Undergraduate Teaching Assistants in Computer Science: A Systematic Literature Review

Diba Mirza
diba@ucsb.edu
UC Santa Barbara

Phillip T. Conrad
phtcon@ucsb.edu
UC Santa Barbara

Christian Lloyd
clloyd@ucsb.edu
UC Santa Barbara

Ziad Matni
zmatni@ucsb.edu
UC Santa Barbara

Arthur Gatin
agatin@ucsb.edu
UC Santa Barbara

ABSTRACT

We present a systematic literature review of the prior work on Undergraduate Teaching Assistants (UTAs) in Computer Science with two goals: (1) to create a taxonomy of practices that relate to the design and implementation of UTA programs, (2) to identify the benefits of using UTAs as claimed by the literature and characterize the level of evidence for those claims. We analyze 336 excerpts from 40 papers related to these goals. We use content analysis on excerpts describing practices to extract high-level themes that include recruiting, UTA and program coordinator duties, training, evaluation and organization of UTA programs. We perform a more fine-grained analysis within each theme to identify specific questions about UTA programs and the answers provided by the literature. Using a similar technique, we report on the claimed benefits of UTA programs to students, UTAs, instructors and institutions. Our analysis follows well-defined protocols involving multiple reviewers and we report on the inter-rater reliability. The results provided in this paper lay the groundwork for developing evidence-based best practices in UTA programs and inform practice and policy related to the use of UTAs at tertiary institutions. As such, it is relevant to educators establishing a new UTA program, expanding an existing program, or continuously improving an established program, as well as those designing research studies of such programs.

CCS CONCEPTS

• **General and reference → Surveys and overviews**; • **Social and professional topics → Computing education programs**.

KEYWORDS

Undergraduate Teaching Assistants, CS1, CS2

ACM Reference Format:
Diba Mirza, Phillip T. Conrad, Christian Lloyd, Ziad Matni, and Arthur Gatin. 2019. Undergraduate Teaching Assistants in Computer Science: A Systematic Literature Review. In *International Computing Education Research Conference (ICER '19), August 12–14, 2019, Toronto, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3291279.3339422

1 INTRODUCTION

Some institutions have used undergraduates teaching assistants (UTAs) for decades. At these institutions, the benefits—to the students enrolled in the courses, the UTAs themselves, and the institution may be well-established in the minds of decision makers. There are other institutions, however, where there is no history of using UTAs, and those seeking to establish a new program, or expand an existing program may find that those controlling the resources have reasonable doubts. Research to establish evidence-based best practices for UTA programs, and study the benefits of UTA programs can aid in the establishment of new programs and inform both current practice and policy.

The earliest paper we found arguing for the benefits of using UTAs for meeting the challenges of rising CS enrollment appeared in 1988 [36]. Over the subsequent thirty years, dozens of papers have appeared in Computer Science Education venues highlighting promising practices in Undergraduate Teaching Assistant (UTA) programs and highlighting the observed benefits of these programs. We present an actionable overview of this literature examining: (1) practices that can be replicated in new or existing programs and (2) claimed benefits for UTA programs in CS, and the type and strength of the evidence presented for the claimed benefits (whether anecdotal, or something more rigorous).

We acknowledge recent efforts such as the Peer-Teaching Summit conducted at SIGCSE 2019 [1] that aim to gather data about practices outside of what is reported in published work. These efforts are complementary to our work, which is limited to publications about UTA programs. Using the systematic literature review (SLR) approach of Kitchenham and Charters [22], we examined all relevant full papers from the ACM Digital Library and the IEEE Xplore Library. To the best of our knowledge, our paper is the first systematic literature review about UTA programs in Computer Science. The research questions addressed in this survey are:

- **RQ1**: What are the **practices** that have been described or studied in the literature about the design of UTA programs?
- **RQ2a**: What are the claimed **benefits** of UTA programs to the UTAs, students, course instructor, and institutions?
- **RQ2b**: What is the nature of the **evidence** in support of these claimed benefits in the literature?

ICER '19, August 12–14, 2019, Toronto, ON, Canada

ACM ISBN 978-1-4503-6185-9/19/08…$15.00
https://doi.org/10.1145/3291279.3339422

2 RELATED WORK

According to Kitchenham and Charters [23], a systematic literature review (SLR) is one that:

- starts by “defining a review protocol" including the “research question being addressed", and “the methods used"
- has a “defined search strategy that aims to detect as much of the relevant literature as possible"
- “documents [this] search strategy so that readers can assess its rigour and completeness"
- has “explicit inclusion and exclusion criteria to assess each potential primary study"
- specifies “the information to be obtained from each primary study"
- has “quality criteria" to evaluate each primary study

An example of an SLR is a tertiary study performed by Kitchenham et al. in software engineering [22]. Kitchenham and Charters also note that an SLR “is a prerequisite for quantitative meta-analysis". While we do not present any quantitative meta-analyses in this paper, even for the qualitative results we report, the purpose of a systematic approach is the same, i.e. to minimize the risks of bias that may result from a less formal approach.

The SLR reported in this paper is based on the original guidelines of Kitchenham and Charters [23]. As far as possible, we conduct our study within this framework and explain when there is a need to deviate. The steps in the systematic review are described in the sections that follow.

3 METHODOLOGY

At a high level, our methodology is to identify a set of primary sources following a defined search strategy and inclusion/exclusion criteria, extract relevant excerpts, and apply content analysis techniques. Our search queries, shown in Table 1, resulted in 255 candidate papers; each of these papers was reviewed to apply our inclusion/exclusion criteria (Section 3.2) to narrow our study to 40 papers[1]. The bibliography of this paper corresponds to this set of 40 papers, excluding citations [1], [22], [23] and [24].

