Interactive and Iterative Peer Assessment: A Social Choice Approach to Eliciting and Aggregating Student Evaluations

Lihi Dery

Abstract

We present a peer assessment model grounded in computational social choice, aimed at mitigating grade inflation and strategic behavior in the evaluation of in-class project presentations. The model elicits individual evaluations through a structured rating process, augmented with pairwise comparison queries, to produce complete and tie-free rankings for each student. Aggregation proceeds in two stages: first, computing the median score for each project; second, applying a ranking-based voting rule to resolve ties. This hybrid approach combines the interpretability of score-based methods with the robustness of ordinal social choice mechanisms. Deployed in a university course, the model significantly reduced both grade inflation and cognitive load, demonstrating the value of integrating elicitation design with voting-based aggregation in peer assessment contexts.

1 Introduction

Peer assessment is a structured procedure where individuals in a group evaluate the performance of their peers who share a similar status within that group [44]. This study concentrates on peer assessment in an academic classroom [45], and specifically on the quantitative peer assessment (or peer grading) of oral class presentations [17, 51]. In this educational setting, students present their projects during class and have the opportunity to evaluate their peers' work and receive assessments themselves. During the assessment activity, students are organized into distinct sessions; in each session, students take turns presenting their work. Peer assessment is done with the understanding that students' performance as assessors and presenters contributes to their final grades.

Even though some studies have described peer assessment systems, and there exists a survey of tools (albeit a nearly 15-year-old one) [24], little attention has been paid to the methodologies used for collecting and aggregating peer assessment data. These methodologies are the focus of this research. They are needed since several challenges emerge when students are requested to assign scores (ratings) to projects. The first two may also arise in other rating scenarios, and the latter is specific to peer assessment settings:

(a) Calibration and standardization. Rating on a wide range numerical scale (e.g., $0-100$) is not always meaningful for students, who might have difficulty in choosing between several close scores (e.g., between 86 and 87). To solve this issue, ratings are typically conducted on a 5-point or 7-point Likert scale. The scale alone is insufficient, as there is no standard agreement on the meaning of the numbers on these scales. One student may perceive a rating of 5 as another perceives a 4. To reach a common understanding, some calibration or elimination of bias must be performed (e.g., [34, 40, 38]). Thus, in this research, a grading language [4] or scoring rubric [2] is employed on top of the scale. Specifically, "Outstanding" is mapped to the grade 5, and "Very Good" to 4.

(b) Strategization. Students have been reported to employ subjective underscoring as a strategy, particularly under rivalry and competitive circumstances [5, 25]. Consider, for example, three student projects: c_1, c_2, and c_3. An assessor who favors c_1 will assign it the highest possible score, but to ensure that c_1 ends up in the first place, the assessor might also assign c_2 and c_3 the lowest possible score. A partial solution adopted in this research is to compute the median rating instead of the average rating, as it is not hard to see that the median is more immune to this sort of manipulation.

(c) Generosity. Students struggle to be critical towards their peers [22] and often assign generous grades, while teachers are more strict [28, 32]. This student tendency to over-mark has been seen to occur when students have no anonymity since they do not want to be seen as penalizing their peers [47]. We found this also occurs in anonymous settings; Figure 1 illustrates the grades that 27 students gave to 10 projects their peers presented in class. The median score for all projects on a rubric of five was either "Outstanding" or "Very Good"; none of the projects received a median grade of "Good" or lower. This is not an irregularity. Peer grading data from ten different class sessions all exhibited similar performances (see Appendix [1]). This does not mean that students think all projects are equally outstanding. As we show later, when prompted, students are indeed able to state which projects they prefer over others.

To address these challenges, some peer assessment systems have suggested collecting ordinal preferences [36] or assigning distinct scores to each project (effectively translating into ranking the alternatives). However, experts and non-experts alike are limited in the number of preferences they can meaningfully order — sometimes, they simply do not hold a rank over the alternatives (see, e.g. Chapter 15 in Balinski and Laraki [4]), and oftentimes they find it hard to rank more than seven items from best to worst [26]. Pairwise comparison queries such as "Do you prefer this alternative over that alternative?" may assist people in forming a ranking [12, 23]. However, this can be perceived as a tiresome task for voters, as the required number of comparison queries is large (e.g., 45 queries for ten projects, 105 queries for 15 projects). In summary, limiting students to rating projects results in the issues detailed above, while limiting students to ranking alternatives requires a cognitive and communication effort that may burden them.

Contributions: We study peer assessment from an algorithmic perspective. We integrate both cardinal preferences, expressed through ratings or grades, and ordinal preferences, expressed through pairwise comparisons. Our contribution is a novel hybrid peer assessment model, $R2R$, comprising (a) an algorithm that elicits student ratings and conducts pairwise comparison queries as needed and (b) a method for combining grades to create one overall ranking. The class instructor can then utilize this ranking to evaluate the projects and select the top projects in the class. The full paper was presented at ECAI 2024 [10].

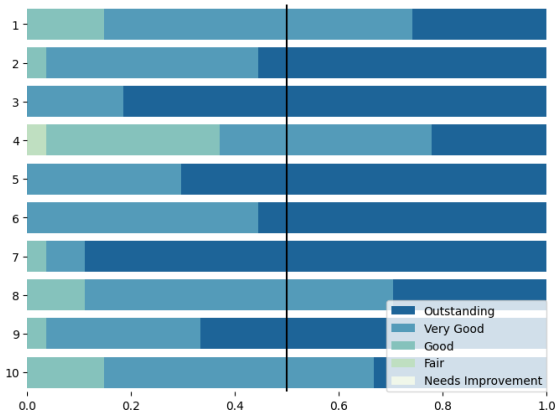


Figure 1: Grade distribution for ten projects. The y-axis lists project numbers; the x-axis shows the cumulative percentage of students assigning each grade (see legend). The vertical black line marks the median score.

2 Related work

The importance of peer assessment transcends mere grading by students and is an integral component of the learning process, commonly referred to as "assessment for learning" rather than "assessment

[1] https://doi.org/10.17605/OSF.IO/K4YMA

of learning" [46]. There exists a large body of literature on the benefits of peer assessment (e.g., [1, 6, 7, 13, 20, 21, 37, 50]).

Ranking-based systems have better reliability [39], and assessors using these systems have been found to be 10% more reliable than rating-based systems [41]. Some systems allocate varying weights to reviewers based on their credibility (e.g., [43]). Others implement controls on who reviews whom or incorporate AI-based methods [9]. Walsh [48] proposes a peer-rank rule in which the grade a user assigns to another user is weighted by the assigning user's own grade. The user's grade is the weighted average of the grades of the users evaluating that user. This approach shares similarities with the PageRank algorithm [31]. However, a prevailing perspective in peer assessment is that all assessors (students) should be considered equally credible. Consequently, most peer assessment systems aggregate ratings using either the average or the median rating (e.g., [33, 35, 49]).

Simply computing the average rating is not a good solution, as it is easily manipulable [42]. This is especially true in peer assessment systems, as the users of the system are also the candidates (or friends of the candidates) and may attempt to strategize, as explained in the previous section. While averaging is commonly used in sports performance evaluations, entities like the Olympic Committee and international federations often eliminate the highest and lowest ratings received by a candidate. In the specific context of sports performance evaluation, Osório [30] introduces a bias correction procedure that addresses deviations from the mean score and individual users' grading patterns. However, these methods do not perform well in the context of peer assessment, because there is typically little variance in peer assessment ratings, and the evaluators are often students without an extensive voting history.

Computing the median rating offers a less manipulable solution that works in peer assessment context, however, it can lead to numerous ties between alternatives. The Majority judgment voting protocol [4] addresses the issue of tie-breaking the median, with recent methods proposed for this purpose by Fabre [14]. Nevertheless, these methods still result in ties when applied to student peer rating grades. In this research, the median is initially computed, but to overcome the limitations associated with ties, rankings are also integrated into the assessment process.

Peer assessment has extended to applications that involve a large number of items, where obtaining evaluations from each user on every item becomes impractical. These applications include massive online courses (MOOCs) [19, 34], freelancing platforms [18], and academic conference reviews([42]). However, the primary algorithmic focus of these applications lies in determining who reviews what and in aggregating incomplete sets of reviews. In our case, we assume that all students evaluate all projects, resulting in complete preference sets.

