NO MORE GRADING: INCENTIVE COMPATIBLE PEER ASSESSMENT IN A PROJECT-BASED COURSE

ETHAN LIGON

Abstract. We describe a system for grading collaborative student work that has been developed and refined over eight years of a project-based undergraduate course. Students complete four team projects per semester, with teams reassigned each round. After each project, every student evaluates other teams' deliverables, assesses their own teammates' contributions, and predicts the scores they themselves will receive. These three streams of assessment data are combined using singular value decomposition (SVD) applied to residualized evaluation matrices. Residualization removes evaluator-specific biases (the tendency to rate generously or harshly); SVD then extracts the dominant latent quality axis from the bias-corrected data, weighting questions by their informativeness. Scores are aggregated via medians for robustness to outlier evaluators. Each student's composite grade reflects five components: team quality, individual contribution as assessed by teammates, evaluator discrimination (rewarding careful assessment of others), and two measures of self-assessment accuracy (penalizing the gap between predicted and actual peer ratings). We map composite scores to letter grades using an anchor-point method that keys grade boundaries to the median performance of the top five students, spacing bands at one-third standard deviation intervals.

The system also feeds assessment information forward into team composition. Teams are formed via a stratified round-robin algorithm that spreads technical skill evenly across groups; for subsequent projects, self-reported skill is replaced by peer-assessed skill. A conflict-avoidance mechanism uses "would you work with this person again" responses to keep incompatible students apart. We present the mathematical foundations in enough detail for replication, discuss incentive properties (robustness to strategic manipulation, encouragement of honest and thoughtful evaluation), and illustrate the procedure with a worked numerical example.

Date: April 3, 2026.

Introduction

How should we grade students who work in teams? The question sounds simple, but it isn't. A team project produces a single deliverable, yet the students who built it contributed unequally, learned different things, and deserve different grades. The instructor observes the deliverable; teammates observe each other. Neither signal is sufficient on its own.

This paper describes a peer assessment system developed over eight years of offering EEP153, a project-based course in food and environmental economics at UC Berkeley.[1] The system asks students to evaluate both other teams' work and their own teammates' contributions, then combines these evaluations using singular value decomposition (SVD) to extract latent quality scores that are robust to evaluator bias. It also rewards accurate self-assessment, creating an incentive for honest reflection. And it feeds information forward: assessments from one project influence team composition for the next.

We present the method with enough mathematical detail that a reader could implement a similar system. The core ideas, however, are simple. Students know more about their teammates' contributions than the instructor does; we should use that information (Topping 1998; Falchikov and Goldfinch 2000). But students are noisy and sometimes strategic evaluators; we need to filter their reports intelligently.

Course Structure

Projects and Teams. The course is organized around four team projects, each lasting roughly three weeks. Projects are substantive: students build computational tools for population analysis, diet optimization, demand estimation, and food-system design. Each project culminates in a presentation and a public code repository.

For each project, students are assigned to new teams of roughly equal size. In a typical offering with 45–60 students, this means about 9 teams of 5–7 members. The system scales naturally to other class sizes; what matters is that the number of teams be large enough for cross-team evaluation to be informative (at least 5 or 6) and that teams be small enough for every member's contribution to be visible to teammates.

[1]The course was first offered in Spring 2019. The peer assessment and team formation system described here has evolved considerably since then; earlier iterations used simpler aggregation methods and less formal approaches to evaluator bias removal. We describe the system in its current mature form.

Reassigning teams each project serves two purposes. First, it gives students experience working with different collaborators, a skill valuable in itself. Second, it generates multiple independent observations of each student's collaborative ability: four different sets of teammates, each providing assessments. A student who free-rides once might get away with it; a student who free-rides four times will not.

Team Formation. Team composition isn't random. We use a stratified assignment algorithm designed to ensure that every team has at least one member with strong technical skills (Oakley et al. 2004; Layton et al. 2010). In a course that requires programming, this matters: a team of five beginners will struggle not because its members lack effort, but because none of them can teach the others.