We then applied a protocol defined in Section 3.3, to extract data from these papers that related to our research questions. We extracted 336 excerpts and contextual data. To minimize bias and subjectivity we extracted these excerpts verbatim (with some glossing to provide context), and deferred any translation or paraphrasing until the final stages of our analysis. We then applied the technique of content analysis [24] to build a hierarchy of practices and benefits that encompass different aspects of UTA programs. Our results are presented as Tables 5 through 13. These tables, together with the discussion in Section 4 and the context information presented in Table 2, constitute our answers to the research questions.

3.1 Search Strategy

Table 1 lists the searches we performed to identify our candidate papers. We conducted our searches using the ACM Digital Library (dl.acm.org) and IEEE Explore library (ieeexplore.ieee.org). The searches were originally performed on August 2, 2018, and updated on Feb 12, 2019.

Table 1: Source searches for 255 candidate papers

source	search terms	in	count
acm	“Undergraduate Teaching Assistant"	abstract	52
acm	“Undergraduate Section Leader"	abstract	3
acm	“UTA"	abstract	10
acm	“Undergraduate TA"	abstract	15
ieee	“Undergraduate Teaching Assistant"	abstract	54
ieee	“Undergraduate Teaching Assistant"	Metadata	73
acm	“Teaching Assistant"	author keywords	32
acm	“Undergraduate TA"	full text	36
acm	"Undergraduate Teaching Assistant"	full text	70

3.2 Inclusion and Exclusion Criteria

For the purposes of our inclusion and exclusion criteria, we define a UTA program as one where undergraduates are assigned to instruction-related duties for a particular offering of a specific course, one in which they are not enrolled, and where they are supervised by the course instructor.

We included papers as our primary sources if they met all of the following criteria:

- Full papers, which may be either experience reports or research papers.
- Papers that are about UTA programs (as defined above) in Computer Science (and possibly other disciplines.)
- Papers that explicitly or implicitly discuss at least one practice, or benefit about UTA programs, with or without evidence.

Our exclusion criteria are listed below. This list may be redundant, as it could be viewed as simply the inverse of our inclusion criteria — however, we have included it to resolve ambiguities. For purposes of this SLR, we excluded the following publications:

- Panels, posters and abstracts (vs. full papers)
- Only describes UTA programs in non-CS disciplines
- Only studies Graduate TA (GTA) training, not UTA training
- Only describes undergrads in mentoring or one-on-one tutoring roles separate from an established UTA program
- Only examines *peer-learning*, defined for our purposes as “undergraduate students enrolled in a particular course assisting other students in the same course.”

3.3 Information Extraction

We extracted three types of information from our primary sources as described in the following sections:

3.3.1 *Context Information.*

We extracted the following context information about the UTA program(s) described in each paper:

(1) Were GTAs included in addition to UTAs?
(2) How large was a typical class?

This information is reported in Table 2. Additional context information was gathered, but is omitted for reasons of space.

3.3.2 *Type and level of evidence.*

As part of our research questions we ask whether the evidence for the benefits claimed in our primary sources consists of anecdotal observations by authors about a single

[1] The counts in Table 1 sum to more than 255 because some papers were found by more than one search.

Table 2: Contexts of Included UTA Programs

Category	Citations
Graduate TAs used?	
Yes	[2], [3], [5], [6], [8], [12], [16], [17], [20], [21], [25], [27], [28], [29], [30], [31], [32], [34], [35], [36], [38], [40], [42]
No	[4], [7], [9], [10], [11], [13], [14], [15], [18], [19], [26], [33], [37], [41] [43], [44]
Not Specified	[39]
Class Size	
<= 20	[8], [9], [13], [14], [34], [43]
21–100	[5], [20], [29], [30], [37], [40], [41]
> 100	[2], [3], [4], [7], [10], [12], [15], [16], [17], [18], [19], [21], [25], [26], [27], [28], [31], [32], [35], [36], [38], [42], [44]
Not Specified	[6], [11], [33], [39]

Table 3: Paper Categories

Category	Citations
Experience Report	[8], [9], [13], [14], [16], [17], [26], [35], [36], [37], [38], [39], [40], [41]
Evaluation	[3], [4], [5], [7], [10], [12], [18], [19], [25], [27], [28], [32], [33], [42], [43], [44]
Research on UTAs	[2], [11], [15], [21], [29], [30], [31], [34]
Research, UTAs incidental	[6], [20]

UTA program at a particular point in time, or something more rigorous. Therefore, we first categorized our sources as one of the following based on the type of evidence presented:

(1) *Experience report*: describes one or more UTA programs.
(2) *Evaluation report*: similar to experience report, but contains some analysis of data.
(3) *Research paper*: (a) contains one or more research question(s) that are stated either explicitly or implicitly (b) uses a clear methodology (qualitative, quantitative or mixed) to investigate the research question(s)

We further categorized research papers as those where some aspect of the use of UTAs was the main focus of the paper, vs. a research paper about another topic where aspects of the use of UTAs were described, but were not the focus of the paper.

The categorization of our primary sources is shown in Table 3. What distinguishes experience reports from evaluation is the presence or absence of data. What distinguishes evaluation from research studies is whether the paper is organized around evaluating the impact of a UTA program. vs. a study designed to answer a specific research question using a well-defined methodology. We categorize post-hoc analysis of survey data as “evaluation" rather than research. For our purposes, a formal research study is one with a priori stated research questions, methodology, analysis, and discussion of threats to validity. This high-level categorization of papers makes it easier to infer the type of evidence for specific claims which we once again select to be one of three levels:

(1) *anecdotal*: claims that are merely asserted, or based on author experience with a UTA program
(2) *evaluated*: claims supported by some analysis of data
(3) *researched*: claims based on a formal research study

3.3.3 Excerpt extraction. Our goal is to compile a thematic list of practices and benefits as reported in our primary sources. The challenge is that most papers do not explicitly use the words “practice" or “benefit" or present this information under well-defined subsections. Since the information we are looking for is not explicitly categorized, we devise a systematic process to arrive at the raw data and use the technique of content analysis to create our list [24].