Herein, we assume that all students are equal and all students rate all projects. Possible bias in evaluations is mitigated by employing a common grading language as done by Balinski and Laraki [4] and by using ordinal as well as cardinal evaluations.

2.1 Median-based voting rules

We provide a short summary of state-of-the-art median-based voting methods, as our model builds upon them.

Let there be n voters (students) $V = \{v_1, v_2, \ldots, v_n\}$ and m candidates (projects) $C = \{c_1, c_2, \ldots, c_m\}$. Let s_i^j denote the score assigned by voter v_i to candidate c_j. The score is assigned from a predefined domain of discrete values $D = \{d_{min} \ldots d_{max}\}$ where d_{min} and d_{max} are the lowest and highest values respectively $(d_{min} \leq s_i^j \leq d_{max})$. The ordered set of all scores assigned to candidate c_j is S^j. The median of S^j is the score in the middle-most position when n is odd, and the score in position $n/2$ when n is even. As illustrated in Figure 1, the median score of most projects is "Outstanding" or "Very Good" (these scores translate into the numerical scores of "5" and "4"). Therefore, projects cannot be ranked solely according to the median, and a tie-breaking mechanism is required.

A few mechanisms exist for tie-breaking the median between projects. They are all based on the same idea: first, compute the projects's median score. Then add a quantity based on what Fabre [14] terms as proponents (p) and opponents(q). The proponent p is the share of scores higher than the median α: $p_j = \sum_{i|s_i^j > \alpha} s_i^j$. The opponent q is the share of scores lower than the median α: $q_j = \sum_{i|s_i^j < \alpha} s_i^j$.

The mechanisms differ in their use of p and q: a candidate's score is the sum of its median grade α and a tie-breaking rule. Fabre [14] defines three rules: Typical judgment, Central judgment, and Usual judgment.

Definition 1 (Typical judgement). *Typical judgment is based on the difference between non-median groups:* $c^{\Delta} = \alpha + p - q$.

Definition 2 (Central judgement). *Central judgment is based on the relative share of proponents:* $c^{\sigma} = \alpha + 0.5 \cdot \frac{p-q}{p+q}$. *When* $p = q = 0$ *we set* $c^{\sigma} = \alpha$.

Definition 3 (Usual judgement). *Usual judgment is based on the normalized difference between non-median groups:* $c^{v} = \alpha + 0.5 \cdot \frac{p-q}{1-p-q}$.

Fabre [14] shows that majority judgment [4] can also be defined using just α, p and q :

Definition 4 (Majority judgement). $c^{mj} = \alpha + \mathbb{1}_{p>q} - \mathbb{1}_{p \leq q}$. [2] *If ties remain, the median of the tied candidates is dropped, and then* mj *is recomputed for the tied candidates. This procedure is repeated until all ties are resolved. Note that for ties Fabre [14] suggest an alternative method that results in the same output.*

Example 1. *Let us consider the set of scores assigned to candidate* c_j *by ten voters:*

$$S^j = \{5, 5, 5, 4, 4, 4, 3, 3, 3, 2\}$$

The calculated median for these scores is $\alpha = 4$. *Among the voters, three assigned scores above the median, while four assigned scores below the median. This leads us to define the proponent and opponent fractions as* $p = \frac{3}{10}$ *and* $q = \frac{4}{10}$, *respectively.*

For the candidate c_j, *the scores according to Typical, Central, Usual, and Majority judgments can be determined as follows:*

$$\begin{aligned} c_j^{\Delta} &= 4 + \frac{3}{10} - \frac{4}{10} = 3.9 \\ c_j^{\sigma} &= 4 + 0.5 \cdot \frac{-0.1}{0.7} \approx 3.92 \\ c_j^{v} &= 4 + 0.5 \cdot \frac{-0.1}{1.1} \approx 3.83 \\ c_j^{mj} &= 4 - 1 = 3 \end{aligned}$$

Project scores can be computed using the rules outlined above. A ranking of the projects can be obtained by simply ordering them according to their scores.

3 The model

Our proposed model has two key components: $R2R$ Elicitation and $R2R$ Aggregation. Together, these components create a hybrid model that incorporates both rankings and ratings.

[2] $\mathbb{1}_{f(x)}$ is an indicator function of $f(x)$. E.g., if $p > q$ then $\mathbb{1}_{p>q} = 1$

3.1 R2R elicitation

The $R2R$ Elicitation phase aims to gather voter preferences while minimizing cognitive and communication overload. Voters are requested to provide ratings for each candidate, with pairwise comparison queries employed only when necessary. We introduce the concept of a *Ranked Rating Set* to capture both scores and rankings.

Definition 5 (Rating Set). *A rating set S_i contains the scores that voter v_i assigned to candidates $c_1 \dots c_m$.*

Definition 6 (Preference profile). *A preference profile $\pi_i \in \Pi$ is a tuple $< c, p >$ containing candidates (c) and their ranked position (p) according to v_i. We write $c_i \succ c_j$ when c_i is preferred over c_j. For $k < l$ the preference profile is thus: $\{\dots (c_i, p_k), (c_j, p_l) \dots\}$.*

When v_i assigns a different score to each candidate, then the preference profile can be constructed directly from these scores. For example, when $S_i = \{s_i^1 = 4, s_i^2 = 5, s_i^3 = 3\}$ then the preference profile π_i is: $\{(c_2, 1), (c_1, 2), (c_3, 3)\}$ or equivalently $c_2 \succ c_1 \succ c_3$.

It is usually assumed that we can elicit the voter's rating set or the voter's preference profile. Here, we deviate from the standard literature and define a ranked rating set. A ranked rating set contains attributes of both a rating set and a preference profile, thus allowing us to save more information about voters' preferences.

Definition 7 (Ranked Rating Set). *A ranked rating set $\psi \in \Psi$ is a tuple $< c, s, p >$ containing candidates (c), their scores (s) and their position (p). As in the ranking set, the position is determined by the ranking. The rating set of voter v_i is denoted ψ_i.*

If v_i assigns the same score to two or more candidates, a pairwise comparison query $q(v_i, c_j, c_k) \in Q$ is executed. We assume that a voter can always respond to a query, i.e., a pairwise comparison query $q(v_i, c_j, c_k)$ has only two possible responses: $c_j \succ c_k$ or $c_k \succ c_j$.

We present a method, Rating to Ranking, $R2R$, which builds a ranked rating set while eliciting the needed information from the voters. A pseudo-code is provided in Algorithm 1. The algorithm receives as input a set of m alternatives, n voters, and a score domain D. Each voter is sequentially asked to provide scores for all candidates (lines 2-5). Note that this process can also be done the other way around: for each candidate, all voters must submit their scores. In either case, voter v_i assigns a score s_i^j to candidate c_j (line 5). Subsequently, the algorithm counts how many times this particular score (s_i^j) has been previously submitted by the voter (line 6). If the voter has not submitted this score before, it is inserted to ψ (lines 7-8) while ensuring its ordered placement (lines 7-8), utilizing the $insert$ function. If the score s_i^j already exists within ψ, the algorithm retrieves the index of the last position where it is found (line 10) and identifies the candidate c_p at that position (line 12). A pairwise query is then executed to determine the relative ranking between the two candidates (line 13). If c_j is ranked higher than c_p ($c_j \succ c_p$) the algorithm proceeds to the next position and repeats the query process (lines 14-16, followed by line 11). Otherwise, the algorithm proceeds to insert c_j at the identified position p (lines 19). Finally, the output is a ranked rating set (line 20).

It is important to note that this pairwise comparison approach prevents the elicitation of Condorcet cycles.

