover from mistakes (T). Responses that did not include any element got a code (G) and those that included several include them all, e.g. DH.

Q2: Application of the DHT framework

We used the same codes that we generated for Q1 to code TAs' narratives after the lesson. Our goal was to compare improvement in TAs responses before and after the lesson. For example, Figure 4 shows a student response that was coded 'DHT'. Such a response would be an improvement because none of the initial responses had that code, see Table 1. In order to quantitatively compare performance before and after training, we scored TAs responses

Student A (Code G)
Notice the log there. If you recall, Logs are the inverse of exponentiation. So 2^log_2 (5) = 5. How do we remove the log in the problem above?
(Explain log rules at this point if needed)

Student B (Code S)
I am assuming this is a problem whose solution I am allowed to discuss with the student in full. So what I would do is use a board to write each part of the solution one by one, and simultaneously repeat it verbally. If something I would feel necessary to explain, I would look at the student's face to see if he is with me. If not, I would explain it more detail. And allow him to ask questions by giving him some time. The last few lines here are trivial and I can just jot those down.

Student C (Code DH)
1. Solving this problem directly may seem complicated. Instead, we can first simplify the problem using various exponent and log rules to make it easier
2. We can first apply exponent properties to rewrite the second term. Specifically, we can use the following property: (x^(ab)) = (x^a)^b
    a. After applying that property, we simplify the expression: 25 x (2^(log_base2(5))^-3
3. Now, we can recognize that when the base of the log term in the exponent matches the base, we can simplify (x^log_base_x(a) = a).
    a. After applying that property, the problem is simplified: 25 x 5^-3
4. From here, we can simplify the problem since both fractions involve multiples of fives

Student D (Code D)
Okay, let's break this down. The expression we are working with is 25 x 2^(-3 log_2(5). Here, the base of the logarithm is the same as the base of the exponent, which is 2. Using the property a^(log_a(x)) we can simplify the expression to just 5. At this point, the expression becomes 25 x 5^(-3).
Notice that 5^(-3) is equivalent to 1/(5^3). Once we multiply out 5^3, we get 1/125. Now the expression is 25 x 1/125. After multiplying these two, we get 25/125. If we divide the numerator and the denominator by 25, since it is the greatest common factor, we get 1/5.

Figure 3: Example TA responses on Activity 1 with their codes

Code	n	Meaning
G	13	General approach to solving the problem
D	8	Definitions: stating definitions
H	9	Heuristics: Identified a way to begin
T	0	Trial and Error: Identified how to recover from error
DH	9	Used Definitions and Heuristics
S	4	Student Focused Response

Table 1: Activity 1 Codes, n: number of TAs with code

```
Coloring problem:
 Definitions:
  Chromatic number, complete graph
 Heuristics
  Look for complete graphs or more restricted areas for coloring
 Trial and error
  Try to color with fewer colors first
```

Figure 4: Example TA response on Activity 3 after the training, coded DHT

in the following way: TAs received one point for each code of D, H, T, and S. The code G earns 0 points. If students used D, H, and T they received a score of 3.

Q3: TAs' perceptions of the quality of the lesson

In our small CS training, we asked TAs to rate the lesson from 1 (Poor) to 5 (Excellent) on the following: "Quality of session content", "Overall quality of session", "Quality of session presentation". The survey also asked two open-ended questions: "What did you like best about the session?", and "What improvements would you suggest?".

We also asked TAs in our engineering training to rate the lesson from 1 (Poor) to 5 (Excellent) on the following: "Quality of session content", "Overall quality of session", "Quality of session presentation".

Q4: Framework applicability

Finally, we evaluate the improved training given to a large group of engineering TAs. In Fall 2025, we asked TAs quantitative questions to see if they think that this training will apply to their specific course. We have over 200 TAs in each semester of this course.

7 Findings

Q1: Most TAs do not spontaneously use components of our framework.