For this step we followed a divide and conquer approach: four coders independently extracted excerpts from a set of papers assigned to them. We chose to extract excerpts verbatim to avoid the introduction of subjective interpretations early on. An excerpt was extracted if it described a practice about UTA programs or a benefit related to a specific practice or the UTA program as a whole. Each coder read the papers assigned to them thoroughly and highlighted relevant excerpts. On the margins, the coder tagged the excerpt as either a practice or a benefit. The coder then transcribed each highlighted excerpt (almost verbatim) on an excel sheet, making only small grammatical modifications or adding text in brackets to contextualize the excerpt. For each excerpt we recorded the following information: a unique identifier for the paper, the excerpt, justification (if any) and page number. For each paper, the coder recorded context information and categorized the paper as either an experience report, an evaluation or research paper. The excerpts and context data formed the basis of our subsequent analysis.

3.4 Content Analysis

3.4.1 Content Analysis: What is it, why and how do we use it? At its core, content analysis provides a systematic way to go through a large amount of textual data in order to categorize it and discover new themes using explicit rules of coding. Content analysis allows researchers to methodically identify emerging themes in texts and helps organize them into categories.

We applied this technique to the excerpts from our candidate papers, which enabled us to characterize *UTA practices and benefits* and to iteratively build a hierarchy of themes concerning these factors.

3.4.2 Content Analysis: Our Protocol. Here we describe our general protocol for content analysis, factoring out specific parameters. We later report on the values of these parameters for each round of the analysis, and highlight when and why we deviate from this general process.

Inputs: A list of excerpts, each tagged with the following meta data: (a) id of the paper containing the excerpt (this id is unique to each paper), (b) title of the paper, (c) web-link to the original paper.

Outputs: (a) Codebook describing themes or categories and (b) coded excerpts.

Below is an enumeration of the steps in our content analysis protocol:

(1) Define an initial codebook that includes a plausible set of codes.
(2) Discuss the codebook with all coders to arrive at a shared (initial) understanding.
(3) Select a random sample (approximately 10% of the total number of excerpts) for tagging by all coders.
(4) Each coder independently codes the excerpts assigned to them.
(5) Any disagreements are resolved through discussion among coders, which may result in refining the excerpt with added context or updating the codebook. Only the excerpts where a consensus is reached are carried forward to the next phase of our analysis.
(6) Assign the next set of randomly selected samples to all coders (once again 10% of the total excerpts) and repeat steps 4 and 5 a few times.
(7) Coders independently tag all the remaining excerpts.
(8) Calculate the inter-coder reliability of tagging on all excerpts that were tagged by all coders.

Excerpts that don't fit into any of the existing codes are tagged as "other". We discovered new codes by examining these excerpts. The parameters of our protocol are:

(1) N: Number of coders, where $N > 1$
(2) M: Total number of excerpts
(3) M_k: Total number of excerpts used to calculate Cohen's Kappa

3.5 Implementation

We used the protocol described above to iteratively partition the excerpts starting with practices vs. benefits and finally arriving at excerpts with the same type of content. The outcome is a list of unique practices and benefits reported in the literature and citations for each of them.

We applied the protocol described above three times. On the first pass, we separated practices from benefits. On the second pass, we worked within the excerpts tagged as practices to find themes (e.g. recruiting/selection, duties, etc.), and within those tagged as benefits to separate out benefits to UTAs, students, instructors and the institution. For practices, the third pass involved grouping items within each of these themes into questions and answers to those questions. For benefits, the third pass involved grouping benefits into related high-level categories.

Table 4: Parameters used for content analysis

	N	M	M_k	Cohen's Kappa
Pass 1: Practices vs. benefits	3	365	262	0.861
Pass 2: Themes in UTA practices	4	277	96	0.582
Pass 3: Themes in UTA benefits	3	88	74	0.662

The parameters of the protocol in each phase and the inter-coder reliability calculations (Cohen's Kappa) are reported in Table 4. Our Kappa values for Pass 1 through 3 indicate perfect, moderate and substantial agreement, respectively [24]. Note that in Table 4, the total number of excerpts that were tagged ($M = 365$) is more than the number of excerpts extracted from the papers (336). This is because during content analysis we split the excerpts that contained multiple themes.

4 RESULTS

Regarding practices, the themes that emerged were: Recruiting and Selection , UTA duties, UTA training, UTA Program Organization, and evaluation of UTA Programs (Tables 5–9). The benefits to UTAs, students, instructors, and institutions that were mentioned in our primary sources and the level of evidence for each benefit summarized in Tables 10–13. The last column in each of Tables 5–13 cites all the papers that report a specific practice or benefit. These tables comprise our answers to research questions RQ1, RQ2a and RQ2b.

We structure the discussion of our results around a few key questions. While this discussion is not exhaustive, it serves as an example of how the information presented in our tables can be used to ask interesting questions about UTA programs.

What are some common practices and claimed benefits? What are some others that may be less commonly mentioned but are nevertheless noteworthy? The three most common duties of UTAs reported in the literature are: (1) assisting students in labs on programming assignments, (2) leading sections to reinforce concepts, and (3) grading student work. Some of the less common, yet potentially impactful UTA roles are creation and testing of course content, assisting students in lectures, instructing lectures, interviewing future UTAs, and organizing social events. Some of these duties allow UTAs to take on a low-stakes role in instruction, while others challenge them to work with a greater sense of autonomy and responsibility. Understanding the benefits and challenges of the various roles is essential to assess their impact on the professional development, learning, and sense of belonging of UTAs, especially those from underrepresented groups.

The most common benefits of UTA programs claimed by the literature are improvements to UTAs' CS knowledge and interpersonal skills, increased student learning and overall satisfaction, and a greater sense of community within the department.