Example 2. *Consider one voter v_1 and four candidates: c_1, c_2, c_3, c_4. Scores are given in domain $D = 1, 2 \dots 5$. Assume that the voter's rating set is: $S_1 = \{s_1^1 = 5, s_1^2 = 4, s_1^3 = 5, s_1^4 = 5\}$ and the voter's preference profile is: $\pi_1 = \{c_3 \succ c_4 \succ c_1 \succ c_2\}$. The rating set and the preference profile are initially unknown. The goal is to determine the ranked rating set.*

At first, v_1 declares her score for c_1: $s_1^1 = 5$. Since this is the first score declared, the ranked rating set is updated to $\psi_1 = \{(c_1, 5, 0)\}$. Then, the next score is declared: $s_1^2 = 4$. Since ψ does not contain this

score, ψ_1 is updated without any queries: $\psi_1 = \{(c_1, 5, 0), (c_2, 4, 1)\}$. The next score declared is $s_1^3 = 5$. Since this score is already in ψ_1, a query is issued: $q(v_1, c_1, c_3)$. The voter responds by stating: $c_3 \succ c_1$. Thus c_3 is now added and ψ_1 positions of candidates is updated: $\psi_1 = \{(\mathbf{c_3}, \mathbf{5}, \mathbf{0}), (c_1, 5, \mathbf{1}), (c_2, 4, \mathbf{2})\}$ (changes to ψ_1 are marked in bold). Lastly, v_1 declares her score for c_4: $s_1^4 = 5$. This causes a query to be issued: $q(v_1, c_1, c_4)$. The voter responds with: $c_4 \succ c_1$, so yet another query is issued: $q(v_1, c_3, c_4)$. The voter responds with: $c_3 \succ c_4$ and ψ_1 is finalized: $\psi_1 = \{(c_3, 5, 0), (\mathbf{c_4}, \mathbf{5}, \mathbf{1}), (c_1, 5, \mathbf{2}), (c_2, 4, \mathbf{3})\}$ (again, changes to ψ_1 are marked in bold).

Algorithm 1 Rating to Ranking (R2R) elicitation

Input: A set of candidates $c_1, c_2, \ldots, c_m$; a set of voters $v_1, v_2, \ldots, v_n$; score domain $D = \{d_{\min}, \ldots, d_{\max}\}$
Output: For each voter v_i, a ranked rating set ψ_i
for $v_i \in V$ **do**
    $\psi_i \leftarrow []$
    for $c_j \in C$ **do**
        Voter declares $s_i^j \in D$
        $count \leftarrow count(\psi_i, s_i^j)$
        if $count = 0$ **then**
            $insert(\psi_i, c_j)$
        else
            $p \leftarrow lastIndex(\psi_i, s_i^j)$
            while $count \geq 1$ **do**
                $c_p \leftarrow candAtIndex(\psi_i, p)$
                Execute query $q(v_i, c_j, c_p)$
                if $c_j \succ c_p$ **then**
                    $p \leftarrow p - 1$
                    $count \leftarrow count - 1$
                else
                    break
                end if
            end while
            $insert(\psi_i, c_j, p)$
        end if
    end for
end for

3.2 R2R aggregation

The $R2R$ Aggregation phase combines voter preferences obtained through the $R2R$ Elicitation phase. Initially, the median α is calculated from the scores in the Ranked Rating Sets Ψ. The projects are grouped into buckets according to their medians. For all projects that share the same median, some voting rule is used to define their order. We herein employ two simple, well-known methods. Let $N(c_j, c_k)$ denote the number of voters who prefer $c_j \succ c_k$. In Borda voting, the candidate's score is based on its ranking in the voters' preference order. Formally:

Definition 8 ($Borda$ voting rule)**.** *The score of candidate c_j is the the total number of voters who prefer c_j over each other candidate c_k: $c_j^{borda} = \sum_{k=1, \forall k \neq j}^{m} N(c_j, c_k)$*

Copeland [8] suggested a voting system based on pairwise comparisons. The method finds a Condorcet winner when such a winner exists. Each candidate is compared to every other candidate. The candidate

obtains one point when it is preferred over another candidate by the majority of the voters. Adding up the points results is the Copeland score of each candidate. A family of Copeland methods, $Copeland^{\alpha}$, was suggested by Faliszewski et al. [15]. The difference is in tie-breaking the alternatives. The fraction of points each candidate receives in case of a tie between two candidates can be set to any rational number: $0 \leq \alpha \leq 1$. The Copeland score of a candidate is thus the number of points it obtained plus the number of ties times α. [3] Formally:

Definition 9 ($Copeland^{\alpha}$ voting rule)**.** *The score of candidate c_i is its sum of victories in pairwise comparisons:*

$c_j^{copeland^{\alpha}} = \sum_{k=1,\forall k \neq j}^{m} [N(c_j, c_k) > n/2] + \alpha \cdot \sum_{k=1,\forall k \neq j}^{m} [N(c_j, c_k) \equiv N(c_k, c_j)]$

$R2R\_b$ and $R2R\_c$ represent instances of $R2R$ that utilize Borda and Copeland voting, respectively, in the aggregation phase. While there are numerous ranking-based voting methods, we chose to begin with these methods since they are well-known and relatively easy to explain. Nonetheless, other voting methods are also applicable.

4 User Study - R2R System

We implemented a peer-rating system according to the $R2R$ model. [4]. Interested parties can utilize the system upon request. To initialize the system, project names and numbers were fed into the system beforehand by a system admin. A link to the system was then distributed to the students. Figures 2-4 present screenshots from the $R2R$ system interface as seen on a smartphone. Students are met with the main screen after logging into the $R2R$ system (Figure 2). This screen displays project numbers and names. Grey boxes indicate that evaluations for those projects have been successfully completed, while blue boxes signal that evaluations are still pending and can be submitted. Additional projects become visible as students scroll down the screen.

After each presentation, students were asked to use the system to assess the quality of the project. Upon selecting a box to initiate input, a project-specific evaluation screen emerges. Within this interface, students are presented with a set of five radio buttons, each corresponding to different evaluation levels. We adopted a grading scale consisting of five grades ($D = 1, 2 \ldots 5$), based on the research of Miller [26], which suggested that people can distinguish between seven plus or minus two levels of intensity. To minimize bias in the ratings, we utilized a standardized grading language proposed by Balinski and Laraki [4] that maps to the following five grades: Outstanding (5), Very Good (4), Good (3), Fair (2), and Needs improvement (1). Students were instructed to evaluate each project holistically and in accordance with the specified project requirements. A project graded as "Outstanding" met the requirements in an exceptional manner.

While an option for providing written feedback is available, it remained non-mandatory (Figure 3). In the event that a student assigned the same grade to more than one project, pairwise comparison queries were executed following Algorithm 1 (Figure 4). Finally, the system produced ranked rating sets, one set for each participating student.

4.1 Data collection

The system was deployed as part of a three-credit course on Data Analysis at Ariel University. Students presented their final projects in front of the class as part of the course. The other students present were instructed to input their evaluations into the $R2R$ system following each project presentation.

[3] The Copeland α differs from the median α. To maintain consistency with prior research and to aid readers familiar with the literature, we use the same symbol, α, for both quantities and specify which α is being referenced.

[4] The code is readily available at https://github.com/nlihin/R2R. The online system is at https://rate2rank-0d561bf6674a.herokuapp.com/

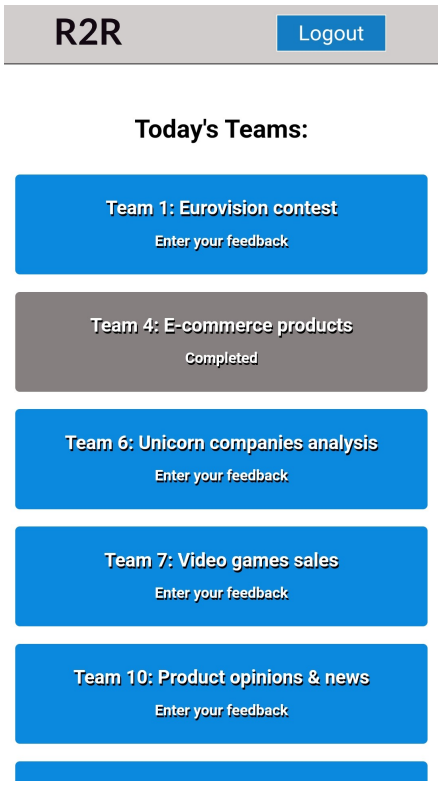


Figure 2: The main screen a student views after logging on to the $R2R$ system using a smartphone.

