Inverted Two-Stage Exams for Prospective Learning

Using an Initial Group Stage to Incentivize Anticipation of Transfer

Patrice Belleville, Steven A. Wolfman, Susanne Bradley, Cinda Heeren
University of British Columbia
Computer Science
Vancouver, BC, Canada
patrice@cs.ubc.ca,wolf@cs.ubc.ca,smbrad@cs.ubc.ca,cheeren@cs.ubc.ca

ABSTRACT

We propose a novel, **inverted** two-stage exam format that encourages anticipation of transfer problems. We report on its design, use, and initial assessment for low-stakes quizzes in an algorithms course. A typical two-stage exam, where the group stage comes after the individual stage, emphasizes **retrospective** learning: reflecting on already-solved problems. Our inverted two-stage format places the group stage first, and incentivizes **prospective** learning: preparing for transfer to novel problems that will appear on the individual stage and subsequent assessments.

TAs reported the new format leads to reliable engagement. In surveys, most students preferred inverted two-stage quizzes to individual quizzes plus a TA walkthrough. Students who preferred this format cited the value of learning from peers, brainstorming in the problem domain, and working out ambiguities in the domain and problems.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

algorithms, two-stage exams, formative assessment, transfer

ACM Reference Format:
Patrice Belleville, Steven A. Wolfman, Susanne Bradley, Cinda Heeren. 2020. Inverted Two-Stage Exams for Prospective Learning: Using an Initial Group Stage to Incentivize Anticipation of Transfer. In *The 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20), March 11–14, 2020, Portland, OR, USA*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3328778.3366938

1 INTRODUCTION

In this paper, we describe the design, use, and initial assessment of a novel, **inverted** two-stage exam format used for low-stakes quizzes in an algorithms course. In this format, students complete a group quiz first, followed by an individual quiz. The format is straightforward to adopt and gives a new, incentivized structure in which students get practice transferring their learning.

SIGCSE '20, March 11–14, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-6793-6/20/03…$15.00
https://doi.org/10.1145/3328778.3366938

In the Jan–Apr 2015 term, we "flipped" our algorithms course's lectures to encourage students to grapple with and explore solutions to vague and ambiguous problems. We assessed the course and were happy with the outcomes for many elements. However, students' scheduled weekly one-hour "tutorial" sessions with teaching assistants (TAs) stood out as problematic. They were rated alongside office hours in a cluster of "least helpful" activities, and TAs reported abysmal attendance.

From 2015 to 2018, we worked to improve tutorials. We wanted to: align them with the pervasive groupwork in the course; emphasize exploration of solutions to ambiguous problem statements, which is closely related to the pedagogical notion of *transfer* [10]; and improve their attendance rate and perceived value to students.

We now structure tutorials around frequent, low-stakes quizzes using an adaptation of the two-stage exam format. In a typical two-stage exam, students first complete and submit an exam normally, as individuals. Immediately after completing this individual stage, small groups of students each receive a single, fresh copy of the **same** exam and complete it again as a group. Their group grade forms a small share of the weight for their overall exam grade, incentivizing work on the group stage. Some experimental work has shown substantial learning gains from this format [4, 13].

The typical two-stage format imposes new grading logistics that scaled poorly to our frequent quizzes. Furthermore, the group stage incentivizes retrospective but not **prospective** learning, i.e., students think about what they did wrong but perhaps not about how to face novel problems in the future.

Thus, we use a novel, **inverted** two-stage format. Unlike typical two-stage exams, students **first** work in groups on the assessment and then work individually. In the group stage, each student receives a copy of a domain description and a problem set. Students work on these in small groups but do not submit the work. In the individual stage, each student retains their group stage notes and receives a fresh copy of the domain description and problem set plus a second problem set that extends the first one. They complete the full quiz individually, for a grade. The repeated problems incentivize engagement, while anticipation of the new problems incentivizes preparation to transfer their learning to new problems.

Our assessment of the use of this format suggests that it drives substantially increased attendance, that students find it far more helpful than our 2015 tutorial format, that students prefer it to a traditional quiz format, and that what students value in the format matches our goals of exploring solutions to ambiguous problems.