What are some places where the community differs on practices or claimed benefits? We observe differences in the minimum eligibility criteria and the process of recruiting UTAs, which indicates variations in the selectivity of UTA programs (Table 5). While some programs advocate for giving a majority of students the opportunity to work as UTAs [20], others have a higher bar for entry [12]. Larger UTA programs typically broaden opportunities for entry by having a hierarchical organization, where the roles of junior and senior UTAs are differentiated. Finally, only one paper in our sources discussed recruiting practices to improve the diversity of UTA programs [41].

In Table 6 we observe differences in UTAs' role in grading across different programs. UTAs are involved in grading in many, though not all programs, while individual grading practices vary. Only two papers explicitly state that UTAs were not allowed to grade [42] [43].

Table 5: Practices about Recruiting and Selection

Question/Answer	Sources
Who can apply to be a UTA?	
Students who completed the course with a B or better	[37]
Students who completed the course with a C or better	[43]
Students who passed the course	[14], [19], [41], [43]
4th Year CS students	[3]
2nd Year CS students	[18]
1st Year CS students	[42]
What materials must be submitted to become a UTA?	
An application form	[42], [44]
An application form, resume, and transcripts	[17]
How can program coordinators attract/retain good UTAs?	
Individually invite promising students to apply	[43]
Encourage all students to apply	[14]
Ask faculty for feedback on UTAs	[18]
What are the practices to improve the diversity of UTA programs?	
Practice Strong Affirmative Action	[41]
What are the considerations for hiring UTAs?	
Use a rubric accompanied by a fixed set of interview questions	[25]
Hire students who would benefit from seeing the material again	[18]
Who is involved in the interview process?	
The instructor/program coordinators only	[25]
UTAs in addition to the instructor/program coordinator	[12], [41]
What are applicants asked in a UTA interview?	
General questions regarding the UTA program	[41]
Simulated office hours	[41]
Debug code in real time	[41]
What are the qualities to look for in a UTA?	
Clear and confident speaker	[25], [41]
Active listener	[25]
Holds students attention	[25]
Empathetic towards struggling students	[25]
Interested in improving own teaching	[25]
Prefers interactive teaching methods	[25]
Sensitive to the first-year experience	[25]
Proponent of CS as a major/discipline	[25]
Familiar with course material	[25]
Approachable	[41]
Friendly	[41]
Helpful	[41]
Has successfully demonstrated the Socratic teaching method	[41]
Concerned about the welfare of their students	[36]
Has applied the course material beyond the class	[40]

Table 6: Practices about UTA Duties

Question/Answer	Sources
How do UTAs support student project groups?	
Create student groups based on skill	[9]
Assign projects to student teams	[3]
Oversee and mentor project groups	[3], [9], [16], [40], [44]
Meet weekly with student groups	[10]
Communicate electronically	[44]
Take attendance at meetings	[40]
What are the duties of UTAs in sections?	
Lead section to reinforce course content	[8], [12], [14], [26], [38], [41]
Lead section to guide students on projects and assignments	[12], [28], [41]
How do UTAs contribute to course material?	
Write supplemental material for course projects	[3], [37], [41]
Work with instructors to develop new course activities	[2], [4], [43]
Tailor future homework assignments to suit student needs	[10]
Validate assignment and grading setup	[13]
What are the duties of UTAs in programming labs?	
Lead programming labs	[19], [37]
Assist students in programming labs	[2], [14], [26], [27], [28]
How do UTAs contribute to the grading process?	
Grade student assignments	[2], [10], [19], [27], [37], [39], [41]
Multiple UTAs grade each student assignment	[10], [17]
Assign coarse, preliminary grades to student assignments	[3], [13], [38]
Check for assignment completion	[14]
UTAs do not grade	[42], [43]
UTAs have a grading party	[21]
How do UTAs contribute to lecture?	
Attend lecture	[12], [2]
Assist students in lecture on an individual basis	[2], [13], [14]
Set an example for students in lecture	[14], [43]
What are some of the non-traditional roles of UTAs?	
Each UTA has an auxillary role from A/V to grading	[41]
Act as customers in an agile development setting	[40]
UTAs are instructors for courses	[7]
Do UTAs offer Office Hours?	
UTAs hold Office Hours on a weekly basis	[12], [36], [38], [41]
How do UTAs prepare?	
UTA completes all coursework to better help students with coursework	[18]
What do UTAs do outside of the program?	
UTAs meet for social/community events	[38], [41]

What are the main findings of current research? We summarize the main findings of papers that were categorized as research papers in Table 3.

In one study, undergraduate tutors were used to create small learning groups within very large classes via the structure of "micro-classes" [2]. This work builds on prior work that reported benefits of community-centered learning groups on women and underrepresented students. Although students who participated in micro-classes had higher measures of their sense of community, no statistically significant differences were found compared to a control group. Similar results were reported on the impact of micro-classes on students' performance in courses. Since students in both micro-classes and the control group had access to UTAs, the results speak more to the impact of using UTAs in the format of micro-classes, rather than the impact of having UTAs vs. only graduate TAs.

Two studies reported that UTAs had a significant positive impact on student learning [15] [34]. Another work found that quantitative evaluations paint an incomplete picture of UTAs' performance, reporting no correlation between UTA evaluations and students' final grades [31].

TA marking parties, where instructors and TAs grade collectively, were found to result in faster and more consistent grading [21]. There is clear evidence that organizing marking parties is likely worth the effort and the cost of providing lunch.

Several environmental factors were found to impact UTAs' job satisfaction and interaction with students. These include the presence of collaborative work spaces, good lighting, setting appropriate workload, and the presence of a social support system [29]. Team-teaching was found to be an effective practice for the development of UTAs [30].

What is missing in the literature about UTA programs? While the literature suggests the benefits of UTA programs to student retention and learning in general, the impact on women and underrepresented students is not well-studied. Further, the development of practices related to recruiting, duties, training, organization and evaluation that promote diversity are not well-studied or reported.