Initial Stratification. For the first project, we measure skills using a pre-course survey that asks students to self-report their familiarity with Python and Git. Let s_i denote student i's skill score, computed as the mean of these two self-reports. We rank students by s_i in descending order, then assign them to teams in a round-robin cycle: the highest-skilled student goes to team 1, the next to team 2, and so on through all K teams, then back to team 1. Ties are broken randomly. The result is a set of teams in which high-skill students are spread roughly evenly.

Peer-Informed Stratification. For subsequent projects, we replace self-reported skill with peer-assessed skill. After each project, teammates identify each other's main strengths from a checklist that includes coding, leadership, presentation, organization, and other categories. Let $c_{ij}^{(p)}$ be an indicator equal to one if student j identifies student i as having coding skill in project p. We compute

$$r_i = \frac{1}{P} \sum_{p=1}^{P} \bar{c}_i^{(p)},$$

where $\bar{c}_i^{(p)}$ is the fraction of i's teammates who identified coding skill, and P is the number of completed projects. We include the initial survey score as $\bar{c}_i^{(0)}$ so that first-project information isn't discarded.

The ranking r_i replaces s_i in the round-robin assignment. This is important: it means the system *learns* who the skilled students are, rather than relying solely on self-reports.

Conflict Avoidance. Teammate assessments include a question: "Would you like to work with this person again?" Responses range from "Definitely" to "Definitely not." We extract all pairs (i, j) for which either i or j (or both) answered "Definitely not." Call this set $\mathcal{A}$, the animosity set.

During team assignment, when a student is about to be placed on a team, we check whether any current team member forms a conflict pair with that student (see Lappas, Liu, and Terzi 2009, for a formal treatment of constraint-based team formation). If so, we skip to the next team in the cycle and try again, up to $2K$ attempts. If all attempts fail (rare in practice), the student is placed on the smallest team with a warning.

This mechanism serves a pedagogical purpose beyond mere conflict resolution. It tells students that their assessments have consequences: if you report that you can't work with someone, the system will try to keep you apart. This encourages honest reporting and discourages strategic manipulation of the "work with again" question.

Assessment Instruments

After each project, every student completes three assessment forms.

Team Assessments. Each student evaluates every *other* team's project (not their own). The assessment covers four dimensions, each rated on a Likert scale: code quality, topic clarity, presentation quality, and organizational effectiveness. Students also provide an overall score on a 0–100 scale and open-ended comments.

Let y_{it}^{j} denote the score that evaluator i assigns to team t on question j. The full dataset is a three-way array indexed by evaluator, team, and question.[2]

Teammate Assessments. Each student evaluates every member of their *own* team, including themselves. The assessment covers four Likert-scaled questions (quality of work, reliability, helpfulness, and teamwork), a checklist of strengths, a relative ranking of all teammates' contributions (1st through nth), and a "work with again" rating.

Importantly, students also rate themselves on the same dimensions. These self-ratings aren't used directly in grading; instead, they serve

[2]The overall 0–100 score and the Likert items are measured on different scales. Before entering the SVD, we rank-transform the overall score so that it occupies the same ordinal scale as the Likert questions. This prevents a single high-variance item from dominating the first principal component.

as *predictions* of what others will say, and the accuracy of those predictions becomes a separate grade component.

Let z_{is}^j denote the score that evaluator i assigns to teammate s on question j. Self-evaluations ($i = s$) are separated from peer evaluations ($i \neq s$) before analysis.[3]

Overall Team Ranking. A third form asks each student to score every team (including their own) on an overall 0–100 scale. For their own team, the instruction is to *predict the median score that other students will assign*. This prediction is evaluated separately for accuracy.

From Assessments to Scores

The Estimation Problem. The raw assessment data are noisy and biased. Some evaluators are generous; others are harsh. Some discriminate carefully among teams; others give everyone the same score. We want to extract a latent quality signal from this mess.

The problem is structurally identical to a classic psychometric challenge: given a matrix of test responses, estimate latent ability (Lord and Novick 1968). Our approach combines two ideas. First, we residualize the data to remove evaluator-specific biases, an idea with antecedents in many-facet Rasch measurement (Linacre 1994; Myford and Wolfe 2003). Second, we apply SVD to the residuals to extract the dominant quality axis.