In the remainder of this paper, we: discuss related work on multi-stage exams and transfer in learning; further describe our algorithms course, particularly the two terms for which we compare

survey results; detail our inverted two-stage quiz design and how it fits into our aligned set of assessments; present and discuss our primarily survey-based assessment of the Jan–Apr 2015 and 2018 course offerings; and conclude, including considering weaknesses in our design and noting some that we are actively addressing.

2 RELATED WORK

"Two-stage", "multi-stage", or "group" exams have been used at least as far back as the 1990s [2]. Team-Based learning's individual and team Readiness Assurance Tests use the same two-stage structure [12]. In their appendix, Gilley and Clarkston lay out what we take to be the "typical" two-stage exam format of a normal individual exam immediately followed by a group stage, with students' final grade for the exam being a weighted combination of these grades [4]. Even within this typical format, there is substantial variation in, e.g., how groups are formed and whether they are stable across the term, the length of the individual and group stages, and the mechanism and weights for computing combined grades.

Some research has shown significant advantages to two-stage exams [4, 13], but other has found less substantive or no positive results [1, 3]. Some of this variation is likely due to the style of question asked, e.g., memorization, conceptual, or applied [1]. Zipp also emphasizes the importance of the course context and alignment of instructional goals to motivate the two-stage format [13]. Thus, we detail below not just our structure but also our course context, especially with respect to assessment.

Ours is not the first multi-stage exam approach to put group work first. Cohen and Henle describe complex, interleaved cycles of individual and group stages in a "pyramid exam" [2]. Breedlove et al. have only a group stage, but effectively add an "individual stage" by allowing students to submit their own individual and independent answers [1]. Lavallee gives an essay-based exam where students first discuss the essay topic together and then individually complete their own essays (from personal communication and [4]). Even providing practice exams to prepare for an exam incentivizes transfer learning, though it does not provide the group-work structure and direct grade incentives of our inverted two-stage format.

In this context, we believe our inverted two-stage format is novel because of the simple-to-adopt structure of repeating the domain statement and one set of problems on both the group and individual stages but adding a new, related set of problems on the individual stage. The structure incentivizes engagement in the group and individual stages and encourages students to specifically plan to *transfer* their learning to new problems.

Transfer in learning is "a student's ability to transfer the knowledge that they've learned and apply it to solving new problems." (page 249 in [10]). Instructors often view transfer as the inevitable outcome of learning, yet research shows that transfer is hard to achieve and comes "naturally" only with substantial expertise. Achieving transfer for students appears to require **explicitly** helping students learn transfer [10].

3 SETTING AND MOTIVATIONS

UBC CPSC 320 "Intermediate Algorithm Design and Analysis" is mainly taught to students in the last two years of a Bachelor's degree in computer science. The course aims for typical algorithms learning outcomes, including:

- Identify the algorithm technique (such as divide and conquer, prune and search, greedy strategies, or dynamic programming) used in a given algorithm.
- Select, adapt and evaluate several promising paradigms and/or data structures for a given problem by analyzing the problem's properties.
- Design and prove the correctness of a solution to a problem using a specified algorithm design paradigm, given sufficient information about the form of that problem's solution.
- Select, evaluate and apply promising mathematical techniques to prove reasonably tight upper and lower bounds on the running time of algorithms.
- Decide how and when to reduce a known problem to another problem of interest, either to obtain an efficient solution to the latter, or to prove that such a solution is unlikely to exist in the context of NP-hardness.
- Explore and apply promising mathematical techniques for modelling and analysis to specify and prove important properties of algorithmic problems and their solutions.

Part of these learning objectives expresses our desire for students to grapple with vague and ambiguous problem statements and explore the space of problems, solutions, and metrics surrounding a particular algorithmic problem. Such ambiguities are common in the real world, and it is important that students become accustomed to evaluate possible interpretations, and choose those (or the only one, as the case may be) that makes sense in a given context.