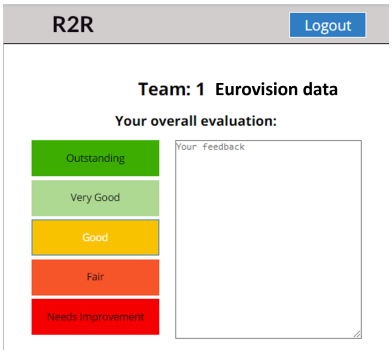


Figure 3: The evaluation screen of a specific project.

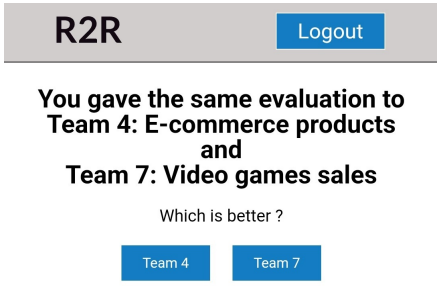


Figure 4: $R2R$ Pairwise comparison question displayed to a student after they assigned the same evaluation to two projects.

The full data collection statistics are presented in Table 1. Each student participated in one class presentation session. In each session, 8-12 projects were evaluated. We held 10 different sessions. The study received ethical approval from the institutional review board of the university. To maintain the participants' privacy, data was anonymized during the analysis, storage, and reporting stage, i.e., session dates, student identification numbers, and project names were concealed. In total, 288 students participated in the experiments. We removed 13 students who did not rate all the projects in their session. Another 31 students did not report their preferences iteratively, according to the project presentation order. These students were also removed, as they might have reported arbitrary preferences. Thus, the responses of a total of 244 students were included in the data analysis.

Most projects received a median grade of either "Outstanding" or "Very Good". For example, in Session I (the first row in Table 1), 27 students were asked to rate ten projects. Six projects received the median grade of "Outstanding", four received the median grade of "Very Good". All data and the Python notebook code written for the evaluation are available upon request.

5 Evaluation

The evaluation focused on two key aspects:

- **Methodological bias:** The project presentation order is predetermined. Students evaluate the projects iteratively after each presentation. While this is inherent to the problem setting and cannot be altered, it may influence student evaluations and, consequently, the final project rankings. We examined two potential biases: one related to grading and the other to comparing. In Section 5.1, we analyzed the presentation order bias, where students might assign higher (or lower) grades to the projects presented at the beginning. In Section 5.2, we investigated primacy

Table 1: Data collection statistics.

Session	Number of participants	Number of projects	Number of median grades (Outstanding, Very Good, Good, Fair, Needs improvement)
I	27	10	6,4,0,0,0
II	24	10	5,4,1,0,0
III	29	10	4,6,0,0,0
IV	22	9	2,7,0,0,0
V	27	10	3,5,2,0,0
VI	34	12	0,8,4,0,0
VII	18	8	1,4,3,0,0
VIII	21	9	3,6,0,0,0
IX	20	9	4,5,0,0,0
X	22	10	2,6,2,0,0

and recency bias, wherein students, when asked to choose between two projects to which they assigned the same grade, may preferentially select the first or latter project.

- **R2R effectiveness:** We compared the effectiveness of $R2R$ to other peer assessment methods. In Section 5.3, we quantified the reduction in communication load when using $R2R$ compared to methods requiring pairwise comparisons. In Section 5.4, we measured the extent to which $R2R$ reduces ties compared to other median-based voting rules and to the computation of the average. In Section 5.5, we compared the rankings produced by $R2R$ with those derived from other methods.

5.1 Presentation order bias

We explored the potential influence of presentation bias on project grading. We hypothesized that students might assign higher grades to the first projects presented. Subsequently, as more projects are presented and students engage in pairwise comparison prompts, they may distribute their scores more evenly, possibly as a strategy to circumvent the prompts. However, Figure 5 dispelled these notions. The y-axis is the percentage of voters that assigned a score of "Outstanding" (left figure) and "Very Good" (right figure). On the x-axis, projects are arranged according to their grading order, from the first project (numbered as project 1) to the last (either project 8, 9, 10, or 12, depending on the session). In the sole session featuring 12 projects, "Very Good" was assigned by few to the 12th project. However, this outcome may be attributed to the project's inherent qualities. We randomly shuffled project orders 100 times to compute an average "Outstanding" score distribution. Comparing it with the distribution in 5, a chi-square test could not reject the null hypothesis that both distributions are the same($p = 0.23$). We repeated this for "Very Good" scores, yielding similar results ($p = 0.26$).

5.2 Primacy and recency bias

Next, we examined whether users are prone to primacy or recency bias during pairwise comparisons. In other words, when prompted with a pairwise query, do users tend to select the first or last projects they see? To evaluate this, we analyzed instances where projects received the same score within a session. Subsequently, we computed the percentage of cases where projects were ranked in ascending or descending order. For example, if a student saw projects in this order of presentations: $c_1, c_2 \ldots c_{10}$, and rated projects c_2, c_3, c_8 as "Outstanding", and after answering pairwise comparisons, their preference profile is revealed as: $c_2 \succ c_3 \succ c_8$, it suggests a potential preference for the first projects seen — a

Table 2: Primacy and recency bias proportions.

Evaluation	Primacy bias	Recency bias
Outstanding	0.19	0.16
Very Good	0.17	0.1
Good	0.19	0.3
Fair	0.32	0.42
Needs improvement	0.14	0.43

primacy bias. Conversely, if a student rated projects c_1 and c_5 as "Very Good" and preferred c_5 to c_1, it suggested a preference for the last projects seen — a recency bias. Importantly, students might display a recency bias in some evaluations and a primacy bias in others. The results are reported in Table 2. In a simulation study, we randomly shuffled the ordering of the projects that received a similar grade from each student. This process was iterated 100 times, from which the average primacy and recency bias proportions were derived. Subsequently, we utilized a two-sided t-test to compare the simulated primacy proportions with those presented in Table 2. The test could not reject the null hypothesis that the proportion means are the same ($p = 0.39$). The same test was conducted for the recency bias, yielding similar results ($p = 0.91$).

5.3 Communication load

Table 3 summarizes the communication load findings. For 8,9,10, and 12 projects, the maximum pairwise queries are 28,36,45, and 66, respectively. In the ten sessions, the average number of issued pairwise queries lay in the range of 6.5-14.6 (the standard deviation is displayed in parentheses). Thus, $R2R$ reduces the communication load by $62\% - 77\%$. In other words, on average, students had to respond to less than 30% of the total number of possible pairwise queries.

Table 3: Communication load statistics

Session	Pairwise comparison queries	Communication reduction
I	13.9(5.7)	69%
II	10.8(2.7)	75%
III	13.2(5.0)	71%
IV	11.8(3.9)	67%
V	10.6(2.5)	75%
VI	14.6(5.9)	77%
VII	6.5(2.5)	76%
VIII	10.4(2.5)	71%
IX	13.5(6.8)	62%
X	10.6(3.1)	76%

5.4 Ties in rankings

We studied two instances of the proposed $R2R$ method: $R2R\_b$ and $R2R\_c$. They use Borda and Copland voting, respectively, in the aggregation phase (see Section 3.2). These instances were compared to existing methods: MAJORITY RANKING [4], TYPICAL RANKING, CENTRAL RANKING, and USUAL RANKING. The three latter are the rankings retrieved from the Typical judgment, Central judgment, and Usual

Table 4: Average (and standard deviation) of Kendall distance between different methods.

