The Research Project in Computer Science Bachelor Education
Undergraduate Research Experience at Scale

Migut, Gosia; Buszydlik, Aleksander; De Weerdt, Mathijs M.

DOI
10.1145/3724363.3729101

Publication date
2025

Document Version
Final published version

Published in
ITiCSE 2025: Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1

Citation (APA)
Migut, G., Buszydlik, A., & De Weerdt, M. M. (2025). The Research Project in Computer Science Bachelor Education: Undergraduate Research Experience at Scale. In *ITiCSE 2025: Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (pp. 410-416). Association for Computing Machinery (ACM). https://doi.org/10.1145/3724363.3729101

The Research Project in Computer Science Bachelor Education: Undergraduate Research Experience at Scale

Gosia Migut*
m.a.migut@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Aleksander Buszydlik*
a.j.buszydlik@student.tudelft.nl
Delft University of Technology
Delft, The Netherlands

Mathijs M. de Weerdt*
m.m.deweerdt@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Abstract

Exposure to research is an important component of undergraduate university education, cultivating critical thinking, problem-solving, and preparation for advanced study. However, providing individual research experiences for large cohorts of undergraduate students poses significant logistical challenges. This paper demonstrates how an undergraduate research experience can be achieved at scale for a large computer science program. Our approach integrates individual research projects into the undergraduate computer science curriculum for up to almost 400 students within a single 10-week course. We describe three key features of our approach: (1) a matching algorithm that assigns students to research projects based on their preferences, (2) peer-group collaboration, and (3) a distributed supervision and assessment model to guide students through key research activities that include reformulating research questions, designing experiments/user studies, and presenting research. Results and feedback indicate that both students and supervisors are satisfied, demonstrating the feasibility and effectiveness of this scalable approach for integrating research experiences into large undergraduate computer science programs.

CCS Concepts

• **Social and professional topics → Computing education**; • **Applied computing → Computer-assisted instruction**.

Keywords

Undergraduate research experience, teaching research methods, peer group collaboration, distributed supervision and assessment, matching projects.

ACM Reference Format:

Gosia Migut, Aleksander Buszydlik, and Mathijs M. de Weerdt. 2025. The Research Project in Computer Science Bachelor Education: Undergraduate Research Experience at Scale. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2025).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3724363.3729101

*All authors contributed equally to this research.

ITiCSE 2025, Nijmegen, Netherlands

ACM ISBN 979-8-4007-1567-9/2025/06
https://doi.org/10.1145/3724363.3729101

1 Introduction

Undergraduate research experiences (UREs) are recognized as an important part of students' education at a university, contributing to the development of critical thinking, problem-solving skills, and academic preparedness. These experiences also improve the diversity in STEM fields by increasing retention and engagement, particularly among underrepresented groups [1]. However, scaling such experiences to accommodate large cohorts of students presents significant logistical and resource challenges. These challenges are particularly pronounced when demand for research opportunities for undergraduate students outpaces faculty availability.

To address them, various scalable models for undergraduate research have been proposed. Course-Based Undergraduate Research Experiences integrate research activities into regular coursework, enabling entire classes to engage in authentic research without requiring individualized mentoring [1, 16, 24, 33]. Programs such as the Early Research Scholars Program adopt a group-based approach, pairing small student teams with mentors to distribute the workload and provide structured guidance [5].

Building on these approaches, best practices have been outlined by the Computing Research Association's Education Committee (CRA-E) [4], including scalable solutions to integrate research into the curriculum, leverage peer mentorship, and utilize structured support systems. These strategies highlight the importance of inclusivity and sustainable resource allocation, addressing barriers to participation for underrepresented groups in computer science.

This paper discusses a full-time 10-weeks long research-oriented course at the Delft University of Technology, which integrates research experience into a large-scale undergraduate computer science program. Our approach enables all students to engage in individual research projects through a thematic peer-group approach under direct supervision of the academic staff (professors) of all CS fields offered at our faculty. Running since 2019, CSE3000 Research Project has accommodated up to almost 400 students in a single run of the course. The approach is based on the CRA-E best practices and incorporates: peer collaboration to simulate a research community, structured organization with decentralized execution, distributed supervision and assessment model that optimizes the use of limited faculty resources, and a matching algorithm that aligns projects with students interests. Results from 12 editions of the course indicate that students and supervisors are satisfied, demonstrating its potential as a scalable and effective solution for integrating research into undergraduate (CS) education.

This paper makes the following contributions:

Presents a peer-group-based, distributed approach for supervising and assessing a large cohort of undergraduate students. Each student enrolled in the undergraduate program conducts individual research, which ensures a meaningful research experience for all. Students are organized into groups according to their research topics and receive guidance from academic staff.

Describes a course design based on constructive alignment aimed at ensuring assessment integrity within distributed supervision approach. It integrates intended learning outcomes, instructional strategies, and assessment mechanisms to create an immersive research environment for undergraduate students.

Demonstrates the effective application of a matching algorithm to assign supervisors' projects to students, ensuring alignment with their preferences and optimizing project allocation.

Analyzes four years of data on student and staff satisfaction, providing key insights and straightforwardly sharing experiences on the implementation of a large-scale research course.