We observe that more papers report on the duties for UTA vs. strategies for recruiting UTAs even though we expect every UTA program to have some way of doing both. This goes on to show the lack of a common protocol for reporting on new UTA programs. We believe that our work is a step towards addressing this problem by making important categories of practices explicit. Future authors of experience reports can use our catalog of practices to provide a more complete description of their programs.

Our main observation of the tables related to benefits is that while all of the claimed benefits are plausible, most are anecdotal and very few claims are supported by research. The open challenge is to do a more rigorous study of both the reported benefits, as well as other benefits that may exist but do not appear in our lists. There is a need for tools and frameworks to enable researchers to collect relevant data about these benefits so that they can conduct empirical studies and be able to replicate studies across multiple institutions. Finally, we found limited research that links specific practices to benefits and program goals.

5 THREATS TO VALIDITY

The most important threat to validity is inherent in the very nature of a literature review, namely that we are restricted to reporting only what people write about. The literature is not necessarily a representative sample of UTA programs, or even all aspects of UTA programs. Beyond this, we identify several additional threats to validity.

Our process for excerpt extraction was not as formalized as the other steps. So, we expect some variability in the kinds of excerpts that were extracted by different reviewers. To mitigate this, we followed a more rigorous process to cross-validate the analysis in all the following steps.

More specifically, in content extraction, we faced a choice between simply reading the papers and pulling out what we thought would be relevant, vs. assuming a specific list of categories of benefits and practices up front. Since we were not able to find any previous literature searches in this area, these categories were not previously known. Each approach runs the risk of bias: extracting without categories may reflect personal biases of those doing the content extraction, in terms of what they think is significant or relevant. However, assuming categories up front biases the content extraction process towards the a priori categories. In this case, the assumed categories may be the wrong ones, excluding some relevant content. We chose the former approach in an attempt to gather more information. We recognized that if we had the time to reexamine all of our primary sources a second time, in light of the categories that emerged, we might find more content. Therefore, there is a risk of some omissions. To mitigate this threat we cross validated some of the data reported in the tables related to the discussion in Section 4.

Table 7: Practices about UTA Training

Question/Answer	Sources
Who trains UTAs?	
Computer Science faculty	[43]
Experienced UTAs train new UTAs	[12], [38]
What are alternatives to formal training of UTAs?	
Allowing UTAs to learn through teaching, self reflection, and from their peers	[42]
What structures are used to train UTAs?	
Offer a UTA training course	[8], [35], [38], [41]
What is done a UTA training course?	
Course conducted in a workshop format	[36], [38], [39]
UTAs are trained together for consistency	[32]
New UTAs are trained separately from experienced UTAs	[30]
Training includes self reflection exercises	[16], [35]
UTAs give lectures or presentations	[16], [36]
Training includes student-TA role playing	[2], [7], [36], [39], [41]
Training includes reviewing lesson plans	[7]
What is training course content?	
Learning and teaching styles	[11], [43]
Communication skills	[2], [7], [8], [14], [16], [30], [33], [39], [43]
Grading	[5], [7], [35], [36], [38], [39]
Professionalism (maintaining role as UTA separate from identity as student)	[2], [7], [33], [41], [43]
Responsibilities of UTAs	[2], [7], [11]
Proper teaching techniques	[2], [7], [8], [18], [33], [38], [39], [43]
How to work with difficult students	[7], [35], [36]
Common course issues and their solutions	[7], [35]
How often are UTAs trained?	
Weekly	[2], [7], [16], [17]
Yearly	[2], [8]
Before the academic year	[7], [17], [41]

A related threat to validity concerns how to work with ambiguous excerpts. Occasionally, we had difficulty understanding an excerpt without additional context. Other times, an excerpt was making more than one point and could therefore be tagged in multiple ways. To mitigate this threat, we ensured that in all cases, at least two authors, usually three, and in some cases four, reviewed each excerpt. In any case, where one or more reviewers flagged an excerpt as lacking context, or needing to be split, we held a discussion, went back to the primary source, and came to consensus on either adding context, and/or splitting the excerpt.

Our team of authors consisted of three faculty in CS and two CS undergraduate students. We observed that the student authors read the excerpts more from a student's point of view and often tag “benefit" excerpts as a benefit to the student instead of the institution. As an example, the student reviewers tagged the excerpt “UTAs build a feeling of community" as a benefit to the students, while the faculty tagged it as a benefit to the institution. Since the target audience of this paper is primarily faculty, in these cases we generally deferred to the faculty point of view.

Table 8: Practices about Program Organization

Question/Answer	Sources
How are UTAs compensated for their time?	
Receive pay	[38], [44]
Receive pay and/or credit	[35], [36], [38], [41]
How are courses designed to use UTAs?	
UTAs oversee and guide student groups	[4], [10], [27], [41]
UTAs available to discuss assignments, course content w/ students	[36], [42]
UTAs tailor section to the observed needs of students	[36]
UTAs teach sections in pairs	[10], [29], [30]
UTAs are used to create hybrid courses	[14]
How are UTA roles differentiated?	
UTAs are organized in a hierarchy	[35], [40]
Senior vs. junior TAs	[42]
How often do the UTAs meet?	
Daily	[41]
Weekly	[2], [9], [12], [17], [20], [29], [41], [43]
How do UTAs report section updates	
UTAs report section updates online	[6]
How do UTAs interact in online forums (e.g. "Piazza")	
Students talk with UTAs online, discussing assignments	[2], [3]
How are UTA alumni integrated	
Former UTAs return to discuss their experiences	[41]
What are the duties of a program coordinator/instructor?	
Set expectations for UTAs	[14], [18]
Give constructive feedback to UTAs	[14], [18], [30], [43]
Provide content training	[43]
Listen to UTA input	[43]
Ensure that UTAs have a task at all times	[14]
Plan and lead meetings	[36], [38]
Encourage UTAs to improve their teaching	[36]
Manage expenses	[35]
Write rubrics for UTAs to follow when grading	[38]
Should know the laws/policies regarding UTA programs	[35]
Be available to help UTAs	[7]
What are the duties of experienced UTAs	
Managing course logistics	[17], [41]
Mentor inexperienced UTAs	[10], [17]
Manage overall program logistics	[38]
How do UTAs advance in the program?	
There is a clear path for new UTAs to become Senior UTAs accompanied by substantial pay increase	[17]
What tools are available to assist UTAs?	
Online grading tools and a wiki exist to streamline the grading process	[17]
What are the channels of communication between UTAs and instructors?	
E-mail	[7]
What is the size of the UTA program?	
The department hires > 300 UTAs for a semester	[41]