	R2R_b	R2R_c	Typical ranking	Central ranking	Usual ranking	Majority ranking
R2R_c	1.3 (0.4)					
Typical ranking	2.3 (2.1)	2.9 (1.5)				
Central ranking	3.4 (1.4)	4.0 (1.3)	2.4 (1.3)			
Usual ranking	2.4 (2.1)	2.9 (1.7)	0.6 (0.5)	3.1 (1.1)		
Majority ranking	2.8 (2.0)	2.9 (2.4)	1.3 (0.9)	3.9 (1.5)	0.8 (0.7)	
Mean	2.6 (2.1)	3.2 (1.4)	0.6 (0.4)	2.4 (1.2)	1.1 (0.7)	1.6 (1.1)

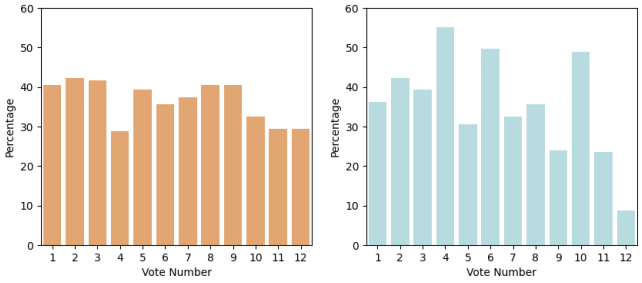


Figure 5: Percentage of voters that assigned a score of "Outstanding" (left) and "Very Good" (right).

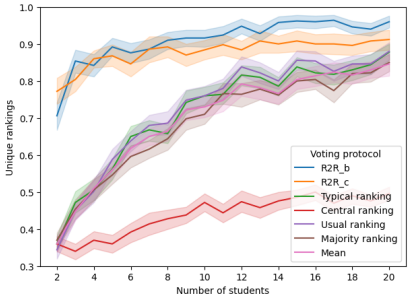


Figure 6: Ties in rankings in Session I.

judgment methods [14] (see Section 2). We also compared these methods to a simple MEAN method that computed the average rating of each alternative and ranked the alternatives accordingly.

Experiments were conducted with a varying number of voters (students in the user study), ranging from 2 to 20 (18 for Session VII), which were sampled without replacement. Each experiment ran 50 times. The α in $Copeland^{\alpha}$ was set to $\alpha = \frac{1}{3}$, as it provided better results than $\alpha = \{0, \frac{1}{2}, 1\}$ (setting α to values smaller than $\frac{1}{3}$ did not yield an improvement).

The unique number of rankings in each method was computed, which refers to the number of projects not tied in their order. Figure 6 compares all methods in Session I. Axis y is the fraction of unique rankings (with 1 meaning there are no ties in the final ranking). All sessions display a similar trend; as the number of voters increased, the methods exhibited better performance with fewer ties and more unique rankings. Note that even the MEAN method has ties in rankings. Employing $R2R\_b$ or $R2R\_c$ results in the fewest ties in the final ranking. In all sessions, $R2R\_b$ exhibited better performance than the other methods, especially when the number of voters was small. For an odd number of voters, $R2R\_c$ outperforms the other methods in some sessions. This advantage stems from the fact underlying the Copeland voting rule. Consequently, the observed increase in performance may not exhibit a smooth progression. Comparisons for the rest of the sessions demonstrate similar results (Appendix [5]).

To statistically compare the seven methods, we followed the robust non-parametric procedure described in [16]. The Friedman Aligned Ranks test with a confidence level of 95% rejected the null hypothesis that all seven methods perform the same. Further applying the Conover post hoc test supported our findings; $R2R\_b$ significantly outperformed all methods (except $R2R\_c$) with a confidence level of 95%. $R2R\_c$ significantly outperformed all methods but USUAL RANKING with a confidence level of 95%. No statistical difference was found between $R2R\_b$ and $R2R\_c$.

[5] https://doi.org/10.17605/OSF.IO/K4YMA

5.5 Consistency and tie-breaking impact

In line with the findings of Fabre [14], our investigation focused on assessing the degree to which different methods yield consistent outcomes while considering the impact of tie-breaking on these results. To quantify this impact, we calculated the Kendall-tau distance between the various rankings. The computed distances are presented in Table 4. A Kendall distance of 5 indicates that, on average, there is a 5% likelihood that the relative order of two items will be reversed. Our results align with those of Fabre [14], showing that in more than 95% of instances, the methods yield equivalent results. For the remaining cases, tie-breaking rules determine the outcome.

6 Discussion

$R2R$ originated from one of the authors' needs for a method to conduct peer assessment during oral presentations in the classroom, driven by the inadequacy of existing options. It provides a hybrid approach combining rating-based evaluations with pairwise comparison queries when necessary. $R2R$ offers a structured methodology for eliciting voter preferences through cardinal evaluations, such as scores or grades, while also accommodating ordinal preferences through pairwise comparisons. By aggregating these preferences, the $R2R$ method generates a ranked list of alternatives that reflects students' collective opinions. While the $R2R$ model may be considered naive in its structure and implementation, its effectiveness stems from its ease of understanding, use, and minimal assumptions, echoing the principles of Occam's razor. This simplicity enhances its potential for widespread adoption.

In a user study involving nearly 300 students, we did not identify any methodological bias in presentation order or primacy-recency bias. Furthermore, the effectiveness of the $R2R$ became evident through several key findings. Firstly, we found that $R2R$ outperformed existing methods by reducing the number of ties in the ranked output, particularly notable for small groups of voters, as often happens in in-class presentations. Moreover, $R2R$ imposes a lower communication load on students compared to pairwise comparison methods, as students only need to evaluate each alternative once instead of repeatedly comparing pairs of alternatives. The cognitive load is also diminished compared to models that necessitate students to rank all alternatives, as students only need to assign a score or grade to each alternative. One of the by-products of the user study is the acquisition of both project ratings and rankings, resulting in a unique dataset that may be utilized for further studies.

We presented two variations of the $R2R$ model, each employing a distinct tie-breaking method. The first variation utilizes the Borda voting rule for tie-breaking ($R2R\_b$), while the second variation employs the Copeland voting rule ($R2R\_c$). $R2R\_b$ excelled in its tie-breaking capabilities and exhibited greater consistency with the outcomes of all other methods. Thus, we recommend its use. It is also feasible to utilize other voting rules. Although the $R2R$ system was initially designed and implemented for a classroom setting, its potential applications extend beyond academia. In the workplace, for instance, group or team leaders often need to evaluate the performance of their employees, as noted by [3]. This is a sensitive task since an employee's position on the final performance list usually determines the bonus they will receive. Managers can use the $R2R$ to establish their personal ranked list of employees while minimizing communication and cognitive overload.

Limitations and future work. The current implementation uses a 5-point Likert scale. Expanding to a 7- or 9-point scale may reduce communication load but increase cognitive burden and bias—an open question for future research. The model also assumes truthful responses to pairwise queries, though strategic behavior remains a concern [27, 11] and was not addressed here. Additionally, students cannot express indifference, ensuring tie-free rankings but potentially distorting true preferences. Future work should assess the effects of relaxing this constraint and investigate how it influences both individual rankings and their aggregation. Another direction is to incorporate multiple types of pairwise comparisons, as proposed by Newman [29].

Acknowledgments

This work was supported by the Ministry of Innovation, Science and Technology, Israel (grant no. 0008125).

References

[1] Oluseyi Oluseun Adesina, Olufunbi Alaba Adesina, Ismail Adelopo, and Godfred Adjappong Afrifa. Managing group work: the impact of peer assessment on student engagement. *Accounting Education*, 32(1):90–113, 2023.

[2] Judith Arter and Jay McTighe. *Scoring rubrics in the classroom: Using performance criteria for assessing and improving student performance*. Corwin Press, 2001.

[3] Richard D Arvey and Kevin R Murphy. Performance evaluation in work settings. *Annual review of psychology*, 49(1):141–168, 1998.

[4] Michel Balinski and Rida Laraki. *Majority judgment: measuring, ranking, and electing*. MIT press, 2011.

[5] Bidyut Baruah, Tony Ward, and Noel Jackson. Should higher education encourage the use of intergroup peer assessment among students? In *2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–7. IEEE, 2018.

[6] Shao-Chen Chang, Ting-Chia Hsu, and Morris Siu-Yung Jong. Integration of the peer assessment approach with a virtual reality design system for learning earth science. *Computers & Education*, 146:103758, 2020.

[7] Shu-Yun Chien, Gwo-Jen Hwang, and Morris Siu-Yung Jong. Effects of peer assessment within the context of spherical video-based virtual reality on efl students' english-speaking performance and learning perceptions. *Computers & Education*, 146:103751, 2020.