2 Related work

Providing an undergraduate research experience (URE) has been widely examined for its significant influence on students' academic success and development [27]. According to a systematic review of URE models from 2011 to 2021 [1], different models vary in scale, structure, support for students and their embedding, pointing to particular effectiveness of structured, course-based research programs, that improve academic outcomes while fostering inclusion, particularly among underrepresented groups. The review also emphasized the importance of mentorship, the integration of research within the curriculum, and the scalability of URE models to accommodate growing student populations. Despite their benefits, challenges such as resource allocation, faculty availability, and maintaining the quality of student experiences in large-scale programs were noted as critical areas requiring further innovation and support.

In many cases, a URE resembles a scaled-down graduate research experience, with an undergraduate student working on an individual project under the guidance of a faculty supervisor and possibly their graduate student(s) [5]. However, this approach requires tailored guidance, which is often unfeasible in undergraduate programs due to large numbers of students and need for efficiency. One possible solution is working with small collaborative peer groups in which students, together with one or more supervisors, conduct research on the same subject or theme. These so-called theses circles promote collaborative learning and mirror the dynamics of the broader research community [15, 17, 30, 32].

Such peer-group models call for a different approach to supervision [5, 9, 12]. Notably, the group supervision process not only enriches the educational experience for students by promoting collaboration and peer learning, but also significantly streamlines the supervisory workload. Common issues can be addressed collectively rather than individually and consistent guidance can be provided to multiple students simultaneously.

Our approach to organizing an undergraduate research experience presented in this paper follows best practices from the above literature, additionally including mechanisms to scale up the organization and implementation to an unprecedented number of up to almost 400 students per single edition of the course, while ensuring the educational quality, and staff and students satisfaction.

3 Course design

The Research Project is a full-time course — 15 European Credit Transfer and Accumulation System credits, 10 weeks, 40 hours/week — for computer science students in the final phase of their bachelor degree. With a pilot in the academic year 2018/2019, 12 editions of the Research Project have been organized and over 1400 students completed the course until 2024. Every academic year sees a small edition of around 40 students in the first semester and a large edition averaging 319 students in the second semester.

Students who successfully complete the Research Project are expected to show an "in-depth understanding of a selected area of computer science" at the state-of-the-art level (learning objective 1) and should be able to "execute a small research project and report about it" (learning objective 2). The latter encompasses a variety of research skills, such as searching for and critically evaluating the literature, applying and motivating research methodologies, writing a research paper, and (orally) presenting own research.

One of the driving principles behind the Research Project is to offer students the experience of working in a research community. Barring rare individual projects, all students work in peer groups of 4-5 members on a shared research problem but with distinct research questions. The larger edition of the course averages 72 projects, typically involving supervisors from all 16 research groups of the computer science departments at our institution. As such, students have a large degree of freedom in terms of the area of computer science in which they would like to specialize.

During the 10-week runtime of a Research Project edition, students meet with their peer group and supervisors on (at least) a weekly basis to discuss progress and plan next steps. To help them stay on track and attain study goals, the course includes a variety of other learning activities supported by coaches. In particular, it starts with an online module developed by the library of the university on advanced information literacy (IL) skills such as database searches. Since 2023/2024 the course is also preceded by a week-long *Research Methods Bootcamp* that acts as a primer on relevant research skills: (1) designing studies; (2) analyzing academic papers; (3) carrying out literature reviews; and (4) carrying out experiment-based research. Moreover, students may make use of the *Academic Communication Skills* (ACS) and the *Responsible Research* (RR) modules, which include assignments, lectures, tutorials, and group coaching. The former helps students write their papers and present their work, while the latter focuses on responsible and reproducible research.

Every edition of the Research Project starts with students making a research plan, containing a refinement of their research question and a study of relevant literature, that they present to the responsible professor and the supervisor. This is effectively the first formative feedback moment, already in Week 2 of the course. During the presentation, students are expected to show their understanding of the goals of the project. After further four weeks of work — in Week 5, the midterm week — students encounter the second formative feedback moment with their supervisors. They inform the student whether they are on track, behind expectations, or significantly behind expectations. Also in this week, the students present their progress to their peers from other groups, serving both as an opportunity to gather external feedback, and to learn from each other. Next, in Week 7, as the third formative feedback moment, students submit the first draft of their final paper, which is evaluated by their supervisors and two peers from random other groups. The process of peer reviews is carried out using a solution integrated in the learning management software (LMS), and it resembles a double-blind review process in that submissions and comments are fully anonymous and students are encouraged to interact with their peers throughout the week. Students are asked to improve their work based on the received feedback and submit the second draft of their paper in Week 8, which is evaluated by the responsible professor as the fourth formative feedback moment. Finally, in Week 9 students submit their final papers and posters in preparation for the summative assessment in Week 10. The outputs are also displayed in a centrally-coordinated online gallery.[1]

The work of the students is evaluated by committees consisting of their supervisors and an external examiner (we elaborate on this setup in Section 4). Four main criteria contribute to the grade:

(1) *Content*: (a) assessment of related work, (b) research / design methodology, (c) application of the scientific method, (d) interpretation of the results, and (e) responsible engineering.
(2) *Writing*: (f) quality of argumentation, (g) structure of the paper and references, (h) motivation, and (i) language;
(3) *Process*: (j) planning of the project, (k) independence, and (l) peer feedback and communication;
(4) *Presentation and poster*: (m) content and performance.