Since January 2015, lectures have been "flipped", with students completing pre-class readings from Kleinberg and Tardos's Algorithm Design [6], supplemental materials, or watching screencasts of worked examples using labelled subgoals [7, 8]. As measured by COPUS observations [11], class time is about half lecture and half student work, all structured around worksheets that follow the same subgoal approach. The labelled subgoal approach is meant to improve learning and help students adopt an algorithms problem-solving "schema" to assist them in approaching new problems.

Group work pervades the course. We encourage group work in class. Students may work in pairs or trios on assignments. Midterms and often even the final exam are two-stage exams.

3.1 Course Details

Most sessions use the same overall course structure, including the Jan–Apr 2015 and 2018 offerings we focus on. Over the 13 week term, students attend 150 minutes of lecture weekly, in a flipped format structured around a series of ungraded worksheets. To prepare for lectures, students read assigned sections from the textbook and occasionally view screencasts. Worksheets and screencasts use a labelled subgoals approach [7, 8]. Pre-class work is assessed roughly weekly by low-stakes online quizzes. The one to three lecture sections per term have about 100–200 students each and about one lecture support TA per 40 students. We offer about six sections annually, including during the summer.

Students also attend weekly 50 minute tutorial sessions led by TAs with 40–60 students each and about one TA to 20 students. In the 2015 tutorials, students worked on proposed problems in

small groups and then explained (or listened to a TA explain) the solutions. In 2018, tutorial weeks alternated between sessions like these and low-stakes quizzes worth 2% of the course grade. Students also complete an assignment roughly every other week, write two midterm exams, and write one final exam.

Instructors and TAs meet weekly for an hour and discuss what went well or poorly in the previous week, including in tutorials. TA reports in these meetings drove much of the innovation that led to the inverted two-stage exam structure.

The Jan–Apr 2015 offering had 222 students across two lecture sections with 4.5 full-time equivalent TAs (who average roughly 12 hours per week) for a ratio of about 50 students per FTE TA. The Jan–Apr 2018 offering had 355 students across two lecture sections with 8.7 FTE TAs for a ratio of about 40 students per FTE TA.

3.2 Getting to Inverted Two-Stage Quizzes

Prior to January 2017, tutorials were ungraded, hands-on, small-group problem solving sessions facilitated by TAs. Attendance was low (sometimes below 20% after the first few weeks), and so we began exploring the use of different quiz formats for tutorials in January of 2017. Our initial tutorial format was a 25 minute, timed individual quiz followed by a TA walkthrough of the quiz solutions. Teaching assistants reported that a large fraction of students left or disengaged before the walkthrough even began.

In Jan–Apr 2017, we tried a 25 minute individual quiz followed by an *ungraded* 25 minute group quiz, with TAs emphasizing that the group work would lead to faster progress on the assignment and be good practice for two-stage exams. TAs again reported substantial student disengagement.

In Sep–Dec 2017, we explored multiple formats, settling in late October on a plan to "spoil" some of the quiz problems by presenting them during a groupwork session *before* the quiz. Unlike our previous plans, TAs reported that this plan engaged students in discussion and work on problems throughout the tutorial session. We also realized the new format fed directly into our goals for transfer learning and gave students with weaker English skills extra time and support in reading the problem domain.

We used this strategy uniformly in the Jan–Apr 2018 term and in most terms since. About half of the tutorials in a given term use this format, while the other half remain ungraded, hands-on, small-group problem solving sessions facilitated by TAs.

In section 6, we describe survey-based assessment of student's reaction to the inverted two-stage format in the Jan–Apr 2018 term.

4 INVERTED TWO-STAGE DESIGN

In the inverted two-stage format shown in Figure 1, each quiz is divided into three parts: a domain description (part 1), a group set of problems (part 2), and an individual set of problems (part 3). Section 5.1 gives examples of what these parts might look like.

For our 50 minute tutorials, the initial, group stage is 25 minutes long. We aim to give students ample time to solve the problems and move on to discussing ambiguities and hypothesizing about questions that might arise in the individual stage. Each student receives a copy of the first two parts. Students self-organize into groups of three to five. Students then read and discuss the domain description and problems, taking whatever notes they wish. This stage focuses students' attention on prospective learning: understanding and exploring a domain by anticipating problems they may need to solve in that domain in the future, i.e., transfer.