[8] Arthur H Copeland. A reasonable social welfare function. In *Mimeographed notes from a Seminar on Applications of Mathematics to the Social Sciences, University of Michigan*, 1951.

[9] Ali Darvishi, Hassan Khosravi, Shazia Sadiq, and Dragan Gašević. Incorporating ai and learning analytics to build trustworthy peer assessment systems. *British Journal of Educational Technology*, 2022.

[10] Lihi Dery. Interactive and iterative peer assessment. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI 2024)*, volume 392, pages 1519–1526, Santiago de Compostela, Spain, October 2024. IOS Press. doi: 10.3233/FAIA240656. URL https://doi.org/10.3233/FAIA240656.

[11] Lihi Dery, Svetlana Obraztsova, Zinovi Rabinovich, and Meir Kalech. Lie on the fly: Strategic voting in an iterative preference elicitation process. *Group Decision and Negotiation*, 28:1077–1107, 2019.

[12] Lihi Naamani Dery, Meir Kalech, Lior Rokach, and Bracha Shapira. Reaching a joint decision with minimal elicitation of voter preferences. *Information Sciences*, 278:466–487, 2014.

[13] Kit S Double, Joshua A McGrane, and Therese N Hopfenbeck. The impact of peer assessment on academic performance: A meta-analysis of control group studies. *Educational Psychology Review*, 32:481–509, 2020.

[14] Adrien Fabre. Tie-breaking the highest median: alternatives to the majority judgment. *Social Choice and Welfare*, 56(1):101–124, 2021.

[15] Piotr Faliszewski, Edith Hemaspaandra, Lane A Hemaspaandra, and Jörg Rothe. Llull and copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[16] Salvador García, Alberto Fernández, Julián Luengo, and Francisco Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064, 2010.

[17] De Grez, Valcke, and Berings. Peer assessment of oral presentation skills. *Procedia - Social and*

Behavioral Sciences, 2(2):1776–1780, January 2010. ISSN 1877-0428. doi: 10.1016/j.sbspro.2010.03.983.

[18] Yasmine Kotturi, Anson Kahng, Ariel Procaccia, and Chinmay Kulkarni. Hirepeer: Impartial peer-assessed hiring at scale in expert crowdsourcing markets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(03), pages 2577–2584, 2020.

[19] Chinmay Kulkarni, Koh Pang Wei, Huy Le, Daniel Chia, Kathryn Papadopoulos, Justin Cheng, Daphne Koller, and Scott R Klemmer. Peer and self assessment in massive online classes. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 20(6):1–31, 2013.

[20] Hongli Li, Yao Xiong, Charles Vincent Hunter, Xiuyan Guo, and Rurik Tywoniw. Does peer assessment promote student learning? a meta-analysis. *Assessment & Evaluation in Higher Education*, 45(2):193–211, 2020.

[21] Hongli Li, Jacquelyn A Bialo, Yao Xiong, Charles Vincent Hunter, and Xiuyan Guo. The effect of peer assessment on non-cognitive outcomes: A meta-analysis. *Applied Measurement in Education*, 34(3):179–203, 2021.

[22] Sari Lindblom-Ylänne, Heikki Pihlajamäki, and Toomas Kotkas. Self-, peer-and teacher-assessment of student essays. *Active learning in higher education*, 7(1):51–62, 2006.

[23] Tyler Lu and Craig Boutilier. Learning mallows models with pairwise preferences. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 145–152, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

[24] Andrew Luxton-Reilly. A systematic review of tools that support peer assessment. *Computer Science Education*, 19(4):209–232, 2009.

[25] Hajo Meijer, Rink Hoekstra, Jasperina Brouwer, and Jan-Willem Strijbos. Unfolding collaborative learning assessment literacy: a reflection on current assessment methods in higher education. *Assessment & Evaluation in Higher Education*, 45(8):1222–1240, 2020.

[26] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[27] Lihi Naamani-Dery, Svetlana Obraztsova, Zinovi Rabinovich, and Meir Kalech. Lie on the fly: iterative voting center with manipulative voters. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 2033–2039. AAAI Press, 2015. ISBN 9781577357384.

[28] Ali Mansoori Nejad and Omer Hassan Ali Mahfoodh. Assessment of oral presentations: Effectiveness of self-, peer-, and teacher assessments. *International Journal of Instruction*, 12(3):615–632, 2019.

[29] Mark EJ Newman. Ranking with multiple types of pairwise comparisons. *Proceedings of the Royal Society A*, 2022.

[30] António Osório. Performance evaluation: subjectivity, bias and judgment style in sport. *Group Decision and Negotiation*, 29:655–678, 2020.

[31] Lawrence Page. The pagerank citation ranking: Bringing order to the web. technical report. *Stanford Digital Library Technologies Project, 1998*, 1998.

[32] Ernesto Panadero, Margarida Romero, and Jan-Willem Strijbos. The impact of a rubric and friendship on peer assessment: Effects on construct validity, performance, and perceptions of fairness and comfort. *Studies in Educational Evaluation*, 39(4):195–203, 2013.

[33] Dwayne E Paré and Steve Joordens. Peering into large lectures: examining peer and expert mark agreement using peerscholar, an online peer assessment tool. *Journal of Computer Assisted Learning*, 24(6):526–540, 2008.

[34] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned models of peer assessment in moocs. In *Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013)*, 2013.

[35] Helen Purchase and John Hamer. Peer-review in practice: eight years of aropä. *Assessment & Evaluation in Higher Education*, 43(7):1146–1165, 2018.

[36] Karthik Raman and Thorsten Joachims. Methods for ordinal peer grading. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages

1037–1046, 2014.

[37] Krishneel Reddy, Tony Harland, Rob Wass, and Nave Wald. Student peer review as a process of knowledge creation through dialogue. *Higher Education Research & Development*, 40(4):825–837, 2021.

[38] Juan Ramón Rico-Juan, Antonio-Javier Gallego, Jose J Valero-Mas, and Jorge Calvo-Zaragoza. Statistical semi-supervised system for grading multiple peer-reviewed open-ended works. *Computers & Education*, 126:264–282, 2018.

[39] Nihar B Shah, Joseph K Bradley, Abhay Parekh, Martin Wainwright, and Kannan Ramchandran. A case for ordinal peer-evaluation in moocs. In *NIPS workshop on data driven education*, volume 15, page 67, 2013.

[40] Yang Song, Zhewei Hu, Edward F Gehringer, Julia Morris, Jennifer Kidd, and Stacie Ringleb. Toward better training in peer assessment: does calibration help? 2016.

[41] Yang Song, Yifan Guo, and Edward F Gehringer. An exploratory study of reliability of ranking vs. rating in peer assessment. *International Journal of Educational and Pedagogical Sciences*, 11(10): 2405–2409, 2017.

[42] Ivan Stelmakh, Nihar B Shah, and Aarti Singh. Catch me if i can: Detecting strategic behaviour in peer assessment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4794–4802, 2021.

[43] Dapeng Tao, Jun Cheng, Zhengtao Yu, Kun Yue, and Lizhen Wang. Domain-weighted majority voting for crowdsourcing. *IEEE transactions on neural networks and learning systems*, 30(1):163–174, 2018.

[44] Keith Topping. Peer assessment between students in colleges and universities. *Review of educational Research*, 68(3):249–276, 1998.

[45] Keith J Topping. Peer assessment. *Theory into practice*, 48(1):20–27, 2009.

[46] Satu Tuomainen. Supportive assessment and feedback practices. In *Supporting Students through High-Quality Teaching: Inspiring Practices for University Teachers*, pages 77–94. Springer, 2023.

[47] Philip Vickerman. Student perspectives on formative peer assessment: an attempt to deepen learning? *Assessment & Evaluation in Higher Education*, 34(2):221–230, 2009.

[48] Toby Walsh. The peerrank method for peer assessment. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence*, pages 909–914, 2014.