Evaluation	Weekly	Midterm	End of Term
UTAs evaluate the course	[32], [42]		[18]
UTAs evaluate the UTA program	[42]		[7], [39]
Students evaluate the course	[10]	[41]	[17], [41]
Students evaluate the UTA program	[10]	[7], [41]	[41]
Students evaluate the UTAs			[17], [31], [39]
Instructors evaluate the UTAs	[7]		

Table 9: Practices about Program Evaluation

During content analysis, we found that there are multiple ways to tag some excerpts. We tried to address this later on in our analysis of benefits, where multiple reviewers first validated each excerpt prior to tagging. We believe that this may be the reason for the improvement of our Kappa values from Pass 2 (0.582) to Pass 3 (0.662).

The work for this paper has spanned a few months during which there were some changes to the makeup of the reviewers and also our primary sources. Some of the initial steps were done by faculty alone, later two undergraduate students were added. Also, we originally performed a search and identified a database of 175 papers, time passed and we did a new search to pull in the most recent publications. At the time the second search was performed we found that an important paper that was recently published was not captured by our search criteria. Further inspection showed that our criteria were incomplete because we were only looking for keywords in the abstracts. We then added new criteria (searching for the keywords "Undergraduate TA" or "Undergraduate Teaching Assistant" in the full text). These papers were later put through our pipeline.

6 SUMMARY AND FUTURE WORK

We have presented a systematic literature review of the available literature (through February 12, 2019) in the ACM Digital Library and IEEE Explore that relates to the use of Undergraduate Teaching Assistants. Our review is organized around identifying practices for UTA programs, and the claimed benefits of such programs to the participating UTAs, students, instructors, and institutions. For each practice or benefit identified, we provide an index to the papers that mention this benefit.

The list of practices can be used as a menu of design options for educators establishing a new UTA program, expanding an existing program, or continuously improving an established program. It can also be used as a starting point for researchers to design studies of various practices to assess their effectiveness. We have avoided calling these "best practices" precisely because the evidence to date falls short of establishing any of these with sufficient rigor to earn that title. It is our hope that this work can provide a basis for further study to establish a set of "best practices" for UTA programs.

Similarly, while the use of UTAs at many institutions, in some cases for decades, suggests that there is a strong benefit in such programs, to date, the evidence for such benefits skews towards the anecdotal. We hope that the catalog of benefits we provide in this paper can serve as a basis for further work to establish a more rigorous foundation for the claimed benefits of UTA programs.

		Rigor of Evidence		
High Level Category	**Benefit Observed**	**Anecdotal**	**Evaluated**	**Researched**
Social/Cognitive	Improved connections with faculty	[13], [16]		
	Improves appreciation and motivation for Computer Science	[18], [30]		
	Got interested in teaching as a career	[16], [38]	[18]	
	Developed their interpersonal skills	[13], [14], [28], [35], [44]	[42]	
	Developed positive social connections with peers	[16], [18]		
	Developed an understanding of professional responsibilities	[13]		
Technical Skill	Improved CS knowledge	[5], [13], [14], [18], [33] [34], [38], [41], [43]		
	Performed better in their own coursework	[9]	[18]	
Teaching	Improved confidence in teaching related tasks		[7]	
	Training improved consistency among UTAs	[7]		

Table 10: Benefits to UTAs

		Rigor of Evidence		
High Level Category	**Benefit Observed**	**Anecdotal**	**Evaluated**	**Researched**
Student Performance	UTAs are effective at providing detailed (written) and more frequent feedback on student work	[14]		
	UTAs can provide assistance that is tailored to individual students' needs	[14], [36]		
	UTAs had a significant positive impact on student learning	[43]	[4]	[15], [34]
	Student participation in the classroom and in labs increased	[10], [13], [43]		
	UTAs improved student retention in the course	[34]		
Student Motivation	UTAs helped improve student moral, attitude, and motivation	[13]		
	Existence of the program motivates students to do better to get into the UTA program	[38]		
Student Satisfaction	UTAs improved student satisfaction	[5], [7], [19]	[12], [27]	
Student-UTA Communication	Students find UTAs relatable and approachable	[12], [14], [38], [41]		
	Informal interactions between UTAs and students helped students better navigate the program	[38]		

Table 11: Benefits to Students

		Rigor of Evidence		
High Level Category	**Benefit Observed**	**Anecdotal**	**Evaluated**	**Researched**
Instructor Awareness	Professors are more informed about students and their performance on assignments	[12], [14]		
	UTAs provide a different perspective on the reception of course material	[5], [43]		
Instructor Workload	UTAs ease the workload on the professor	[14], [41]		[10]
	UTAs resolve student problems before they are elevated to the instructor	[44]		

Table 12: Benefits to Instructors

		Rigor of Evidence		
High Level Category	**Benefit Observed**	**Anecdotal**	**Evaluated**	**Researched**
Quality of Instruction	UTAs make it feasible to offer non-traditional course structures	[14]		
Budgeting	UTAs are less expensive than graduate TAs	[38]	[28], [35]	
Community	UTAs create a sense of community that benefits the entire department	[14], [27], [35], [41]		
	UTA programs improved diversity	[35]	[18]	

Table 13: Benefits to Institution

7 ACKNOWLEDGMENTS

We would like to acknowledge the DEERS project for their support of this work (NSF DUE 1525373, 1525173, 1525028), with special thanks to Sarah Heckman for her guidance and mentoring.