Residualization. Consider the team assessment data. We stack observations into a matrix with rows indexed by (evaluator, team) pairs and columns indexed by questions. The model is

$$y_{it}^j = \alpha_i + \beta^j + \varepsilon_{it}^j,$$

where α_i is an evaluator fixed effect (capturing the tendency of evaluator i to rate high or low across all teams and questions) and β^j is a question fixed effect (capturing the fact that some questions elicit systematically higher responses).

We remove these effects by regressing each column of the data matrix on a saturated set of evaluator dummies. Let D be the indicator matrix for evaluators. The residualized data are

$$\hat{\varepsilon} = Y - D(D'D)^{-1}D'Y.$$

[3]Some students misinterpret the ranking question, assigning "1st" to the teammate who contributed *least* rather than most. We detect this by computing the Spearman rank correlation between each evaluator's "Overall contribution" ranking and their other responses. If the median correlation is positive (indicating that higher-ranked teammates received *lower* Likert scores), we invert that evaluator's rankings before proceeding.

Because D is saturated (one dummy per evaluator, spanning the constant), this simultaneously removes both α_i and the column means β^j.

The residuals $\hat{\varepsilon}^j_{it}$ capture how evaluator i's rating of team t on question j deviates from what we'd expect given i's overall rating tendencies. A positive residual means "this evaluator rated this team higher than their usual baseline on this question."

Singular Value Decomposition. We compute the SVD of the residual matrix:

$$\hat{\varepsilon} = U\Sigma V'.$$

The first column of U, scaled by the first singular value σ_1, gives each (evaluator, team) pair a score on the dominant latent axis:

$$\text{pc1}_{it} = u_{it,1} \cdot \sigma_1.$$

The corresponding row of V' gives the loadings of each question on this axis. We orient the axis so that the loadings sum to a positive number, ensuring that higher scores mean better quality.

Why SVD rather than, say, simple averaging? Because SVD finds the linear combination of questions that explains the most variance in the residuals. If all questions measure the same underlying quality, the first principal component will load roughly equally on all of them. If some questions are more informative than others, SVD will up-weight those questions automatically. This is precisely the "simultaneous estimation of ability and question informativeness" that makes SVD powerful in psychometric applications (Eckart and Young 1936; Golub and Van Loan 2013).

Robust Aggregation. Each team receives multiple pc1_{it} scores, one from each evaluator. We aggregate to a single team score using the *median*:

$$\text{TeamPC1}_t = \text{median}_i\{\text{pc1}_{it}\}.$$

The median is robust to a single outlier evaluator. If one student assigns wildly atypical scores, the team's score is unaffected. This robustness is important in practice: with class sizes of 40–60, there are typically a few evaluators whose assessments are essentially noise.

Teammate Scores. The same residualize-then-SVD procedure applies to teammate assessments, with one modification: the observation unit is now (team, student, evaluator) rather than (evaluator, team).

Let $\hat{\varepsilon}^j_{is}$ denote the residual from teammate i's evaluation of student s on question j, after removing evaluator fixed effects. The SVD extracts a first principal component pc1_{is}, and we aggregate to the student level

by taking the median across evaluators within each (team, student) pair:

$$\text{MatePC1}_s = \text{median}_i\{\text{pc1}_{is}\}.$$

Evaluator Discrimination. Not all evaluators are equally informative. Some assign the same score to every team; others discriminate carefully. We measure evaluator quality as the standard deviation of an evaluator's PC1 scores across the teams they evaluated:

$$\text{EvalPC1}_i = \text{sd}_t\{\text{pc1}_{it}\}.$$

A high standard deviation means the evaluator's ratings co-vary with the consensus quality axis. A low standard deviation means the evaluator's ratings are essentially flat, contributing little information.

Including this as a grade component creates an incentive for careful evaluation: students who put thought into their assessments earn a (small) grade reward.

Prediction Accuracy. Recall that students predict their own scores. On the teammate form, each student's self-assessment serves as a prediction of the median peer rating they'll receive. On the overall form, a team's self-rating serves as a prediction of the median score from other teams.