For each of the above 13 sub-criteria, the committee members may mark insufficient (below a passing grade of 6), sufficient (roughly between a 6 and an 8), or excellent (above an 8).[2] For each of the four main criteria, a partial grade is asked. The grading platform proposes a range of partial grades based on the combination of marks for the sub-criteria. Additionally, we do not require all committee members to evaluate the students on every sub-criterion. For example, the supervisors are deemed to have the best insight into the students' process whereas the examiners only observe the outputs of the project. Thus, the final grades require the consensus of all parties involved in the evaluation. If they cannot agree, the course coordinators get involved as mediators.

4 Peer group distributed supervision

Each group is guided by an assistant, associate, or full professor (referred to as *responsible professor*), supported by a PhD student or a postdoctoral researcher (referred to as *supervisor*). Initially, we allowed projects where these roles were jointly held by one person, but currently we ask all responsible professors to involve an additional supervisor in the project for two reasons. First, this balances the workload of the faculty. Second, the four-eyes principle diminishes the risk that students do not receive adequate supervision and/or experience an unsafe working environment. On top of the responsible professor duties, academic staff typically also fulfills the role of the examiner for a peer group from a different research group. While it may cost a bit more time for colleagues in a different field of CS to assess the work, our goal with this allocation is over time to (1) create a more uniform understanding of what is expected in this course, (2) help ensure impartiality, integrity, and fairness in the assessment process, and (3) increase grading consistency between groups. Examiners are only involved in the last week, grading the students based on papers and presentations. Finally, coaches support students in academic communication and responsible research skills. This structure is shown in Figure 1.

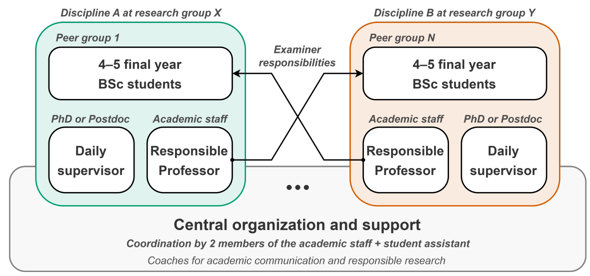

Figure 1: Distributed supervision structure of the course

Wherever possible, we encourage decisions at the lowest possible level, thus operating on the principle of subsidiarity. For example, the central aspects of the final presentations — such as assigning examiners and organizing rooms — remain in the care of the course coordinators. However, the scheduling is independently done by each peer group: the students are asked to contact their examination committees, settle on a date, and submit it to the schedule-tracking platform. Only if the students are unable to find a date that works for everyone in the peer group and the committee, they may contact the course coordinators for alternative arrangements, such as a presentation held outside of the standard timeslots.

While the course remains an organizational challenge due to its large scale, the distributed supervision setup ensures that a course of almost 400 students pursuing 80 independent research projects can run successfully overseen by only two faculty coordinators supported by one teaching assistant. Naturally, during every edition of the course, we remain in close contact with students and staff, but a large majority of groups never need to involve the course coordinators in their process, because communication through the learning management software (LMS) is generally sufficient.

5 Project proposals and assignment

Our experience indicates that the success of a project depends on the motivation of the students, a fit to the expertise (and motivation) of their supervisors, and the feasibility of the projects within the available time. In this section, we describe the goals for project proposals and the related process, which ultimately results in an allocation of students in groups of 4 or 5 to the accepted projects.

5.1 Project proposals

Following the learning objectives of this course, we have established four main evaluation criteria for the project proposals:

- projects can be completed in the available time (10 weeks),
- project proposals provide a truthful representation of (a part of) research at the department(s),
- prerequisite knowledge and/or skills are clear from the proposal,
- project proposals contain 5 research questions or subprojects, allowing each student to write a unique paper.

[1]Please see the poster website: https://cse3000-research-project.github.io.
[2]The mentioned grades are on a scale from 1 to 10, where 10 is the best possible.

About 3-4 months before projects are supposed to start, we inform all professors in the department about the process, with a deadline for submitting proposals through a web-based system. Proposals are a roughly one-page description with a title, prerequisites (if any, we have three variants of 3 courses each in the 2^{nd} year and each student can choose 3 elective courses in the 3^{rd} year which may be listed as a prerequisite), background and motivation, research questions, two Q&A slots, and references to the literature. Additionally, we ask supervisors to provide a plan for how to execute the project (e.g., available code and data, baseline solutions, further references), which is not visible to students. Proposals are labeled by the name(s) of the supervisor(s), their research group, and tags for both the (CS) domain and the research method (e.g., experiment, user study, literature review, reproduction). We ask to provide 1–1.5 proposals on average per professor (and not more than 3). In the past years this resulted in about 100 unique proposals per year for 350–400 students (so 80% of the proposals are allocated).

Each submitted proposal is evaluated by the course staff on the criteria mentioned above, with a focus on the balance between feasibility and openness: when a project is too open, too much time can be lost by students exploring potential research approaches, but when it is presented only as an implementation task, the research experience can be too limited. When students are starting, they are challenged to refine the provided research question and motivation.