During this stage, we allow students to engage in unusual activities for an exam, like asking questions of the TAs or checking resources online. TAs discourage working across groups, as this could devolve into a whole-class discussion where most students will have no opportunity to actively engage, but they enforce this with a much lighter touch than during individual exams. Students' work on this stage is neither collected nor graded.

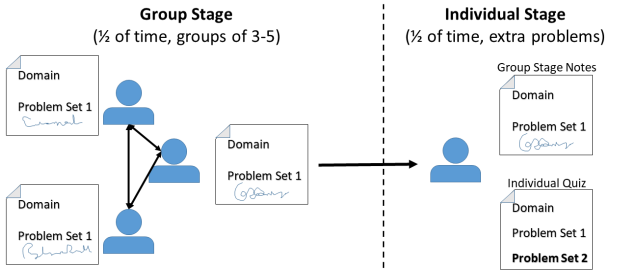

Figure 1: Inverted two-stage format: groups collaborate on a problem set in some domain; they retain their notes for the individual stage, which adds a new, related set of problems.

When the group stage ends, students retain their personal notes. TAs then distribute a fresh copy of the quiz—this time with all **three** parts—to each student. Students complete this stage individually in 25 minutes and submit their solutions for grading. We expect students to finish the repeated group stage problems rapidly and then focus on the problems that are unique to the individual stage.

To reduce grading load, quizzes' answers are short and often partly multiple-choice. We grade any open-ended problems using a coarse rubric with a handful of levels for "essentially correct", "on track but with significant errors", etc. With this format, a single TA normally scanned, uploaded, and graded the quizzes within a week even for the 355 student offering, using the AI in Gradescope[1].

5 ALIGNED ASSESSMENTS

All assessments in the course—but particularly the tutorial quizzes, assignments, and exams—are designed to align with each other and support our course learning objectives, including repeated practice transferring algorithm problem-solving learning to new problems. Students also work cooperatively across many stages of the course: during in-class worksheet exercises, on group stages of quizzes and exams (often including the final exam), and on assignments.

Each tutorial quiz is roughly the equivalent of a single exam problem in size, style, and content. Although tutorial quizzes are low-stakes, we encourage students to treat them as practice for the exams. Each tutorial time-slot has its own quiz, and the collected quizzes form the foundation of the assignment. We release brief solutions for all the tutorial quizzes alongside the assignment at the end of the quiz week. The assignment is then due ten days later, either shortly before a midterm exam or shortly before the next set of tutorial quizzes begin. We then rapidly release much

[1] https://www.gradescope.com/

more detailed solutions to assignments that discuss how and why answers were constructed and why they are correct.

Typically, the assignment builds on the quiz questions by mimicking (asking more of the same type of question), deepening (e.g., asking for justification or explanation of a closed-ended response), extending (e.g., asking follow-on questions about data structure choices for an underspecified algorithm), or altering (e.g., re-asking an optimal matching problem but with an added constraint on expressed preferences). Students are allowed to complete assignments in groups of 2–3. Collaboration rules require each group to produce their own solution but encourage discussion of problems and ideas.

This quiz/assignment alignment can be tricky. Instructors may find it difficult to create sufficient variations for the group quiz, individual quiz, and then assignment. Also, we rely on sufficient "seeds" in the quizzes to lay the groundwork for the assignment without having so many as to make an unwieldy assignment. Further, instructors' design work grows linearly with the number of tutorial section times. Our solution has been to schedule large tutorials so that they are offered at a small number of distinct times, aiming for three to five. Larger tutorials require proportionally larger TA support but only during the group stage and non-quiz weeks, which saves a small amount of TA time. Especially large timeslots can break into two parallel tutorials using the same quiz.

Exam questions are also similar to quiz, assignment, and worksheet problems, although they tend to be more closed-ended than assignments or worksheets. Some exam questions build on ungraded practice problems released in advance of the exam. Unlike the old exams we release for practice, these problems specify that they share a domain or question style with questions on the upcoming exam, which echoes the connection between the group and individual stages of the tutorial quizzes.