Let $\hat{z}_s^j$ denote student s's self-prediction for question j, and let $\bar{z}_s^j$ denote the actual median peer rating. The prediction accuracy score is the negative mean squared error:

$$\text{MatePredMSE}_s = -\frac{1}{J}\sum_{j=1}^{J}(\hat{z}_s^j - \bar{z}_s^j)^2.$$

Higher (less negative) is better. An analogous quantity TeamPredMSE_s measures accuracy of team-level predictions.

Why reward prediction accuracy? Two reasons. First, it encourages self-awareness, a valuable metacognitive skill (Boud 1995; Nicol and Macfarlane-Dick 2006). The well-documented gap between self-assessed and actual competence (Kruger and Dunning 1999; Falchikov and Boud 1989) suggests that calibrating one's self-image is itself educationally valuable. Second, it discourages strategic manipulation of teammate ratings: if you inflate your self-assessment, your prediction error increases, costing you points. This incentive structure resembles the reviewer accuracy mechanism in CrowdGrader (de Alfaro and Shavlovsky 2014).

Composite Scores and Grade Assignment

Combining Components. Each student's grade for a given project is a weighted sum of five standardized components. Let X be the $n \times 5$ matrix of raw component scores, with columns for MatePC1, TeamPC1, EvalPC1, MatePredMSE, and TeamPredMSE. We first standardize each column to have mean zero and unit variance:

$$\tilde{X}_{\cdot k} = \frac{X_{\cdot k} - \bar{X}_k}{\mathrm{sd}(X_{\cdot k})}.$$

The composite score for student i is then

$$S_i = \sum_{k=1}^{5} w_k \, \tilde{X}_{ik},$$

with default weights approximately

$$w = (0.45,\ 0.45,\ 0.05,\ 0.025,\ 0.025).$$

Teammate and team quality each contribute roughly 45% of the grade; evaluator discrimination contributes 5%; and the two prediction accuracy components together contribute 5%.

These weights reflect a judgment about what matters. The dominant signals are: how good was your team's work, and how much did you contribute to it? The minor signals are: did you evaluate others carefully, and do you know how others see you? The weights are configurable and can be adjusted by project if warranted.

The Anchor-Point Method. We map composite scores to letter grades using a method anchored to the performance of the strongest students. The idea is to let the data tell us where the top of the distribution is, then space grade boundaries at regular intervals below it.

Let $\bar{x}$ denote the median composite score among the top five students (i.e., the third-highest score, ignoring ties). Grade boundaries are set at intervals of $1/3$ of a standard deviation below $\bar{x}$:

$$\begin{aligned}
\text{A+}: &\quad S_i > \bar{x} - \tfrac{1}{3} \\
\text{A}: &\quad \bar{x} - \tfrac{2}{3} < S_i \leq \bar{x} - \tfrac{1}{3} \\
\text{A-}: &\quad \bar{x} - 1 < S_i \leq \bar{x} - \tfrac{2}{3} \\
\text{B+}: &\quad \bar{x} - \tfrac{4}{3} < S_i \leq \bar{x} - 1 \\
&\ \ \vdots \\
\text{F}: &\quad S_i \leq \bar{x} - 4.
\end{aligned}$$

By construction, at least three students receive an A+ (since $\bar{x}$ is the third-highest score, and the A+ band extends 1/3 of a standard deviation below it). The remaining grade boundaries follow mechanically.

If scores are approximately normal, we can predict the grade distribution. With $\bar{x} \approx 1.517$ (the expected value of the third order statistic from a standard normal sample of size 40), the predicted distribution is:

Grade	Cutoff	Predicted %
A+	$S > \bar{x} - 1/3$	12.0%
A	$\bar{x} - 2/3 < S$	8.0%
A-	$\bar{x} - 1 < S$	10.6%
B+	$\bar{x} - 4/3 < S$	12.5%
B	$\bar{x} - 5/3 < S$	13.2%
B-	$\bar{x} - 2 < S$	12.6%
C+	$\bar{x} - 7/3 < S$	10.7%
C	$\bar{x} - 8/3 < S$	8.2%
C-	$\bar{x} - 3 < S$	5.6%
D/F	$S \leq \bar{x} - 3$	6.7%

These predicted percentages align reasonably well with the grade distribution reported for economics courses at Berkeley.