To further stimulate the motivation of students, they are also allowed to submit their own proposals. This process results in around 5 extra proposals per year, on average. The course staff then finds a professor with relevant expertise and interests, and supports students in turning their proposal into a project for a group of five. This proposal is subsequently treated similarly to other proposals, with the exception that the proposing students are prioritized in the allocation. Students may also submit project proposals where they join a research group at another institute, and receive remote supervision from a professor of our own university. This process has led to new collaborations with universities around the world.

5.2 Allocation process

We prioritize allocating students to projects that are genuinely interesting for them. In order, our goals are: (1) aligning projects with student interests; (2) allowing students to choose preferred peers; and (3) ensuring a fair distribution of supervision workload.

Allocating nearly 400 students to 80 projects is not trivial, so we use a web-tool that allows students to view proposals, rank them, and optionally indicate group preferences. We expect students to rank at least their top-10 proposals. The provided ranking is used to formulate a mixed-integer linear optimization (MILP) model to compute an optimal allocation of students to projects [11, 25].

We use the so-called *profile-based* optimization variant, which aims to find a Pareto-optimal allocation (so it cannot be improved for one student without harming another) that is also fair in the following sense. Suppose r is the rank of the project allocated to the student receiving the worst allocation across the cohort. We aim to minimize the number of students that receive a project ranked r in their preferences, then for $r-1$, $r-2$, and so on. This is represented as a single MILP by ordered weighted averaging [43]. We have extended the earlier model of [11] in several ways [25]. First, we allocate preferred student groups only if this does not harm the allocation of any of the other students. We do a manual check whether this leads to an acceptable rank for these groups (see Figure 2). Second, we ensure that no supervisor is assigned more than 2 project groups. Third, every year around 1% of students fail to submit their preferences, even after a personal reminder from the course staff. The allocation makes sure that unresponsive students are allocated to groups of five, and that no two of these students end up in the same group. Without this constraint, this could easily be a result of the optimization and has led to a poorly performing group in the past. Finally, for students with self-proposed projects, the allocation is fixed by the course staff. After the allocation has been settled, we manually assign an examiner to each project.

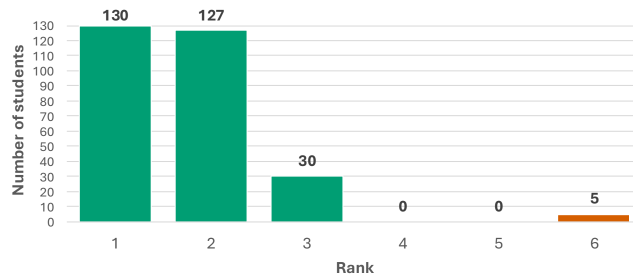

Figure 2: Number of students allocated to a project which they ranked as 1, 2, . . . , 6 (2023); rank 6 was given to a group of students who requested to necessarily work together.

6 Collected data

Given the distributed nature of supervision and evaluation, it is important to discuss to what extent we can ensure unbiased grading. We use the grades for four editions of the course in years 2022/2023 and 2023/2024; they are summarized in numbers in Table 1.

Table 1: Overview of the editions considered in the analysis.

Edition	# Projects	# Students	# Resp. professors	# Supervisors	# Examiners
22/23 Q2	8	38	8	6	8
22/23 Q4	65	304	49	65	61
23/24 Q2	13	59	12	12	13
23/24 Q4	81	393	57	75	57

In Figure 3 we present the complete distribution of grades. With an average of 8.00 (standard deviation of 0.91) after excluding 41 students that dropped out, we believe the grades align with the construction of the evaluation rubric where an 8 generally indicates the strongest sufficient grade. The average partial grades range from 7.80 (or 7.87, if students who failed the course are excluded) for *Writing* to 8.07 (or 8.15) for *Process*, again aligning with the rubric.

Out of 794 students that registered for the course in 2022–2024, 41 cancelled their enrollment before the final presentation (not included in Figure 3), 8 did not cancel their enrollment but dropped out of the project, and further 13 received an insufficient grade for their work, leading to a pass rate of 92.2%. Additionally, our course has strict entrance requirements, e.g., the students may not take other courses along with the Research Project. In 2022–2024, the registration of 117 students was rejected on the grounds of not fulfilling the prerequisites. This means that around 80% of students interested in a given edition of the course complete it successfully.

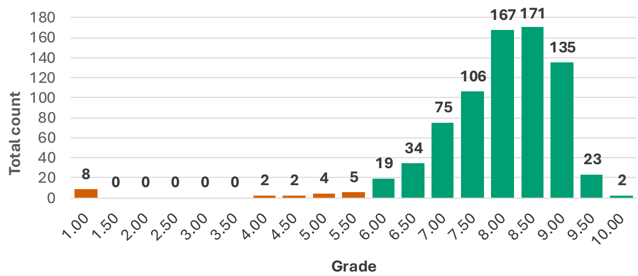

Figure 3: Final grades over four editions in 2022–2024 for students enrolled in the course at the moment of evaluation.