To understand the narratives used by novice TAs in our pilot group, we examine the coding results for Activities 1. As seen in Table 1, 43 TAs engaged with this activity. None of the students used our full framework. Out of these, only 9 responses were coded with DH, meaning that they described definitions and pointed out a way to get started. 13 TAs'

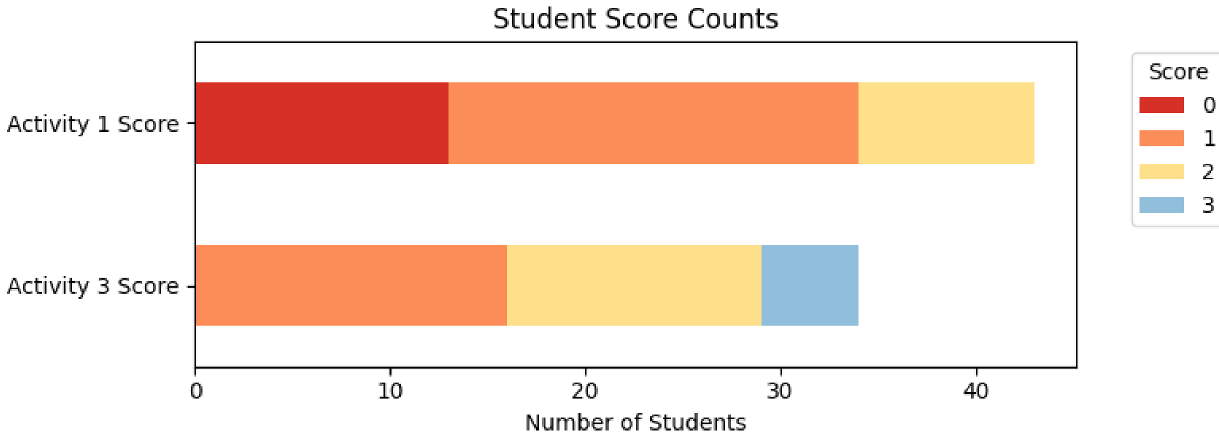


Figure 5: Scores for student responses for Activities 1 and 3

responses, coded with G, did not indicate these at all, describing the solution in a way that only answers the question asked and does not help students answer similar problems. As seen in Figure 5, the average score on Activity 1 is 0.91 out of a possible total of 4 points (SDHT), and none of the students received a 3 or a 4. This suggests that TAs do not spontaneously generate narratives aligned with our framework and that they therefore might benefit from it. Examples of TA responses along with their codes can be seen in Figure 3.

Q2: TAs can apply the framework.

In this subsection, we examine if TAs were able to apply our framework after a one-hour lesson. We scored TAs' responses to Activity 1 and 3 using the same codes. For Activity 1 TAs could have gotten a 4 (SDHT) but the average was 0.9. In Activity 3, we asked TAs to provide an outline of definitions, heuristics, and trial and error rather than their own narrative in free form. As shown in Figure 5, in Activity 3, the majority of TAs had a score of 2 or 3. After our one-hour lesson the TAs were able to apply the framework to their narratives when prompted to do so. It is possible that they might still need more practice before they spontaneously include it in their future narratives. However, our findings suggest that the TAs understood how to apply our framework and how to find the correct definitions, heuristics, and trial and error for a new problem.

Q3: TAs appreciated our lesson.

To assess students' perceptions of our lesson, we analyzed the survey provided by small CS training and large engineering training. As seen in Figure 6, the small CS training evaluations survey included responses from the 53 TA participants. More than 90% rated it as "excellent" or "good" with no students rating it as "poor" on "content", "delivery", or "overall quality". These suggest that TA perceptions of our lesson were favorable. In the large, required engineering course, 175 students responded to the survey. Most students rated the lesson as "excellent" or "good", although some students rated it "poor." Still, these results also suggest most students had positive perceptions of the lesson, especially given it was required.

