Peer grading the peer reviews

A dual-role approach for lightening the scholarly paper review process

Arous, Ines; Yang, Jie; Khayati, Mourad; Cudre-Mauroux, Philippe

DOI
10.1145/3442381.3450088

Publication date
2021

Document Version
Final published version

Published in
The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021

Peer Grading the Peer Reviews: A Dual-Role Approach for Lightening the Scholarly Paper Review Process

Ines Arous
University of Fribourg
Fribourg, Switzerland
ines.arous@unifr.ch

Jie Yang
Delft University of Technology
Delft, Netherlands
j.yang-3@tudelft.nl

Mourad Khayati
University of Fribourg
Fribourg, Switzerland
mourad.khayati@unifr.ch

Philippe Cudré-Mauroux
University of Fribourg
Fribourg, Switzerland
pcm@unifr.ch

ABSTRACT

Scientific peer review is pivotal to maintain quality standards for academic publication. The effectiveness of the reviewing process is currently being challenged by the rapid increase of paper submissions in various conferences. Those venues need to recruit a large number of reviewers of different levels of expertise and background. The submitted reviews often do not meet the conformity standards of the conferences. Such a situation poses an ever-bigger burden on the meta-reviewers when trying to reach a final decision.

In this work, we propose a human-AI approach that estimates the conformity of reviews to the conference standards. Specifically, we ask peers to grade each other's reviews anonymously with respect to important criteria of review conformity such as sufficient justification and objectivity. We introduce a Bayesian framework that learns the conformity of reviews from both the peer grading process, historical reviews and decisions of a conference, while taking into account grading reliability. Our approach helps meta-reviewers easily identify reviews that require clarification and detect submissions requiring discussions while not inducing additional overhead from reviewers. Through a large-scale crowdsourced study where crowd workers are recruited as graders, we show that the proposed approach outperforms machine learning or review grades alone and that it can be easily integrated into existing peer review systems.

CCS CONCEPTS

• **Information systems → Crowdsourcing**; • **Mathematics of computing → Bayesian computation**; • **Computing methodologies → Neural networks**; **Learning latent representations**.

KEYWORDS

Peer grading, peer review, crowdsourcing, human-AI collaboration

ACM Reference Format:
Ines Arous, Jie Yang, Mourad Khayati, and Philippe Cudré-Mauroux. 2021. Peer Grading the Peer Reviews: A Dual-Role Approach for Lightening the Scholarly Paper Review Process. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3450088

1 INTRODUCTION

Peer review is the standard process of evaluating the scientific work of researchers submitted to academic journals or conferences. An essential task in this process comes at the end when the meta-reviewers have to make a decision as to accept a paper or not. Recently, peer review has been challenged by the rapid increase of paper submissions. Consider the example of computer science conferences: The Conference on Neural Information Processing Systems (NeurIPS) and the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) received 9467 and 6656 submissions in 2020, respectively; the numbers are five times the number of submissions they received in 2010.

To guarantee a minimum number of reviews per paper, those conferences recruit a large number of reviewers of different expertise levels and background. For example, due to the very high number of submissions, some conferences decided to lift the restriction of having published papers in former editions of the same venue to be part of the reviewing board [11]. The submitted reviews do not always meet the conformity standards of the conferences such as the presence of sufficient justification for the claims, the validity of argumentation (e.g., not self-contradictory), and the objectivity of comments. Such a situation poses an ever-bigger burden on the meta-reviewers, who not only have to handle more papers and reviews, but also have to carefully validate the reviews in terms of the conformity to the review standards. For instance, in the NeurIPS example we cite above, each meta-reviewer had to handle up to 19 submissions with around 76 reviews total.