The midterm and final exams usually are *normal* two-stage exams, with about two-thirds of the time dedicated to the individual stage. In this and all group assessments, students choose their own groups, choosing anew for each assessment if they prefer.

5.1 A Sample Multi-Stage Problem

As an example of each of these assessment pieces and their alignment, a tutorial quiz contained the following problem domain:

> In the "Stable Wedding" problem (SWP), you are given a set of n guests, a list of "disallowed" subsets of the guests (each with at least two guests), and a list of table sizes. You cannot seat **all** the guests in a disallowed subset together at the same table, but you could seat, e.g., all but one of them. You want to find an assignment of guests to tables (that may leave some guests unassigned) that maximizes the total number of guests at the tables, breaking ties by minimum number of tables used.

The *group portion* of the quiz defined the Independent Set (IS) problem and asked students to complete a partial reduction from IS to SWP. The *individual portion* defined a similar problem SWP2 where one is not allowed to sit **any** of the guests in a disallowed subset at the same table, and asked the students to fill in a single, key blank in a reduction from SWP2 to SWP. The *assignment* discussed yet another (decision) variant called dSWP where each instance had two extra parameters j and k and asked whether a seating assignment existed that sat at least j guests using at most k tables. Finally, the *exam* asked students to establish that dSWP is NP-hard using a reduction from Graph 3-coloring.

A second tutorial quiz contained the following variation of Borůvka's algorithm meant to find a spanning tree of a graph G:

> **Algorithm** Spanning($G = (V, E)$, weights())
>
> Let $G' = (V, E')$ where $E' = \emptyset$
> While G' is not connected
> E-new = $\emptyset$
> For each connected component C of $G' = (V, E')$
> Find an edge $e = (u, v) \in E$ of minimum weight
> connecting a node u in C to a node v *not* in C
> E-new = E-new $\cup \{e\}$
> $E' = E' \cup E$-new
> Return G'

The *group portion* of the quiz asked about the worst case number of iterations of the `while` loop, to draw a graph where this worst case occurs, and whether the algorithm always/sometimes/never returns a tree. The *individual portion* of the quiz contained the first two questions of the group portion, but about the best case number of iterations instead of the worst case, and asked the third question for the case where all edges have different weights. The *assignment* did not ask questions about this algorithm, instead concentrating on problems introduced in that week's other tutorial quizzes, but the *exam* asked to explain why the algorithm always returns a tree when all edge weights are different, and to describe a graph where the worst case number of iterations of the `while` loop occurs and the costliest edge is added in the first iteration of the loop.

6 ANALYSIS

We leveraged larger, ongoing course assessments in 2015 and 2018 and various other data sources to assess our new two-stage design. Particularly relevant to this work were: an end-of-term survey in the Jan–Apr 2015 term, a similar end-of-term survey in Jan–Apr 2018, minutes from weekly staff meetings, and per-assessment grade records. In this section, we describe the relevant portions of the two surveys and our analysis methods in more detail and then present and discuss results from these data sources.

6.1 Methods

2018 Survey: In April 2018, a third-party educational specialist helped us to design an end-of-term survey. They then delivered the survey to students and anonymized and aggregated results. Students received an incentive worth ~0.3% of their course grade if they at least filled in their student number. We received a list of students who earned this bonus but no other identifiable results.

Within the survey we focus on perceived helpfulness of course activities and on the tutorial quizzes. The survey asked "Overall, how helpful or unhelpful were each of the following course activities for your learning in CPSC 320?" with responses of "Very helpful", "Somewhat helpful", "Neither helpful nor unhelpful", "Somewhat unhelpful", or "Very unhelpful". Listed activities were: "required pre-class readings", "assignments", "tutorial quizzes", and "tutorials where there was no quiz". The survey also asked specifically about the quiz format: "Would you recommend continuing the group and

then individual stage format for the tutorial quizzes, or would you recommend an individual stage of the same length followed by a walkthrough of the problems by a TA?" Finally, the survey asked students to briefly explain their reasoning for their recommendation. Of 355 students, 212–213 answered these questions.