We also looked at the same grading data grouped by the research group of (a) the supervisor and (b) the responsible professor. First, we observed that supervisors tend to give higher grades than responsible professors, with the average final grade, respectively 8.2 and 7.7. This leniency may stem from, e.g., less didactic experience or a tighter social relationship with the students. Still, in our setup, supervisors inform the grades but do not officially grade the students' work and it may even be that this discrepancy encourages discussion, so we do not perceive it as an important problem.

There is also some discrepancy in the evaluation between research groups. For supervisors, in one research group the average grade (based on 52 evaluations) is 2.5 standard deviations below the mean. For responsible professors, also one research group is a clear outlier: the average grade of 8.8 is slightly over 3 standard deviations above the mean. However, this group has supervised only 9 students, where the average is 47, so we simply may not have enough data. Overall, our approach to grading with several "fail-safe" mechanisms, such as a simple rubric, automated suggestions, and course coordinator oversight seems to work well, but it is not without shortcomings. Notwithstanding, these are not specific to the Research Project and we would expect to see similar discrepancies, e.g., in the evaluation of Master's theses.

Finally, to motivate our willingness to award students with high grades, we highlight that students of the Research Project have successfully contributed to many peer-reviewed publications over the years. We are aware of at least 34 papers that were submitted to a variety of venues, including A-level conferences and journals. To our knowledge, 27 of them have been published at the time of completion of this manuscript [2, 3, 6–8, 10, 13, 14, 18–23, 26, 28, 29, 31, 34–42]. Most often, the publications follow from the outputs of individual students, but in at least three cases, the outputs of a peer group were combined into one larger contribution.

7 Student experiences

After every edition of the Research Project, all students are asked to fill a course evaluation form. It includes two open questions that we find particularly interesting to emphasize. The phrasing of the questions remains exactly the same year-to-year, so we can apply a thematic content analysis approach to all responses collected since the academic year 2020/2021. The following paragraphs thus form a reflection on what works well and what requires special attention.

The 3 things I liked best about the Research Project are ... (n = 85). Students most often (n = 42, or 49.4%) point to *supervisors* as the highlight of their Research Project experience. This generally seems to follow from the opportunity to closely interact with academic staff, which is not often the case with large-scale courses. *Autonomy, flexibility, or freedom* are mentioned in 34 (40%) responses; for example, in terms of the ability to approach research from different angles or to structure the process without major constraints from the course. We also find that 22 (25.8%) students discuss the *peer group setup* of the course and further 22 (25.8%) students appreciate the general *organization of the course*. Next, 18 (21.2%) responses highlight the opportunity to *contribute to real-world research*. Other notable themes include: *gaining or practicing academic skills* (14, or 16.5%), *course outputs* and specifically the process of writing a thesis (14, or 16.5%), the *project selection process* (13, or 15.3%), and the *ability to explore a field of computer science* in depth (12, or 14.1%).

"If you were the course manager of this course, what would you do to further improve the course" (n = 84). A major feedback point (25, 29.8%) is related to the *IL, ACS and RR modules* whose lectures (*not* group coaching) have been perceived as not informative enough. Acknowledging that some students may have prior experience in this regard, we decided in recent editions of the course to make these sessions optional while keeping the assignments mandatory. Next, 20 responses (23.8%) point to *communication* as a shortcoming. These comments generally discuss our LMS announcements: we publish information on a need-to-know basis for organizational reasons (i.e., to ensure that all groups are at a similar stage of the process) but this is not appreciated by students who would like to work well ahead. *Scheduling* follows as the subsequent theme with 19 (22.6%) mentions. In 16 (19.0%) of the cases, students discuss *general organization of the course*, which we addressed in several ways over the years, such as consistently improving the course manuals and other LMS materials. Finally, 15 (17.8%) students mention that they have *not received enough feedback*. Given that the course offers seven feedback moments (three from supervisors, two from coaches, two from peers), we believe that this ties to the quality rather than quantity of feedback. Indeed, the *quality of supervisor support* — a major theme with 13, or 15.4% responses — may vary between groups; we now address this by checking in with students midway through the course, and requesting that they raise any complaints to the coordinators as soon as they arise. Some other feedback points include *requirements for final papers and posters* with 14 (16.7%) mentions, and *project selection* in 14 (16.7%) responses.

8 Staff experiences

Since 2019/2020 we also solicit yearly feedback from supervisors and responsible professors. In total, we have received 73 responses and focus the following discussion on three main (open) questions. Examiners are excluded due to their limited involvement.

"Can we establish that the students have met the learning objectives?" (n = 62). We find that 35 (56.5%) responses are an unconditional "yes", meaning that the attainment of the learning objectives could be verified. In the further 25 (40.3%) positive responses, we find respondents who generally agree about the attainment of the learning objectives with minor caveats. First, 8 (12.9%) academic staff members observe that the main objectives may be very difficult to "properly" achieve in only 10 weeks of the project. Second, 7 (11.3%) responses highlight that the attainment of some goals depends on the format of the project. Indeed, certain projects may make it easier for students to demonstrate their expertise. We accommodate for this challenge in the assessment rubric, where marking some criterion as "insufficient", does not necessarily have a major negative impact on the students' evaluation. We observe 2 (3.2%) negative responses, but, unfortunately, they are without explanation.