REFERENCES

[1] 2019. Peer-Teaching Summit, SIGCSE 2019. http://www.cs.unc.edu/~kmp/peerteachingfellows/. (2019).

[2] 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 (SIGCSE '17)*. ACM, New York, NY, USA, 21–26. DOI: http://dx.doi.org/10.1145/3017680.3017727

[3] 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 (ICSE-SEET '18)*. ACM, New York, NY, USA, 58–67. DOI: http://dx.doi.org/10.1145/3183377.3183387

[4] 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 (SIGCSE '09)*. ACM, New York, NY, USA, 9–13. DOI: http://dx.doi.org/10.1145/1508865.1508872

[5] Saúl A. Blanco. 2018. Active Learning in a Discrete Mathematics Class. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM, New York, NY, USA, 828–833. DOI: http://dx.doi.org/10.1145/3159450.3159604

[6] Kristy Elizabeth Boyer, Robert Phillips, Michael D. Wallis, Mladen A. Vouk, and James C. Lester. 2009. The Impact of Instructor Initiative on Student Learning: A Tutoring Study. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE '09)*. ACM, New York, NY, USA, 14–18. DOI: http://dx.doi.org/10.1145/1508865.1508873

[7] R. Brent, J. Maners, D. Raubenheimer, and A. 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*. F1J–19–F1J–22. DOI: http://dx.doi.org/10.1109/FIE.2007.4418053

[8] Shearon Brown and Xiaohong Yuan. 2014. Experiences with Retaining Computer Science Students. *J. Comput. Sci. Coll.* 29, 5 (May 2014), 34–41. http://dl.acm.org/citation.cfm?id=2600623.2600629

[9] A. J. Budd and H. J. C. Ellis. 2008. Spanning the gap between software engineering instructor and student. In *2008 38th Annual Frontiers in Education Conference*. S3H–10–S3H–15. DOI: http://dx.doi.org/10.1109/FIE.2008.4720516

[10] Mark J. Canup and Russell L. Shackelford. 1998. Using Software to Solve Problems in Large Computing Courses. In *Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education (SIGCSE '98)*. ACM, New York, NY, USA, 135–139. DOI: http://dx.doi.org/10.1145/273133.273178

[11] H. Danielsiek, J. Vahrenhold, P. Hubwieser, J. Krugel, J. Magenheim, L. Ohrndorf, D. Ossenschmidt, and N. Schaper. 2017. Undergraduate teaching assistants in computer science: Teaching-related beliefs, tasks, and competences. In *2017 IEEE Global Engineering Education Conference (EDUCON)*. 718–725. DOI: http://dx.doi.org/10.1109/EDUCON.2017.7942927

[12] 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 (SIGCSE '06)*. ACM, New York, NY, USA, 46–50. DOI: http://dx.doi.org/10.1145/1121341.1121358

[13] Paul E. Dickson. 2011. Using Undergraduate Teaching Assistants in a Small College Environment. In *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education (SIGCSE '11)*. ACM, New York, NY, USA, 75–80. DOI: http://dx.doi.org/10.1145/1953163.1953187

[14] 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 (SIGCSE '17)*. ACM, New York, NY, USA, 165–170. DOI: http://dx.doi.org/10.1145/3017680.3017725

[15] R. Erdei, J. A. Springer, and D. 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)*. 1–6. DOI: http://dx.doi.org/10.1109/FIE.2017.8190650

[16] Francisco J. Estrada and Anya Tafliovich. 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 (ITiCSE '17)*. ACM, New York, NY, USA, 134–139. DOI: http://dx.doi.org/10.1145/3059009.3059023

[17] Jeffrey Forbes, David J. Malan, Heather Pon-Barry, Stuart Reges, and Mehran Sahami. 2017. Scaling Introductory Courses Using Undergraduate Teaching Assistants. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. ACM, New York, NY, USA, 657–658. DOI: http://dx.doi.org/10.1145/3017680.3017694

[18] Meg Fryling, MaryAnne Egan, Robin Y. Flatland, Scott Vandenberg, and Sharon Small. 2018. Catch 'Em Early: Internship and Assistantship CS Mentoring Programs for Underclassmen. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM, New York, NY, USA, 658–663. DOI: http://dx.doi.org/10.1145/3159450.3159556

[19] T. G. Gill. 2005. Learning C++ "Submarine Style": a case study. *IEEE Transactions on Education* 48, 1 (Feb 2005), 150–156. DOI: http://dx.doi.org/10.1109/TE.2004.837044

[20] Luke Gusukuma, Austin Cory Bart, Dennis Kafura, and Jeremy Ernst. 2018. Misconception-Driven Feedback: Results from an Experimental Study. In *Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER '18)*. ACM, New York, NY, USA, 160–168. DOI: http://dx.doi.org/10.1145/3230977.3231002

[21] Brian Harrington, Marzieh Ahmadzadeh, Nick Cheng, Eric Heqi Wang, and Vladimir Efimov. 2018. TA Marking Parties: Worth the Price of Pizza?. In *Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER '18)*. ACM, New York, NY, USA, 232–240. DOI: http://dx.doi.org/10.1145/3230977.3230997

[22] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering - A systematic literature review. *Information and Software Technology* 51, 1 (2009), 7 – 15. DOI: http://dx.doi.org/10.1016/j.infsof.2008.09.009 Special Section - Most Cited Articles in 2002 and Regular Research Papers.