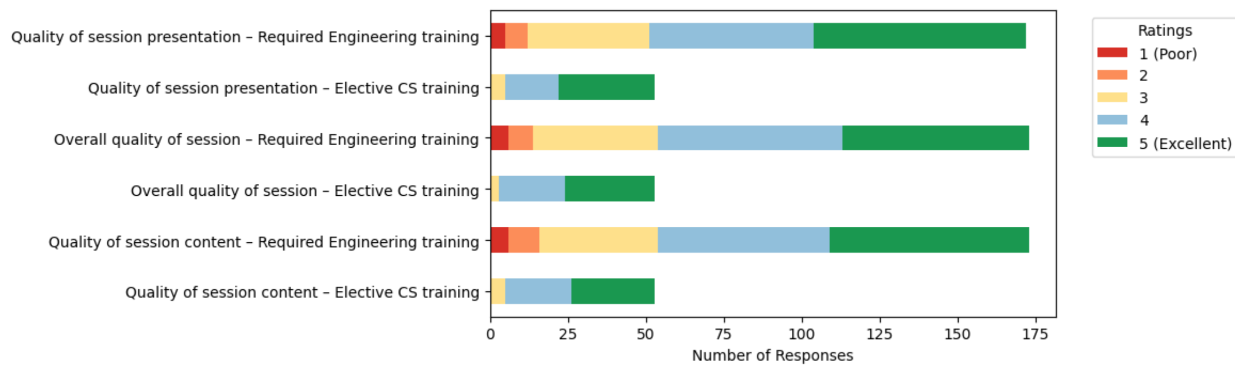


Figure 6: Responses from Post-Survey

In order to understand what TAs like about our lesson, we used responses from the surveys of the CS TAs. We coded their answers to the question "What did you like best about the session?". 26 students responded to it and all responses were positive. 8 students said they liked our framework (F). For example, one student response was: "I think it's very cool to learn and work with a formula on how to explain a solution to a student. One of the things I am nervous about is working in office hours and working with students but this helps create a formula for me to follow in my explanations." 8 students said they liked our activities (A). For example, one student said, "I really liked how interactive this session was and I really liked how they had us practice our problem-solving". The rest said something that was generally good (G), for example, "Practical questions and practice on answering questions".

We also wanted to know what students wanted to improve about our lesson. We coded answers to the question "What improvements would you suggest?". 18 students filled out this question with only 4 indicating something about the content (C), for example that the log problem was confusing and maybe another example would be better. The rest of the improvement suggestions were positive, with 6 TAs saying that there is nothing they would like to improve (G), 2 TAs saying they would like to spend more time on such a lesson (T), and 6 indicating changes in modality (M), for example that they would have liked to meet in person. Overall this indicates that TAs liked our content and that in future lessons we might spend more time by providing more examples from a variety of problems and offer some in-person sessions.

Out of the 53 participants that attended our session, 43 students engaged with Activity 1 and 36 with Activity 3. We hypothesized that the zoom modality of our session might have been a barrier for engagement for some students.

Q4: TAs find the framework applicable to their own classes.

Finally, we evaluated our training on perceived applicability of the TAs, shown in Table 2. We received responses from 176 TAs. About 80% of TAs agreed that they can use the framework and only two students did not think the framework was applicable. These results show that framework is useful to almost all students.

I can apply this to other problems for the class I teach.	
Agree	79
Strongly agree	71
Neutral	24
Disagree	1
Strongly Disagree	1

Table 2: Response count from large, engineering training

8 Conclusion

We have described a framework and training for TAs on problem solving. We fill a gap in literature by developing this training for TAs based on existing research about problem solving. We found that TAs are in need of this training. Most TAs do not spontaneously use parts of our framework when they explain solutions. However, we also saw that TAs could use the framework after our lesson. Their new narratives used the framework much more frequently and all TAs improved. We also found that most TAs enjoyed the lesson; although, the lesson was less enjoyable in for TAs in a required class. Lastly, we saw that all but two TAs in our large, required engineering course thought that the framework was applicable to their course.