Along with graphing quantitative results, we computed the fraction of students who answered either of somewhat or very helpful on each activity (broken down by gender, which was recorded as binary by our institution), and used χ^2 tests for significance on comparisons of interest.

For the open-ended question, one author took a quick-and-dirty open coding approach: reading each response to the prompt, tagging it with existing or—where something novel arose—new categories, and iteratively refining the categories as they went. At the end, they took a pass to merge too-similar categories and eliminate some rarely-used categories. A single response could be tagged with multiple or—for responses like "none"—zero categories.

2015 Survey: In April 2015, a previous third-party educational specialist similarly designed and offered a survey for us, again anonymized but for data needed for an incentive of ~0.3% of course grade. The survey included the question about "course activities" listed for 2018, except it used the words "course resources" instead and asked about more "resources": instructor and TA office hours (separately), assignment sample solutions, exam practice problems, screencasts of worked problems, and the online course discussion board. (The 2018 survey broke these resources out into a separate question which we do not discuss here.) Of 222 students, 180–181 answered these questions.

Other Data Sources: The instructor prepared an agenda for each week's one-hour staff meeting and took minutes. Each agenda included a section to discuss what went well or poorly in the previous week, which is where insights about tutorial mechanisms often came out, including complaints of low attendance in the 2015 term. We used grade records for a conservative estimate of tutorial attendance in 2018 (by non-zero quiz grades, which overlooks a few students who **earned** a zero grade). 2015 tutorials were ungraded.

6.2 Results and Discussion

Figure 2 shows 2015 and 2018 ratings of course activities' helpfulness. In 2015, only 54.1% of students perceived the small-group, TA-facilitated problem-solving tutorials as helpful (98 out of 181). In 2018, the similarly structured non-quiz tutorials received a similar response (128 of 213 or 60.1% found them helpful), but 85% found tutorial quizzes helpful (180 of 213).

In line with this, in 2018, 77.5% of students (165 of 213) indicated they preferred two-stage quizzes over an individual stage and then a TA walkthrough of the solution. Two caveats to this preference are that students hadn't experienced a quiz-plus-walkthrough in our course, and the two-stage format improves grades slightly.

This preference for the two-stage format was stronger for students who identified themselves as women (83.5%) than for those who identified themselves as men (74.0%). Figure 3 shows the helpfulness data on tutorial quizzes and non-quiz tutorials broken down by gender, which also reflects this strong preference.

We analyzed the significance of the difference in preference between tutorial quizzes and non-quiz tutorials in the 2018 survey using a χ^2 test on a two by two table, by comparing the fractions of the students who selected "Very helpful" or "Helpful" for each type of tutorial. This difference was highly statistically significant ($p < 10^{-6}$). We then repeated the analysis for students who identified themselves as men (statistically significant with $p = 0.002$) and women (highly statistically significant with $p < 10^{-6}$).

There was also a statistically significant difference ($p = 0.01$) between the fractions of the women who selected "Very helpful" or "Helpful" for tutorial quizzes, compared with the fraction of men who did. The difference between the fractions of the women who selected "Very helpful" or "Helpful" for non-quiz tutorials, compared with men, was **not** statistically significant ($p = 0.2$). The stronger preference for tutorial quizzes among women is interesting, but we have insufficient qualitative data to explain the difference.

Students' explanations of their reasoning deepen our understanding of the overall preference for the two-stage format. We have non-trivial comments from 129 of the 165 students who preferred two-stage quizzes and 36 of the 48 who preferred a TA walkthrough. Open coding yielded 11 categories in the two-stage group and 9 in the walkthrough, but the first four of the two-stage group and the first two of the walkthrough group dominated. The table below shows the most frequently-used categories.

Table 1: Most frequently-used categories for the students who preferred the two-stage format and then for those who preferred walkthroughs. Percents are of non-trivial comments overall and then of the current preference group.