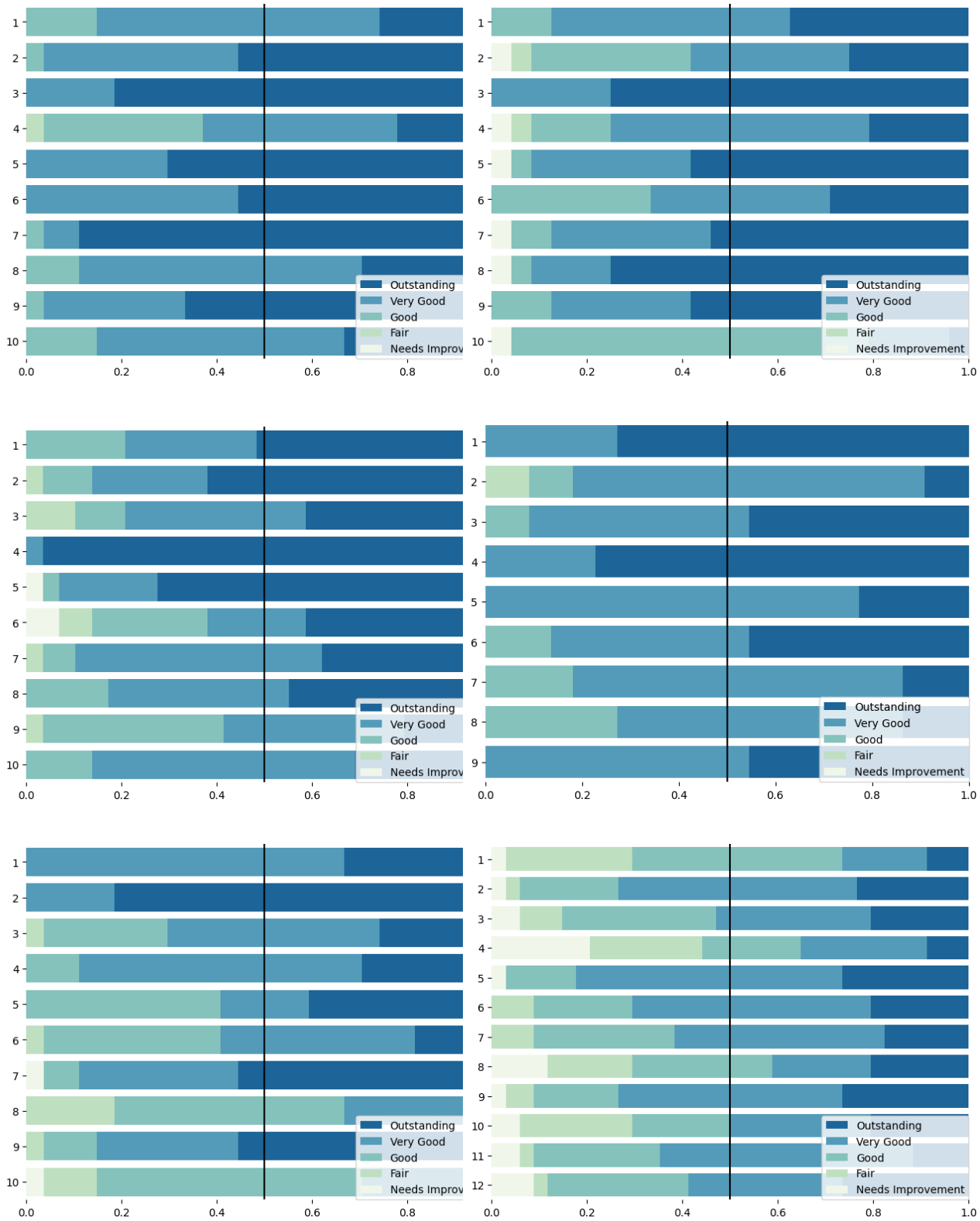
[49] David Kofoed Wind, Rasmus Malthe Jørgensen, and Simon Lind Hansen. Peer feedback with peergrade. In *ICEL 2018 13th International Conference on e-Learning*, page 184. Academic Conferences and publishing limited, 2018.

[50] Zi Yan, Hongling Lao, Ernesto Panadero, Belén Fernández-Castilla, Lan Yang, and Min Yang. Effects of self-assessment and peer-assessment interventions on academic performance: A pairwise and network meta-analysis. *Educational Research Review*, page 100484, 2022.

[51] Fu-Yun Yu, Yu-Hsin Liu, and Kristine Liu. Online peer-assessment quality control: A proactive measure, validation study, and sensitivity analysis. *Studies in Educational Evaluation*, 78:101279, September 2023. ISSN 0191491X.