Our training and evaluation did have limitations. We were only able to work with students one time for one hour and the training may have been more effective if spaced out during the semester. Additionally, because we only met for a single training, we were not able to evaluate the long-term retention of the lesson by TAs.

In future work, we plan to (1) improve the training, (2) evaluate the training's long-term impact on TAs, and (3) evaluate the impact on students. We will refine our training by evaluating the effectiveness of different types of examples. For example, we may try more technical examples and compare them to fun examples that remind TAs of being novices. We would also like to examine the long-term impact of our training. In the future, we will check with TAs at the end of the semester to see if they retained the information throughout the semester. We also plan to evaluate how this training impacts TAs interactions with students. We plan to see if this training leads to better teaching by TAs and a deeper understanding of concepts for students.

References

[1] E. C. Clausen and J. T. McAllister, "Connecting with students in large classrooms," in *2018 ASEE Midwest Section Conference*, 2018.

[2] K. A. Wingate and M. Holzinger, "Scaling an aerospace engineering senior design program to handle increased enrollment," in *2024 ASEE Annual Conference & Exposition*, 2024.

[3] M. Ball, A. DeOrio, J. Hsia, and A. Blank, "Teaching tas to teach: Strategies for ta

training," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pp. 461–462, 2021.

[4] D. Wilson, "The role of teaching assistants and faculty in student engagement," in *2020 ASEE Virtual Annual Conference Content Access*, 2020.

[5] Z. Gao, S. Heckman, and C. Lynch, "Who uses office hours? a comparison of in-person and virtual office hours utilization," in *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1*, SIGCSE 2022, (New York, NY, USA), p. 300–306, Association for Computing Machinery, 2022.

[6] S. Shehab, J. E. Katz, E. Kirby, M. Pool, Y. W. Chen, and B. E. Johnson, "The impact of a 16-week preparation course on the technological pedagogical content knowledge of graduate teaching assistants in engineering," in *2023 ASEE Annual Conference & Exposition*, 2023.

[7] J. M. O. Hill, J. Agarwal, and J. Kastner, "Training for peer teaching assistants in engineering classrooms: A review," in *2025 ASEE Annual Conference & Exposition*, 2025.

[8] G. Polya, "How to solve it," 1957.

[9] A. H. Schoenfeld, *Mathematical problem solving*. Elsevier, 2014.

[10] J. P. Mullen, "Helping students develop better skills in solving word problems," in *2015 ASEE Annual Conference & Exposition*, pp. 26–842, 2015.

[11] E. Harskamp and C. Suhre, "Schoenfeld's problem solving theory in a student controlled learning environment," *Computers & Education*, vol. 49, no. 3, pp. 822–839, 2007.

[12] I. Paredes, K. Burns, J. Bringardner, R. Li, A. Palav, E. R. Hume, V. Bill, and C. Woods, "Developing inclusive leadership training for undergraduate engineering teaching assistants," in *2023 ASEE Annual Conference & Exposition*, 2023.

[13] M. Di Benedetti, S. Plumb, and S. B. Beck, "Effective use of peer teaching and self-reflection for the pedagogical training of graduate teaching assistants in engineering," *European Journal of Engineering Education*, vol. 48, no. 1, pp. 59–74, 2023.

[14] H. Briel, D. Chadha, and E. J. Hagen, "A comparison of ta training programs across multiple institutions," in *2025 ASEE Annual Conference & Exposition*, 2025.

[15] K. I. Vielma and J. A. Mejia, "Analyzing the needs of engineering teaching assistants: Examining hidden deficit ideas," in *2023 ASEE Annual Conference & Exposition*, 2023.

[16] A. H. Phillips, K. M. Kecskemety, and D. A. Delaine, "First-year engineering teaching assistant training: Examining different training models and teaching assistant empowerment," in *2018 ASEE Annual Conference & Exposition*, 2018.

[17] J. Fereday and E. Muir-Cochrane, "Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development," *International journal of qualitative methods*, vol. 5, no. 1, pp. 80–92, 2006.

[18] J. Corbin and A. Strauss, *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage publications, 2014.