Small-Class Adjustments. The anchor-point method works well when the class is large enough for the top-five median to be a stable statistic. In smaller classes (say, fewer than 25 students), $\bar{x}$ becomes noisy, and the mechanical application of 1/3-standard-deviation intervals may produce implausible grade distributions. In such settings, we may need to make less principled adjustments to the cutoffs, informed by the instructor's overall assessment of student performance. The framework remains useful as a starting point even when it can't be applied mechanically.

Course Grades

Each project contributes equally to the course grade. With four projects, each accounts for roughly 22.5% of the total, or about 90% collectively. The remaining portion can accommodate other inputs: an optional final exam, contributions to class discussion on an online forum, or other measures of engagement. These additional components aren't peer-assessed; they provide an individual signal that complements the team-based assessments.

The system is deliberately modular. The SVD machinery operates independently on each project's assessment data, producing a vector of

component scores for each student. These project-level scores are standardized and weighted identically (or with project-specific adjustments if warranted). Adding a new input, such as a discussion participation score, requires only appending a column to the score matrix and adjusting the weight vector.

Properties and Incentives

Robustness to Strategic Behavior. Several features of the system discourage strategic manipulation:

- **Evaluator bias removal:** The residualization step removes any evaluator's tendency to rate systematically high or low. An evaluator who inflates all scores gains nothing; only *relative* assessments matter.
- **Median aggregation:** Taking medians rather than means makes scores robust to a single outlier evaluator. A coordinated attack would require a majority of evaluators to collude, which is unlikely in classes of this size.
- **Prediction accuracy:** Students who inflate their self-assessments incur a prediction error penalty. This creates a tension between wanting to look good and wanting to predict accurately, and the system rewards the latter.
- **Forward feedback:** The "work with again" responses feed into team formation for subsequent projects. A student who behaves badly isn't just graded poorly; they're also less likely to be placed with students who want to work with them.

Transparency. Students are told, at the beginning of the course, exactly how the system works. They know that evaluator biases are removed, that self-assessments are compared against peer assessments, and that "work with again" responses affect future teams. This transparency is itself an incentive: students who understand the system are more likely to provide thoughtful assessments, because they can see that strategic manipulation is difficult and that honest reporting is rewarded.

Limitations. The system assumes that the first principal component of the residual assessment matrix captures "quality." This is a strong assumption. If the dominant axis of variation is something other than quality (for example, evaluator familiarity with the topic), the scores will be misleading. In practice, the loadings on the first principal component tend to be roughly equal across questions, which is consistent

with a single quality factor. But we don't have a formal test of this assumption, and it's worth monitoring.

The prediction accuracy components are small (5% of the grade collectively), which limits their incentive power. We've kept them small deliberately: the evidence that prediction accuracy correlates with learning outcomes is suggestive but not conclusive, and we don't want to over-weight a noisy signal.

Finally, the system works best when teams are large enough for within-team assessments to be informative (at least 4–5 members) and when there are enough teams for cross-team assessments to be meaningful (at least 5–6 teams). Outside these ranges, the SVD may not have enough data to separate signal from noise.

Discussion

Pedagogical Rationale. The system is designed around a simple premise: students learn more from collaborative work when they take that collaboration seriously (D. W. Johnson and R. T. Johnson 2009; D. W. Johnson, R. T. Johnson, and Smith 2014). Peer assessment isn't just an input to grading; it's a learning activity (Black and Wiliam 1998; Sadler and Good 2006). Evaluating other teams' presentations requires students to think critically about what makes a project good. Assessing teammates requires reflection on what makes a collaborator effective. Predicting one's own scores requires self-awareness.

The team formation algorithm reinforces this. By stratifying on skill, we ensure that every team has someone who can teach and someone who can learn. By rotating teams across projects, we expose students to different working styles and skill sets. By avoiding conflict pairs, we remove the most destructive interpersonal dynamics while preserving the productive friction that comes from working with people you didn't choose.

Relation to Prior Work. Peer assessment has a long history in education research (Topping 1998; Dochy, Segers, and Sluijsmans 1999; Strijbos and Sluijsmans 2010), and we don't claim to have invented the idea. What we contribute is a specific implementation that combines several elements, including residualization for bias removal, SVD for latent quality estimation, and a feedback loop from assessments to team composition, into a coherent system with explicit mathematical foundations.