The load could be reduced if we were able to develop methods to automatically detect low-conformity reviews. The need has been explicitly discussed recently by program chairs of the ACM SIGMOD conference [2]: "The chairs discovered low-confidence reviews manually; such reviews, however, should be flagged automatically to allow for immediate action", "automated analysis of the reviews as they come in to spot problematic text ... could dramatically alleviate the overhead that chairs and meta-reviewers endure while trying to detect the problem cases manually". We note that computational methods have provided strong support to streamline several parts of the peer review process, such as those for paper assignment to

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450088

reviewers [13, 21, 23, 27, 42], finding expert reviewers [12, 15, 30], and reviewer score calibration [4, 16, 31]; however, relatively little work can be found on developing computational methods for detecting low-conformity reviews.

Automatic detection of low-conformity reviews is nontrivial for two main reasons. First, the task is highly complex and requires to assess reviews from a multitude of dimensions [1, 18, 36, 38, 41] including justification, argumentation, objectivity, etc. Assessment on those dimensions is cognitively demanding as it requires to comprehend the review text to understand the various relations among its statements. Second, submission and review information of most conferences are not openly accessible for privacy and confidentiality concerns. This lack of training data limits the performance of existing natural language processing techniques.

To tackle these challenges, we advocate a human-AI collaborative approach for the semi-automatic detection of low-conformity reviews. We involve peer reviewers to grade each other's reviews anonymously with respect to important criteria of review conformity. Simultaneously, a machine learning model joins the assessment for less ambiguous reviews while learning from new peer grading to make connections between the review features and their conformity level. The main advantage of involving machine learning is that the model encapsulates and accumulates human knowledge of review conformity over time: what it learned in the previous editions of a conference can be used for a new edition by simply applying the model to new reviews. Over time, the model improves and the human-AI approach requires less amount of grading from humans to detect low-conformity reviews. The "peer grading peer reviews" mechanism does not disrupt current peer review process: reviewers of the same paper are supposed to read each other's reviews and make adjustments to their own reviews whenever necessary. Making such an explicit step by asking them to grade each other's reviews can potentially stimulate reviewers to be more engaged and promote the quality of the discussion thereafter. Our proposed mechanism is, therefore, a lightweight add-on to the current peer review systems without inducing much extra effort from the reviewers.

At the technical level, we introduce a Bayesian framework that seamlessly integrates machine learning with peer grading for assessing review conformity while allowing the model to learn from peer grading. An important consideration of our framework design is that it models the reliability of the graders, thus taking into account the effect of their various background and expertise levels. To learn the reliability and the parameters of the machine learning model, we derive a principled optimization algorithm based on variational inference. In particular, we derive efficient updating rules that allow both model parameters and grader reliability to be updated incrementally at each iteration. By doing so, both types of parameters can be efficiently learned with little extra computational cost compared to the computational cost for training a machine learning model alone.

To evaluate our proposed approach, we first conduct a small-scale online experiment with real expert reviewers, where we simulate the real peer review process with peer grading. We evaluate the effectiveness of peer grading by taking into account the grading as a weight of the reviewers' recommendation scores in the aggregation and we show that the aggregated score is a better approximation of the meta-decisions as compared to existing aggregation methods, e.g., average or weighted average by self-reported confidence. The number of expert grading is, however, not sufficient for evaluating proposed Bayesian framework. Inspired by the positive results of worker performance in judging the relevance of both scientific papers and search results to specific topics [8, 24], we conduct a larger-scale crowdsourcing study where we collect worker grading to approximate expert grading. We then use worker grading to evaluate our framework on the dataset we collected from the ICLR conference over a three-year time period, which allows us to observe the gradual model improvement over time.

In summary, we make the following key contributions:

- We propose a new dual-role mechanism called "peer grading peer reviews" to lighten the review process. Our approach can be easily integrated into current scholarly peer review systems;
- We introduce a Bayesian framework that integrates a machine learning model with peer grading to collaboratively assess the conformity of scholarly reviews while allowing the model to improve over time;
- We conduct a longitudinal evaluation of our framework across multiple years of a conference, showing that our method substantially improves the state of the art by 10.85% accuracy and that the model improves by 6.67% accuracy over three years.

2 RELATED WORK

In this section, we first discuss the state of the art in peer reviewing, then review existing work methodologically related to our framework in review assessment and peer grading.