[23] Barbara Kitchenham and Stuart Charters. 2007. *Guidelines for performing systematic literature reviews in software engineering (v2.3)*. Technical Report EBSE-2007-01. Technical Report. Keele University and Durham University Joint Report. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.117.471

[24] Klaus Krippendorff. 2019. *Content analysis an introduction to its methodology* (4 ed.). SAGE.

[25] Dan Leyzberg, Jérémie Lumbroso, and Christopher Moretti. 2017. Nailing the TA Interview: Using a Rubric to Hire Teaching Assistants. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17)*. ACM, New York, NY, USA, 128–133. DOI: http://dx.doi.org/10.1145/3059009.3059057

[26] Alexander Miller, Stuart Reges, and Allison Obourn. 2017. jGRASP: A Simple, Visual, Intuitive Programming Environment for CS1 and CS2. *ACM Inroads* 8, 4 (October 2017), 53–58. DOI: http://dx.doi.org/10.1145/3148562

[27] Mia Minnes, Christine Alvarado, and Leo Porter. 2018. Lightweight Techniques to Support Students in Large Classes. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM, New York, NY, USA, 122–127. DOI: http://dx.doi.org/10.1145/3159450.3159601

[28] Jeff Offutt, Paul Ammann, Kinga Dobolyi, Chris Kauffmann, Jaime Lester, Upsorn Praphamontripong, Huzefa Rangwala, Sanjeev Setia, Pearl Wang, and Liz White. 2017. A Novel Self-Paced Model for Teaching Programming. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale (L@S '17)*. ACM, New York, NY, USA, 177–180. DOI: http://dx.doi.org/10.1145/3051457.3053978

[29] Elizabeth Patitsas. 2012. A Case Study of Environmental Factors Influencing Teaching Assistant Job Satisfaction. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research (ICER '12)*. ACM, New York, NY, USA, 11–16. DOI: http://dx.doi.org/10.1145/2361276.2361280

[30] Elizabeth Patitsas. 2013. A Case Study of the Development of CS Teaching Assistants and Their Experiences with Team Teaching. In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research (Koli Calling '13)*. ACM, New York, NY, USA, 115–124. DOI: http://dx.doi.org/10.1145/2526968.2526981

[31] Elizabeth Patitsas and Patrice Belleville. 2012. What Can We Learn from Quantitative Teaching Assistant Evaluations?. In *Proceedings of the Seventeenth Western Canadian Conference on Computing Education (WCCCE '12)*. ACM, New York, NY, USA, 36–40. DOI: http://dx.doi.org/10.1145/2247569.2247582

[32] Elizabeth Ann Patitsas and Steven Andrew Wolfman. 2012. Effective Closed Labs in Early CS Courses: Lessons from Eight Terms of Action Research. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE '12)*. ACM, New York, NY, USA, 637–642. DOI: http://dx.doi.org/10.1145/2157136.2157318

[33] John Paxton. 2005. Undergraduate Consultation: Opportunities and Challenges. *J. Comput. Sci. Coll.* 21, 1 (Oct. 2005), 231–238. http://dl.acm.org/citation.cfm?id=1088791.1088832

[34] I. Pivkina. 2016. Peer learning assistants in undergraduate computer science courses. In *2016 IEEE Frontiers in Education Conference (FIE)*. 1–4. DOI: http://dx.doi.org/10.1109/FIE.2016.7757658

[35] Stuart Reges. 2003. Using Undergraduates As Teaching Assistants at a State University. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '03)*. ACM, New York, NY, USA, 103–107. DOI: http://dx.doi.org/10.1145/611892.611943

[36] Stuart Reges, John McGrory, and Jeff Smith. 1988. The Effective Use of Undergraduates to Staff Large Introductory CS Courses. In *Proceedings of the Nineteenth SIGCSE Technical Symposium on Computer Science Education (SIGCSE '88)*. ACM, New York, NY, USA, 22–25. DOI: http://dx.doi.org/10.1145/52964.52971

[37] William M. Rivera. 2004. Report on the Implementation of NSF-SFS Scholarship Grant #0210644 and NSF-SFS Capacity Grant # 0210147 from 2002-2004 at Morehouse College. In *Proceedings of the 1st Annual Conference on Information Security*

Curriculum Development (InfoSecCD '04). ACM, New York, NY, USA, 136–142. DOI: http://dx.doi.org/10.1145/1059524.1059556

[38] 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 (SIGCSE '95)*. ACM, New York, NY, USA, 48–52. DOI: http://dx.doi.org/10.1145/199688.199716

[39] Guido Rössling and Jacqueline Gölz. 2018. Preparing First-time CS Student Teaching Assistants. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018)*. ACM, New York, NY, USA, 376–376. DOI: http://dx.doi.org/10.1145/3197091.3205829

[40] A. Scharf and A. Koch. 2013. Scrum in a software engineering course: An in-depth praxis report. In *2013 26th International Conference on Software Engineering Education and Training (CSEE T)*. 159–168. DOI: http://dx.doi.org/10.1109/CSEET.2013.6595247

[41] Andries van Dam. 2018. Reflections on an Introductory CS Course, CS15, at Brown University. *ACM Inroads* 9, 4 (November 2018), 58–62. DOI: http://dx.doi.org/10.1145/3284639

[42] 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 (ITiCSE '13)*. ACM, New York, NY, USA, 123–128. DOI: http://dx.doi.org/10.1145/2462476.2462508

[43] Dee A. B. Weikle. 2016. More Insights on a Peer Tutoring Model for Small Schools with Limited Funding and Resources. *J. Comput. Sci. Coll.* 31, 3 (Jan. 2016), 101–109. http://dl.acm.org/citation.cfm?id=2835377.2835393

[44] C. E. Willis and D. Finkel. 1997. Study of a group project model in computer science. In *Proceedings Frontiers in Education 1997 27th Annual Conference. Teaching and Learning in an Era of Change*, Vol. 1. 299–303 vol.1. DOI: http://dx.doi.org/10.1109/FIE.1997.644861