The use of SVD for grading is related to item response theory (IRT) in psychometrics (Lord and Novick 1968; Rasch 1960), which also models test responses as a function of latent ability and item difficulty. Our

approach is less parametric than standard IRT: we don't assume a specific functional form for the relationship between ability and responses. Instead, we let the SVD find the dominant axis of variation, which is a weaker but more flexible assumption. Recent work by Mignemi, Chen, and Moustaki (2025) develops a full latent variable model for peer grading networks that accounts for the dual role of students as both raters and examinees; their Bayesian approach offers a natural complement to the frequentist procedure described here.

Several algorithmic approaches to peer grading have emerged from the MOOC literature. Piech et al. (2013) model grader biases and reliabilities; Raman and Joachims (2014) develop ordinal methods; Walsh (2014) weight grades by grader quality in an approach analogous to PageRank. Kulkarni et al. (2013) report peer-staff correlations of $r = 0.73$ in a large online class. Our system operates at smaller scale but in a richer setting: we observe the same students across multiple projects, and we use assessment data not only for grading but for team formation.

Extensions. The framework admits several natural extensions. One could incorporate GitHub contribution data (commit frequency, lines of code, code review activity) as additional columns in the score matrix. One could weight projects unequally, giving more weight to later projects on the theory that students improve over the semester. One could use the prediction accuracy scores to identify students who would benefit from metacognitive coaching (Boud and Falchikov 1989).

The most ambitious extension would be to use the assessment data longitudinally, tracking how students' peer-assessed skills evolve across projects. This would require linking student identities across team assignments, which the current system supports through a panel identifier.

Conclusion

We've described a peer assessment system for project-based courses that extracts latent quality scores from noisy, biased evaluations using SVD, maps those scores to letter grades using a data-driven anchor-point method, and feeds assessment information forward into team composition. The system is transparent, reasonably robust to strategic manipulation, and designed to make the assessment process itself a learning experience.

The mathematical machinery, residualization followed by SVD followed by robust aggregation, is simple enough to implement in a few hundred lines of code. The pedagogical machinery, skill stratification,

conflict avoidance, and prediction accuracy rewards, requires more institutional infrastructure but no conceptual leaps.

We don't claim this system is optimal. We do claim it's principled, transparent, and that it takes peer assessment seriously as both a measurement tool and a learning activity. That seems like the right starting point.

Acknowledgments

The authors thank Claude (Anthropic) for assistance with manuscript preparation.

References

References

Black, Paul and Dylan Wiliam (1998). "Assessment and Classroom Learning". In: *Assessment in Education: Principles, Policy & Practice* 5.1, pp. 7–74. DOI: 10.1080/0969595980050102.

Boud, David (1995). *Enhancing Learning Through Self-Assessment*. London: Kogan Page. ISBN: 0-7494-1368-9.

Boud, David and Nancy Falchikov (1989). "Quantitative Studies of Student Self-Assessment in Higher Education: A Critical Analysis of Findings". In: *Higher Education* 18.5, pp. 529–549. DOI: 10.1007/BF00138746.

de Alfaro, Luca and Michael Shavlovsky (2014). "CrowdGrader: A Tool for Crowdsourcing the Evaluation of Homework Assignments". In: *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14)*. New York: ACM, pp. 415–420. DOI: 10.1145/2538862.2538900.

Dochy, Filip, Mien Segers, and Dominique Sluijsmans (1999). "The Use of Self-, Peer and Co-Assessment in Higher Education: A Review". In: *Studies in Higher Education* 24.3, pp. 331–350. DOI: 10.1080/03075079912331379935.

Eckart, Carl and Gale Young (1936). "The Approximation of One Matrix by Another of Lower Rank". In: *Psychometrika* 1.3, pp. 211–218. DOI: 10.1007/BF02288367.

Falchikov, Nancy and David Boud (1989). "Student Self-Assessment in Higher Education: A Meta-Analysis". In: *Review of Educational Research* 59.4, pp. 395–430. DOI: 10.3102/00346543059004395.