Category	#	% of all	% of this group
Peers help learning	36	22.0	27.9
Value brainstorming	33	20.1	24.0
Work out meaning	32	19.5	22.5
Prep for individual	24	14.6	18.6
Explaining helps me	9	5.5	7.0
Confident solution	18	11.0	50.0
Need TA time	7	4.3	19.4
Authoritative answer	4	2.4	11.1

Students who preferred the two-stage exam found value in working with peers, brainstorming in the problem domain, working out the ambiguous meaning of domain and problem statements, preparing for the individual stage, and to a much lesser extent explaining themselves aloud. For example, one student said "Gaining insights from my peers is very helpful and helps me gain more insight before the individual parts." A student who preferred two stage exams praised brainstorming with "I liked working with a group to share possible solutions with." Strikingly, one student who preferred a TA walkthrough objected on the same grounds: "Group discussion isn't really helpful for me because I usually find that people have several different approaches to a problem."

By far the most common concern among students who preferred an individual walkthrough was being confident that they knew the solution to the problem: "Most of the time after I finish each quiz, I have no idea if my answers were on track. The provided solutions are also confusing." This may be because we posted only minimal quiz solutions with no explanations to avoid spoiling the

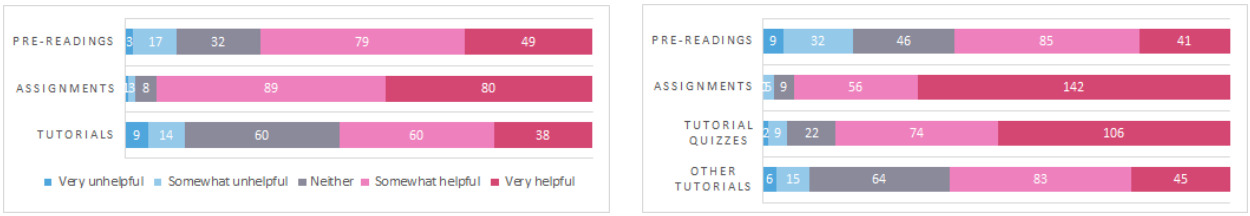

Figure 2: How helpful students found activities in 2015 (left) and 2018 (right), labelled by student counts. In 2015, tutorials are least helpful. In 2018, tutorials without quizzes are rated similarly, but tutorial quizzes are broadly seen as helpful.

Figure 3: How helpful men and women found tutorial quizzes (left) and non-quiz tutorials (right). Key matches Figure 2.

follow-up assignment, while assignment solutions were detailed and annotated. The authoritative answer category was similar but focused on wanting an authority figure to present solutions. In the second category, students wished for more Q&A time with TAs: "In tutorial there is only ten minutes or less for TA explanation. And we don't have an QA session."

Unsurprisingly, given that tutorial quizzes were graded, attendance was high in 2018, between 78 and 90% over the term.

To summarize, many students perceived value in the two-stage format for the reasons we hoped: that it presents a venue in which they need to collaboratively explore the solution space around an ambiguous problem statement. A substantial fraction of the students who dispreferred the two-stage format focused on wanting confidence in their solutions, which might be alleviated by providing detailed, annotated solutions to the quizzes, at least where that doesn't interfere with follow-up assignment questions.

7 CONCLUSIONS AND FUTURE WORK

This paper discusses a novel, **inverted** two-stage exam format that incentivizes **prospective** learning—anticipating and preparing for transfer to novel problems that will appear on the subsequent individual stage—which can be difficult to incentivize otherwise. Key to the format is that students work on the domain and a first set of problems in small groups without submitting and then individually solve for a grade the same set of problems plus a new set that extends it. The repeated problems incentivize engagement, while anticipation of the new problems incentivizes deep exploration to prepare for transfer.

TAs and students (via surveys) report that the format engages students in productive discussion and brainstorming. Most students prefer the inverted two-stage quizzes to individual quizzes plus a TA walkthrough, and students perceive tutorial quizzes as much more helpful than ungraded problem work sessions.

There still remain some substantial questions, however.

We mentioned in Section 5 that work scales linearly with the number of tutorials, which can be exhausting. Reusing or adapting quizzes and assignments from term-to-term can help. One author also experimented with a different approach. In the Summer of 2019, the grading scheme was coarse and based on apparent effort rather than correctness. This reduced cheating concerns. So, the same quiz was used for every tutorial section. Also, since effort is straightforward to judge even on open-ended questions, it allowed a broader choice of question styles in the individual stage. The effort-based quiz grades did correlate modestly with exam grades, but we do not yet have deep insight into this approach.