2.1 Scientific Peer Review

In the following, we discuss two relevant topics: computational support for scientific peer review and biases in reviews. State-of-the-art tools from artificial intelligence are making inroads to automate parts of the peer-review process [37]. A typical example is automatic paper assignment to appropriate reviewers. The problem has been formulated as an information retrieval problem [13, 19, 22, 30], where a paper to be assigned is a "query" and each review is represented as a document (e.g., an expertise statement or publications of the reviewer). This problem has also been formulated as matching problem, where the goal is to match a set of papers with reviewers under a given set of constraints, like workload, interest, and conflicts-of-interest [20, 21, 23, 27, 42]. Another important topic is finding expert reviewers. The task generally relies on automatic content analysis of textual documents (e.g., academic publications) and scientometrics (e.g., number of grants and patents), as well as link analysis based on cross-references between documents [12, 15, 30]. Apart from those, work has also been devoted to developing methods for identifying sentiments in reviews [45] and for predicting rebuttal results [17]. Recently, a pre-trained language model SciBERT has been introduced for modeling text in scientific publications [6].

Compared to the large body of work on those problems, relatively little effort can be found on developing automatic tools for review conformity assessment. Recent discussions have pointed to problems in low-conformity reviews, where reviewers can exhibit bias or only support expected, simple results, or ask for unnecessary experiments [2, 3, 5, 7, 14, 37].

Among those problems, biases in reviews is the most extensively studied topic. An important source of review biases comes from the setup of the review process being single- or double-blind. Snodgrass [40] reviews over 600 pieces of literature on reviewing, summarizing the implications of single- and double-blind reviews on fairness, review quality, and efficacy of blinding. In particular, the author points out the significant amount of evidence showing review biases in a single-blind setup, favoring high-prestigious institutions and famous authors. A more recent study by Tomikins et al. [43] through a controlled experiment on the ACM WSDM conference confirms such a finding. Another important source of bias is varying standards of reviewers in providing recommendations. A recent analysis by Shah et al. [39] over the reviews of the Neurips conference finds that the fraction of papers receiving scores over a threshold is not aligned with the meaning of the threshold defined by the conference. For example, nearly 60% of scores were above 3 despite the fact that the reviewers were asked to give a score of 3+ only if the paper lies in the top 30% submissions. This leads to the frustration of many authors whose papers get rejected despite receiving good scores.

Compared to those studies on review biases, other aspects of low-conformity reviews are much less discussed such as the lack of justification for decisions and of arguments. We show in Section 4 through an online survey that the lack of justification for arguments and decisions is most often due to low-conformity reviews, which increases the complexity of the meta decisions and, if not handled well, lower the authors' trust in the venue. We envision that automatic methods for low-conformity reviews detection can significantly reduce this issue, similar to what automatic methods for paper-reviewer assignment achieved in the past decades. Our work makes a first attempt along this direction, providing a first-of-it-kind human-in-the-loop AI method that leverages both human and machine intelligence in determining review conformity.

2.2 Review Assessment and Peer Grading

In the design of our approach, we draw inspiration from existing methods for review assessment and peer grading, developed in different domains. Methods for review assessment have been mainly developed for e-commerce and online rating platforms. Olatunji et al. [33] propose a convolutional neural network with a context-aware encoding mechanism to predict the product reviews' helpfulness based on the review text. Zhang et al. [47] study the problem of predicting the helpfulness of answers to users' questions on specific product features. Their model is based on a dual attention mechanism to attend the important aspects in QA pairs and common opinions reflected in the reviews. These methods rely in their core on pre-trained language models such as Glove [35] or ALBERT [26]. These language models are trained on massive and heterogeneous corpora to capture text semantics, which provide useful information for review classification. Prediction for scholarly reviews is more challenging than for other types of reviews due to both the cognitive complexity of the task, the highly specialized topic, and the lack of available datasets for model training. Unlike those fully automatic methods, we consider the role of humans (i.e., peers) in our approach as indispensable, as we show in our experiments.