Falchikov, Nancy and Judy Goldfinch (2000). "Student Peer Assessment in Higher Education: A Meta-Analysis Comparing Peer and Teacher Marks". In: *Review of Educational Research* 70.3, pp. 287–322. DOI: 10.3102/00346543070003287.

Golub, Gene H. and Charles F. Van Loan (2013). *Matrix Computations*. 4th. Baltimore: Johns Hopkins University Press. ISBN: 978-1-4214-0794-4.

Johnson, David W. and Roger T. Johnson (2009). "An Educational Psychology Success Story: Social Interdependence Theory and Cooperative Learning". In: *Educational Researcher* 38.5, pp. 365–379. DOI: 10.3102/0013189X09339057.

Johnson, David W., Roger T. Johnson, and Karl A. Smith (2014). "Cooperative Learning: Improving University Instruction by Basing Practice on Validated Theory". In: *Journal on Excellence in College Teaching* 25.3–4, pp. 85–118.

Kruger, Justin and David Dunning (1999). "Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments". In: *Journal of Personality and Social Psychology* 77.6, pp. 1121–1134. DOI: 10.1037/0022-3514.77.6.1121.

Kulkarni, Chinmay et al. (2013). "Peer and Self Assessment in Massive Online Classes". In: *ACM Transactions on Computer-Human Interaction* 20.6, pp. 1–31. DOI: 10.1145/2505057.

Lappas, Theodoros, Kun Liu, and Evimaria Terzi (2009). "Finding a Team of Experts in Social Networks". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*. New York: ACM, pp. 467–476. DOI: 10.1145/1557019.1557074.

Layton, Richard A. et al. (2010). "Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria". In: *Advances in Engineering Education* 2.1, pp. 1–28.

Linacre, J. Michael (1994). *Many-Facet Rasch Measurement*. 2nd. Chicago: MESA Press.

Lord, Frederic M. and Melvin R. Novick (1968). *Statistical Theories of Mental Test Scores*. With contributions by Allan Birnbaum. Reading, MA: Addison-Wesley.

Mignemi, Giuseppe, Yunxiao Chen, and Irini Moustaki (2025). "Unfolding the Network of Peer Grades: A Latent Variable Approach". In: *Psychometrika* 90.3, pp. 1153–1174. DOI: 10.1017/psy.2025.10021.

Myford, Carol M. and Edward W. Wolfe (2003). "Detecting and Measuring Rater Effects Using Many-Facet Rasch Measurement: Part I". In: *Journal of Applied Measurement* 4.4, pp. 386–422.

Nicol, David J. and Debra Macfarlane-Dick (2006). "Formative Assessment and Self-Regulated Learning: A Model and Seven Principles

of Good Feedback Practice". In: *Studies in Higher Education* 31.2, pp. 199–218. DOI: 10.1080/03075070600572090.

Oakley, Barbara et al. (2004). "Turning Student Groups into Effective Teams". In: *Journal of Student Centered Learning* 2.1, pp. 9–34.

Piech, Chris et al. (2013). "Tuned Models of Peer Assessment in MOOCs". In: *Proceedings of the 6th International Conference on Educational Data Mining (EDM)*. arXiv:1307.2579. Memphis, TN.

Raman, Karthik and Thorsten Joachims (2014). "Methods for Ordinal Peer Grading". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. New York: ACM, pp. 1037–1046. DOI: 10.1145/2623330.2623654.

Rasch, Georg (1960). *Probabilistic Models for Some Intelligence and Attainment Tests*. Expanded edition, University of Chicago Press, 1980. Copenhagen: Danish Institute for Educational Research.

Sadler, Philip M. and Eddie Good (2006). "The Impact of Self- and Peer-Grading on Student Learning". In: *Educational Assessment* 11.1, pp. 1–31. DOI: 10.1207/s15326977ea1101_1.

Strijbos, Jan-Willem and Dominique Sluijsmans (2010). "Unravelling Peer Assessment: Methodological, Functional, and Conceptual Developments". In: *Learning and Instruction* 20.4, pp. 265–269. DOI: 10.1016/j.learninstruc.2009.08.002.

Topping, Keith (1998). "Peer Assessment Between Students in Colleges and Universities". In: *Review of Educational Research* 68.3, pp. 249–276. DOI: 10.3102/00346543068003249.