Another major question is whether our format and incentive structure "scales" to higher-stakes exams. Our incentives encourage students during the group stage to find good answers to the problems and envision what could come next. The same incentives drive them to get help from their peers. However, nothing directly incentivizes students to **give** help to their group. Our structure does not establish positive interdependence—a perception within the group that an individual cannot succeed unless the group succeeds—or individual or group accountability [5]. Indeed, with norm-referenced ("curved") grading, students may have an incentive to hoard their own insights. A normal two-stage exam handles this by requiring students to agree on a single, group answer. Zipp uses an even stronger incentive structure by basing the grade bonus on the difference between the group grade and the average of individuals' grades [13]. However, two-stage exam formats where each individual—rather than the group—submits their own response suffer much the same incentive problem as ours [9].

Next, can our two-stage format be successful outside of our tightly aligned course structure? The pervasive nature of group work in our course and recurring relationships among problems may substantially reinforce the value of our two-stage format.

Most importantly, do inverted two-stage exams actually deliver on transfer learning? Being able to transfer approaches from one problem to a related one is an important skill for a computer scientist, but it is also one that is difficult to learn ([10] section 9.6). Our inverted two-stage format provides students with structured, incentivized practice, but further studies will be required to determine how helpful this practice is at improving their transfer skills.

ACKNOWLEDGMENTS

We thank Jessica Dawson and Alice Campbell for assessment work; our TAs and co-instructors for data, feedback, and guidance; Ido Roll, Brett Gilley, Suzie Lavallee, the CS Ed Reading Group and the anonymous reviewers for advice; and the CPSC 320 students!

REFERENCES

[1] Breedlove, W., Burkett, T., and Winfield, I. Collaborative testing and test performance. *Academic Exchange Quarterly 8*, 3 (2004), 36–40.

[2] Cohen, D., and Henle, J. The pyramid exam. *UME Trends 2* (01 1998).

[3] Fournier, K., Couret, J., B Ramsay, J., and L Caulkins, J. Using collaborative two-stage examinations to address test anxiety in a large enrollment gateway course. *Anatomical sciences education 10* (01 2017).

[4] Gilley, B., and Clarkston, B. Research and teaching: Collaborative testing: Evidence of learning in a controlled in-class study of undergraduate students. *Journal of College Science Teaching 043* (01 2014).

[5] Johnson, D. W., and Johnson, R. T. *The impact of cooperative, competitive, and individualistic learning environments on achievement.* Routledge, 01 2013, pp. 372–374.

[6] Kleinberg, J., and Tardos, E. *Algorithm Design.* Addison Wesley, 2006.

[7] Margulieux, L. E., Guzdial, M., and Catrambone, R. Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research* (New York, NY, USA, 2012), ICER '12, ACM, pp. 71–78.

[8] Morrison, B. B., Margulieux, L. E., and Guzdial, M. Subgoals, context, and worked examples in learning computing problem solving. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research* (New York, NY, USA, 2015), ICER '15, ACM, pp. 21–29.

[9] Reiger, G. W., and Heiner, C. E. Examinations that support collaborative learning: The students' perspective. *Journal of College Science Teaching 43*, 4 (2014), 41–47.

[10] Robins, A. V., Margulieux, L. E., and Morrison, B. B. *Cognitive Sciences for Computing Education.* Cambridge Handbooks in Psychology. Cambridge University Press, 2019, p. 231–275.

[11] Smith, M. K., Jones, F. H. M., Gilbert, S. L., and Wieman, C. E. The classroom observation protocol for undergraduate stem (copus): a new instrument to characterize university stem classroom practices. *CBE Life Sci Education 12*, 4 (2013), 618–627.

[12] Team-Based Learning Collaborative. http://www.teambasedlearning.org/definition/, 2019. [Online; accessed 30-August-2019].

[13] Zipp, J. F. Learning by exams: The impact of two-stage cooperative tests. *Teaching Sociology 35*, 1 (2007), 62–76.