Lihi Dery
Ariel University
Ariel, Israel
Email: lihid@ariel.ac.il

Appendix



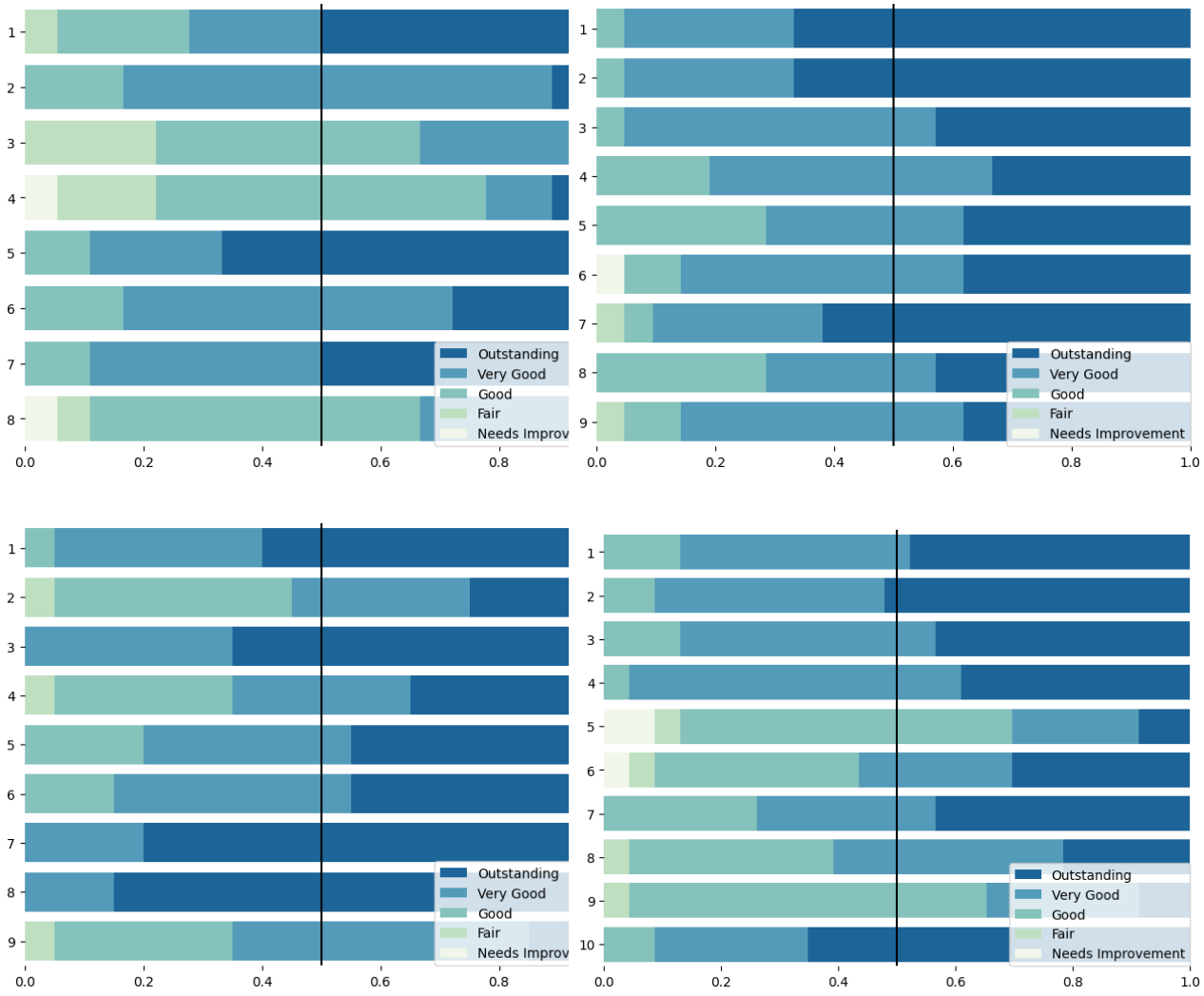
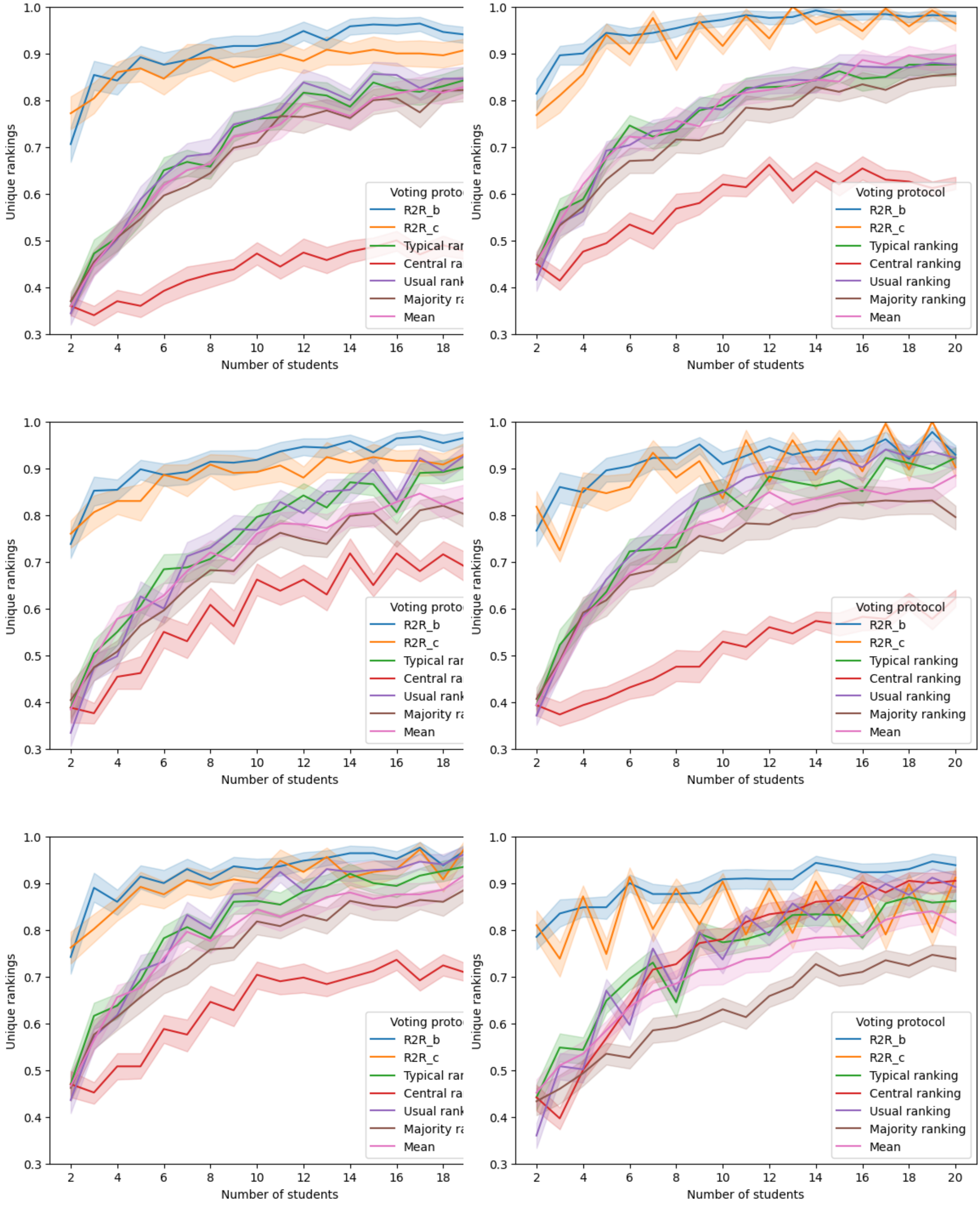


Figure 7: The grade distribution in Sessions I-X (from left to right, top to bottom).


Unique rankings
Number of students
Voting protocol
R2R_b
R2R_c
Typical ranking
Central ranking
Usual ranking
Majority ranking
Mean

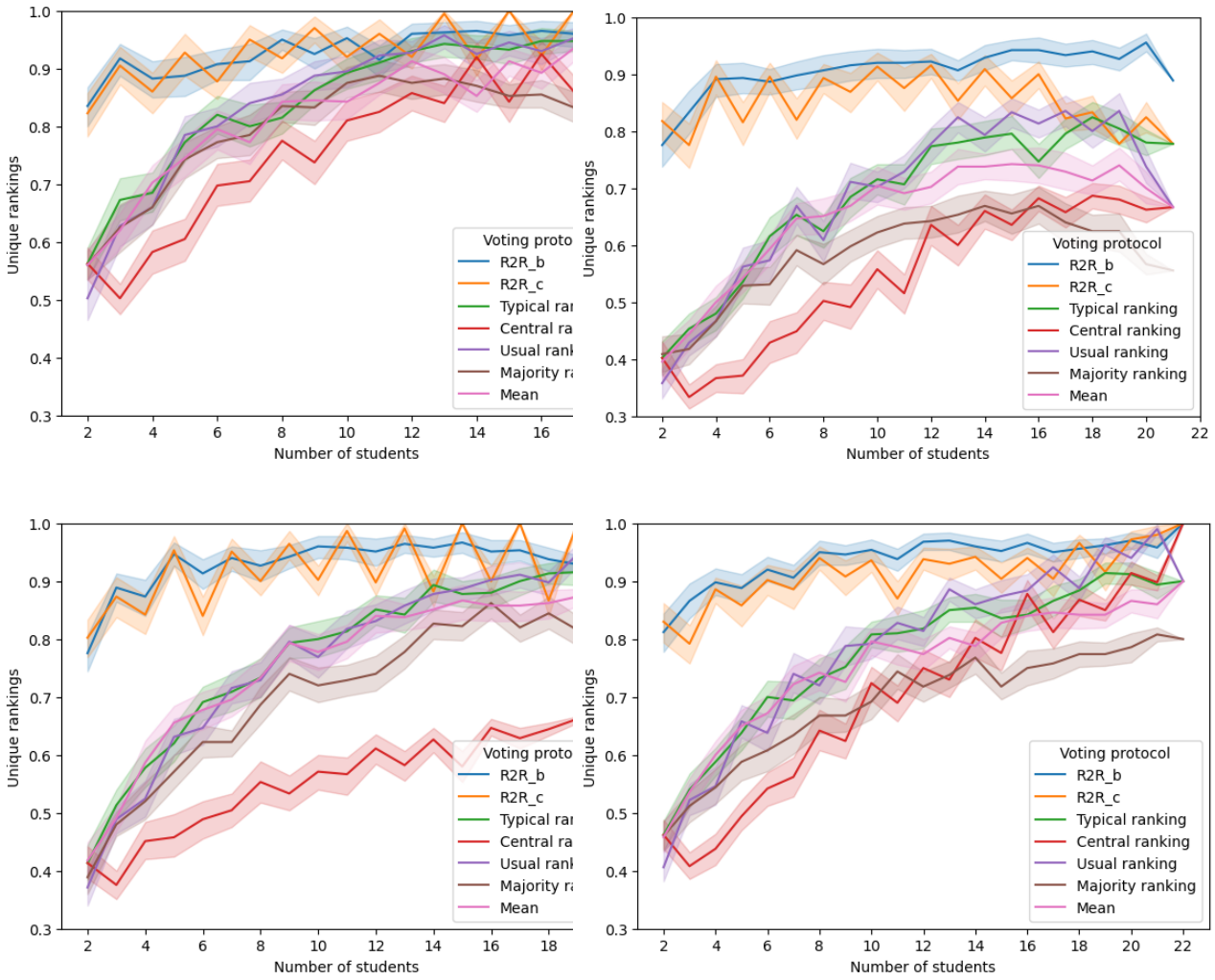


Figure 8: Unique rankings in Sessions I-X (from left to right, top to bottom).