"Did you enjoy the experience? Did it help your research?" (n = 69). We are happy to report that 57 (82.6%) of the respondents enjoyed their Research Project experience, many mentioning the close collaboration with students as a highlight of their experience. Only six (8.7%) staff members did not enjoy the experience, due to it being very stressful and/or too high of a workload, and further six (8.7%) did not answer the question. Although most of the respondents do not think that the projects were useful for their research, we still find that in 25 (36.2%) cases the projects were at least somewhat helpful. We made three related observations: (1) some projects culminate in publications as discussed in Section 6; (2) some staff members make use of the Research Project as an opportunity to explore a new area of research; (3) especially for PhD students, the Research Project is an opportunity to practice with academic skills. Thus, supervisors and responsible professors who design their projects well can benefit from the Research Project even if their students' work is not sufficiently strong to be published.

"How much time (hours per group) did you spend?" (n = 70). All answers to this question are *ex post facto* estimates, so we cannot draw strong conclusions. Four main components require time investment from the staff: developing a proposal, contact hours with students, asynchronous feedback, and final evaluation. In general, responsible professors estimate their total workload for the whole period to be 15–25 hours; supervisors' responses are more varied, starting at 20 hours but in some cases exceeding 50 hours. Naturally, the amount of time dedicated to student support is the most difficult to predict, as it depends on the complexity of the project and the independence of the students. Our guidelines of two hours per week per group still seem to be sufficient for most projects.

9 Conclusion

The success of the Research Project demonstrates that undergraduate research experiences can be scaled effectively without compromising quality. In line with the literature, the results highlight the efficacy of a peer-group-based approach and distributed supervision to help manage the logistical complexities of large cohorts, while keeping students and staff satisfied. The use of a matching algorithm for project assignment ensures alignment between student preferences, which appears to enhance students' and staff's motivation and engagement. Despite its success, we have encountered notable challenges; we summarize our own experiences in the form of actionable *lessons we learned* in the gray box below. Also, we recognize that the Research Project has been specifically designed for a large CS program at a large university, so it may require some adaptation for smaller institutions. Still, we believe that our experience can be a valuable blueprint for other educators to democratize access to research opportunities, and for other universities to implement (large-scale) URE for their students.

Lessons *we* learned from the Research Project:

- ☞ Interest in the topic is extremely important for success, so the allocation of students to projects should be made carefully (but it can be automated).
- ☞ Decentralizing organization as much as possible helps in dealing with larger numbers of students.
- ☞ Not everyone involved (students and staff included) will follow all instructions, so every essential step (such as publishing proposals, allocating examiners, finding presentation slots) requires a form of verification.
- ☞ Provide all information both statically (for reference) and by notification at the relevant time (e.g., by email).
- ☞ Preparing a proposal and supervising a group of bachelor students is a valuable experience for PhD students.
- ☞ Include preparation on reading literature and scientific writing earlier on in the bachelor curriculum.
- ☞ Intermediate, formative deadlines as soon as one week after the start are important to keep students involved.
- ☞ Half-way (formative) assessment helps give a strong signal to students: some decide to drop out, some start working more effectively.
- ☞ Half-way elicitation of feedback from students helps improve/support supervision.
- ☞ Peer review allows students to see a sample of another paper and reflect on the writing criteria.
- ☞ Tailored tools are essential for many aspects of a large-scale URE: proposing projects, allocating projects, distributed assessment, peer feedback, and evaluation.
- ☞ Discuss the intellectual property rights with the legal department of your institution in advance.
- ☞ Decide on a policy for generative AI tools.
- ☞ In case of computation-heavy projects, ensure student access to a high-performance computing cluster.
- ☞ Keep calibrating your assessment practices to ensure grading consistency.

Acknowledgement

We would like to thank Thomas Overklift Vaupel Klein for contributing to the assessment data analysis, Philippe Louchtch who analyzed various optimization models for the student-project allocation for his MSc thesis project [25], and the team developing the web-interface, called ProjectForum, in particular Taico Aerts.

References

[1] Zubair Ahmad and Noora J. Al-Thani. 2022. Undergraduate Research Experience Models: A systematic review of the literature from 2011 to 2021. *International Journal of Educational Research* 114 (2022), 101996. doi:10.1016/j.ijer.2022.101996

[2] Nele Albers, Mark A. Neerincx, Nadyne L. Aretz, Mahira Ali, Arsen Ekinci, and Willem-Paul Brinkman. 2023. Attitudes Toward a Virtual Smoking Cessation Coach: Relationship and Willingness to Continue. In *Persuasive Technology*. Springer Nature Switzerland, Cham, 265–274.

[3] Patrick Altmeyer, Giovan Angela, Aleksander Buszydlik, Karol Dobiczek, Arie van Deursen, and Cynthia C. S. Liem. 2023. Endogenous Macrodynamics in Algorithmic Recourse. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE Computer Society, Los Alamitos, CA, USA, 418–431. doi:10.1109/SaTML54575.2023.00036

[4] Christine Alvarado, Michael Hilton, Amy J. Ko, Lori Pollock, Kelly Shaw, and Neil Spring. 2020. *Best Practices for Expanding Pathways to Undergraduate CS Research*. Technical Report. Computing Research Association 2013 Education Committee (CRA-E).