Walsh, Toby (2014). "The PeerRank Method for Peer Assessment". In: *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, pp. 909–914. DOI: 10.3233/978-1-61499-419-0-909.

Appendix: A Worked Example

To make the procedure concrete, consider a small example with three teams (A, B, C) and two assessment questions ($j = 1, 2$). Each team is evaluated by the members of the other two teams; for simplicity, suppose each team has a single evaluator, so evaluator a comes from team A, evaluator b from B, and evaluator c from C.

Raw Assessments. The raw scores y_{it}^{j} are:

	$j = 1$ (Code)	$j = 2$ (Presentation)
a on B	4	3
a on C	2	1
b on A	5	4
b on C	3	2
c on A	3	4
c on B	2	3

Evaluator b is generous (scores 5 and 4 for team A); evaluator c is harsh (scores 3 and 2 for team B). We want to remove these biases.

Residualization. The data matrix Y is 6×2. The evaluator dummy matrix D is 6×3 (one column per evaluator). The projection $D(D'D)^{-1}D'Y$ computes each evaluator's mean across all their assessments and subtracts it.

Evaluator means:

- a: $(3, 2)$ — mean of rows 1–2
- b: $(4, 3)$ — mean of rows 3–4
- c: $(2.5, 3.5)$ — mean of rows 5–6

The residuals are:

	$\hat{\varepsilon}^1$	$\hat{\varepsilon}^2$
a on B	1.0	1.0
a on C	-1.0	-1.0
b on A	1.0	1.0
b on C	-1.0	-1.0
c on A	0.5	0.5
c on B	-0.5	-0.5

Notice that the evaluator biases are gone. What remains is the relative signal: teams A and B are rated above their evaluator's baseline, while team C is rated below.

SVD. The residual matrix has rank 1 (the two columns are identical). The SVD gives:

$$\hat{\varepsilon} = U\Sigma V', \quad \sigma_1 = 3.0, \quad V_1' \approx (0.707,\ 0.707).$$

The loadings are equal on both questions, confirming that both questions measure the same underlying quality. The PC1 scores $u_{it,1} \cdot \sigma_1$ are:

	pc1_{it}
a on B	1.41
a on C	-1.41
b on A	1.41
b on C	-1.41
c on A	0.71
c on B	-0.71

Aggregation. Taking medians by team:

$$\text{TeamPC1}_A = \text{median}(1.41, 0.71) = 1.06, \quad \text{TeamPC1}_B = \text{median}(1.41, -0.71) = 0.35, \quad \text{TeamP}$$

The ranking is $A > B > C$. Despite evaluator b's generosity and evaluator c's harshness, the system recovers the same team ordering that every evaluator implicitly agrees on.

Evaluator discrimination scores:

$$\text{EvalPC1}_a = \text{sd}(1.41, -1.41) = 2.00, \quad \text{EvalPC1}_b = 2.00, \quad \text{EvalPC1}_c = 1.00.$$

Evaluators a and b discriminate sharply between teams; evaluator c less so. In this example, c's scores happen to be closer together, so c receives a lower evaluator discrimination score.

From Scores to Grades. Suppose the composite scores (after adding teammate and prediction components and standardizing) for five students are

$$S = (1.8,\ 1.4,\ 0.6,\ -0.3,\ -1.1).$$

The top-five median is $\bar{x} = 0.6$ (the third-highest score). Grade boundaries at 1/3-standard-deviation intervals below $\bar{x}$:

Grade	Boundary	Students
A+	$S > 0.6 - 0.33 = 0.27$	1, 2, 3
A	$S > 0.6 - 0.67 = -0.07$	
A-	$S > 0.6 - 1.00 = -0.40$	4
B+	$S > 0.6 - 1.33 = -0.73$	
B	$S > 0.6 - 1.67 = -1.07$	
B-	$S > 0.6 - 2.00 = -1.40$	5

Students 1, 2, and 3 all fall in the A+ band (by construction, the anchor ensures at least three do). Student 4 receives an A-. Student 5 receives a B-. No one falls below B- in this tiny class, which illustrates the small-class caveat: with only five students, the mechanical rule can't populate the full grade distribution.