[5] Christine Alvarado, Joe Hummel, Diba Mirza, Renata Revelo, and Lisa Yan. 2022. Scaling and Adapting a Program for Early Undergraduate Research in Computing. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, 50–56. doi:10.1145/3478431.3499336

[6] Agathe Balayn, Gaole He, Andrea Hu, Jie Yang, and Ujwal Gadiraju. 2022. Ready Player One! Eliciting Diverse Knowledge Using A Configurable Game. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. ACM, New York, NY, USA, 1709–1719. doi:10.1145/3485447.3512241

[7] Yessin Beyhan and Maria Soledad Pera. 2023. Covering Covers: Characterization Of Visual Elements Regarding Sleeves. In *Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*. ACM, New York, NY, USA, 28–33. doi:10.1145/3563359.3597404

[8] Prajit Bhaskaran and Tom Viering. 2022. To Tune or not to Tune: Hyperparameter Influence on the Learning Curve. BNAIC/BeneLearn 2022. Student Theses.

[9] David Broman. 2015. *A Process for Student Group Supervision*. Technical Report. KTH Royal Institute of Technology.

[10] Zhiyi Chen, Marco Loog, and Jesse H. Krijthe. 2023. Explaining Two Strange Learning Curves. In *Artificial Intelligence and Machine Learning*. Springer Nature Switzerland, Cham, 16–30.

[11] Marco Chiarandini, Rolf Fagerberg, and Stefano Gualandi. 2019. Handling preferences in student-project allocation. *Ann Oper Res* 275, 1 (2019), 39–78.

[12] Jocelyn B. Dautel. 2020. Applying a collective academic supervision model to the undergraduate dissertation. *Psychology Teaching Review* 26, 1 (2020), 18–26.

[13] Natanael Djajadi, Amirhossein Deljouyi, and Andy Zaidman. 2025. Using Large Language Models to Generate Concise and Understandable Test Case Summaries. In *ICPC 2025*. ACM, New York, NY, USA, 5 pages.

[14] Neda Džiugaitė, Baris Ardic, and Andy Zaidman. 2024. What Are Massive Open Online Courses (MOOCs) Teaching About Software Testing?. In *Proceedings of the 5th ACM/IEEE International Conference on Automation of Software Test (AST '24)*. ACM, New York, NY, USA, 204–208. doi:10.1145/3644032.3644469

[15] Amanda S. Fernandez. 2024. Generalizing the CS Course-based Undergraduate Research Experience (CURE). In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*. ACM, New York, NY, USA, 1642–1643. doi:10.1145/3626253.3635502

[16] Fahmida Hamid. 2020. A course-based undergraduate research experience (CURE) in computer science: an experience report. *J. Comput. Sci. Coll.* 35, 6 (2020), 56–65.

[17] Ya Ping Hsiao and Kamakshi Rajagopal. 2023. Support Student Integration of Multiple Peer Feedback on Research Writing in Thesis Circles. In *The Power of Peer Learning: Fostering Students' Learning Processes and Outcomes*, Omid Noroozi and Bram De Wever (Eds.). Springer International Publishing, Cham, 47–70. doi:10.1007/978-3-031-29411-2_3

[18] Tim Huisman, Jacobus G. M. van der Linden, and Emir Demirović. 2024. Optimal Survival Trees: A Dynamic Programming Approach. *Proceedings of the AAAI Conference on Artificial Intelligence* 38 (03 2024), 12680–12688. doi:10.1609/aaai.v38i11.29163

[19] Andra Ionescu, Kiril Vasilev, Florena Buse, Rihan Hai, and Asterios Katsifodimos. 2024. AutoFeat: Transitive Feature Discovery over Join Paths. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 1861–1873. doi:10.1109/ICDE60146.2024.00150

[20] Donghwi Kim and Tom Viering. 2022. Different approaches to fitting and extrapolating the learning curve. BNAIC/BeneLearn 2022. Student Theses.

[21] Eduardo M. Lanzini and Sara M. Beery. 2021. Image-to-Image Translation of Synthetic Samples for Rare Classes. In *CV4Animals Workshop: Computer Vision for Animal Behavior Tracking and Modeling*.

[22] Enrico Liscio, Alin E. Dondera, Andrei Geadău, Catholijn M. Jonker, and Pradeep K. Murukannaiah. 2022. Cross-Domain Classification of Moral Values. In *Findings of the Association for Computational Linguistics: NAACL 2022*. Association for Computational Linguistics, Seattle, United States, 2727–2745. doi:10.18653/v1/2022.findings-naacl.209

[23] Ailin Liu, Pepijn Vunderink, Jose Vargas Quiros, Chirag Raman, and Hayley Hung. 2024. How Private is Low-Frequency Speech Audio in the Wild? An Analysis of Verbal Intelligibility by Humans and Machines. In *Proceedings of Interspeech 2024*. 2725–2729. doi:10.21437/Interspeech.2024-1258

[24] David Lopatto. 2004. Survey of Undergraduate Research Experiences (SURE): First Findings. *Cell Biology Education* 3, 4 (2004), 270–277. doi:10.1187/cbe.04-07-0045

[25] Philippe Andreevic Louchtch. 2025. *Improving group project matchings of TU Delft's Project Forum*. Master's thesis. Delft University of Technology.

[26] Stavros Makrodimitris, Bram Pronk, Tamim Abdelaal, and Marcel Reinders. 2023. An in-depth comparison of linear and non-linear joint embedding methods for bulk and single-cell multi-omics. *Briefings in Bioinformatics* 25, 1 (11 2023). doi:10.1093/bib/bbad416

[27] National Academies of Sciences, Engineering, and Medicine. 2017. *Undergraduate Research Experiences for STEM Students: Successes, Challenges, and Opportunities*. The National Academies Press, Washington, DC. doi:10.17226/24622

[28] Nienke Nijkamp, Carolin Brandt, and Andy Zaidman. 2021. Naming Amplified Tests Based on Improved Coverage. In *21st International Working Conference on Source Code Analysis and Manipulation (SCAM)*. IEEE Computer Society, Los Alamitos, CA, USA, 237–241. doi:10.1109/SCAM52516.2021.00036

[29] Wessel Oosterbroek, Carolin Brandt, and Andy Zaidman. 2021. Removing Redundant Statements in Amplified Test Cases. In *21st International Working Conference on Source Code Analysis and Manipulation (SCAM)*. IEEE Computer Society, Los Alamitos, CA, USA, 242–246. doi:10.1109/SCAM52516.2021.00037

[30] Kamakshi Rajagopal, Emmy Vrieling-Teunter, Ya Ping (Amy) Hsiao, Inge Van Seggelen-Damen, and Steven Verjans. 2024. Guiding thesis circles in higher education: towards a typology. *Professional Development in Education* 50, 2 (2024), 315–332. doi:10.1080/19415257.2021.1973072

[31] Pablo Rodrigo, Johan Pouwelse, and Martijn de Vos. 2021. UniCon: Universal and Scalable Infrastructure for Digital Asset Management. In *Proceedings of the 2nd International Workshop on Distributed Infrastructure for Common Good*. ACM, New York, NY, USA, 5—-10. doi:10.1145/3493426.3493823

[32] A. G. L. Romme and J. F. H. Nijhuis. 2002. *Collaborative Learning in Thesis Rings*. Wolters-Noordhoff, Groningen, The Netherlands.

[33] Audrey Smith Rorrer, Joseph Allen, and Huifang Zuo. 2018. A National Study of Undergraduate Research Experiences in Computing: Implications for Culturally Relevant Pedagogy. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, 604–609. doi:10.1145/3159450.3159510

[34] Katja Geertruida Schmahl, Tom Julian Viering, Stavros Makrodimitris, Arman Naseri Jahfari, David Tax, and Marco Loog. 2020. Is Wikipedia succeeding in reducing gender bias? Assessing changes in gender bias in Wikipedia using word embeddings. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*. ACM, New York, NY, USA, 94–103. doi:10.18653/v1/2020.nlpcss-1.11

[35] Dimitri Stallenberg, Mitchell Olsthoorn, and Annibale Panichella. 2022. Improving test case generation for REST APIs through hierarchical clustering. In *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering*. IEEE Press, 117–128. doi:10.1109/ASE51524.2021.9678586

[36] Andrei C. Tociu, Yasin I. Tepeli, and Joana P. Gonçalves. 2023. Importance Weighting Domain Adaptation Beyond Specific Target Domains. BNAIC/BeneLearn 2023. Student Theses.

[37] Martin Toman and Neil Yorke-Smith. 2021. Localised Reputation in the Prisoner's Dilemma. In *BNAIC/BeneLearn 2021*. 761–763.

[38] Jules van der Toorn, Ruben Wiersma, Abbie Vandivere, Ricardo Marroquim, and Elmar Eisemann. 2022. A New Baseline for Feature Description on Multimodal Imaging of Paintings. In *Eurographics Workshop on Graphics and Cultural Heritage*. The Eurographics Association. doi:10.2312/gch.20221223

[39] Mim Van Den Bos, Jacobus G. M. Van Der Linden, and Emir Demirović. 2024. Piecewise Constant and Linear Regression Trees: An Optimal Dynamic Programming Approach. In *Proceedings of the 41st International Conference on Machine Learning*, Vol. 235. PMLR, 48994–49007.

[40] Ruben S. Verhagen, Alexandra Marcu, Mark A. Neerincx, and Myrthe L. Tielman. 2024. The Influence of Interdependence on Trust Calibration in Human-Machine Teams. In *Proceedings of HHAI 2024*. 300–314. doi:10.3233/FAIA240203

[41] Ruben S. Verhagen, Mark A. Neerincx, Can Parlar, Marin Vogel, and Myrthe L. Tielman. 2023. Personalized Agent Explanations for Human-Agent Teamwork: Adapting Explanations to User Trust, Workload, and Performance. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (London, United Kingdom) *(AAMAS '23)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2316–2318.

[42] Emiel Witting, Yasin I. Tepeli, and Joana P. Gonçalves. 2023. Deep Domain Adaptation without Access to a Specific Target Domain. BNAIC/BeneLearn 2023. Student Theses.

[43] Ronald R. Yager. 1997. On the analytic representation of the Leximin ordering and its application to flexible constraint propagation. *European Journal of Operational Research* 102, 1 (1997), 176–192.