Methods for peer grading have been mainly developed for (online) education and crowdsourcing platforms. In the educational context, Wang et al. [46] study the phenomenon of students dividing up their time between their own homework and grading others from a game theory perspective. Crowd workers have been used to simulate the role of students and to assess homework quality. Mi et al. [29] propose a probabilistic graphical model to aggregate peer grading. Their method considers an online course setup and models both the student and the grader's reliability, imposing a probabilistic relationship between the reliability of a student and the true grade. Carbonara et al. [10] model the peer grading process in MOOCs as an audit game where students play the role of attackers and the course staff play defenders. In the context of crowdsourcing, Labutov et al. [25] propose a framework that fuses both task execution and grading. They adopt an Expectation Maximization algorithm to aggregate the grading by inferring both worker's reliability and task difficulty. From a methodological perspective, our framework is different from those aforementioned methods in that we take a human-AI approach that integrates peer grading and a supervised machine learning model, which is important for both improving the accuracy of review conformity and for reducing manual efforts.

3 THE PGPR FRAMEWORK

In this section, we introduce our proposed Bayesian PGPR framework that learns to predict the conformity of reviews from a few peer-graded reviews as well as from historical data (reviews and decisions) of a given venue. We first formally define our problem and then describe our overall framework, followed by our variational inference algorithm for learning PGPR parameters.

3.1 Notations and Problem Formulation

3.1.1 Notations. Throughout this paper, we use boldface lowercase letters to denote vectors and boldface uppercase letters to denote matrices. For an arbitrary matrix $\boldsymbol{M}$, we use $\boldsymbol{M}_{i,j}$ to denote the entry at the i-th row and j-th column. We use capital letters (e.g., $\mathcal{P}$) in calligraphic math font to denote sets and $|\mathcal{P}|$ to denote the cardinality of a set $\mathcal{P}$.

Table 1 summarizes the notations used throughout this paper. We denote the set of reviews with $\mathcal{I}$ and the set of graders as $\mathcal{G}$. We restrict $\mathcal{I}$ to include only the graded reviews without ground truth of conformity – our framework can be initialized with any number of reviews with ground truth, thereby utilizing historical data (see Section 3.4). For each review $i \in \mathcal{I}$, we extract a set of features as described in detail in Section 5.1.5 and denote the resulting vector by $\boldsymbol{x}_i$. We use $\boldsymbol{A}_{i,g}$ to denote the grade given by grader $g \in \mathcal{G}$ when reviewing $i \in \mathcal{I}$. Due to the fact that an individual grader can only grade a limited number of reviews, $\boldsymbol{A}$ is a sparse matrix where only a small proportion of the entries are known.

3.1.2 Problem Definition. Let $\mathcal{I}$ be the set of reviews, where each review $i \in \mathcal{I}$ is represented by a feature vector $\boldsymbol{x}_i$. Let $\boldsymbol{A}$ be the grader-review matrix where each element $\boldsymbol{A}_{i,g}$ is a grade given by a grader $g \in \mathcal{G}$ to a review i. Our goal is to infer the conformity score z_i for all reviews $i \in \mathcal{I}$ using $\boldsymbol{x}_i$ and $\boldsymbol{A}$.

Table 1: Notations.

Notation	Description
$\mathcal{I}$	Set of reviews
$\mathcal{G}$	Set of graders
$\boldsymbol{A}$	Grader-Review matrix
$\boldsymbol{x}_i$	Feature vector of a review
z_i	Review conformity distribution
r_g	Grader reliability distribution
b_g	Grader bias distribution
μ_i, σ_i	Parameters of the review conformity distribution
A_g, B_g	Parameters of the distribution of grader reliability
m_g, α_g	Parameters of the distribution of grader bias

3.2 PGPR as a Bayesian Model

PGPR is a unified Bayesian framework that integrates a machine learning model –modeling review conformity from features– with peer grading for predicting review conformity. Once trained, the machine learning part of PGPR can be used alone to predict conformity of reviews without peer grading.

The overall framework is depicted as a graphical model in Figure 1. It models review conformity from both the features (through the machine learning model) and peer grading, which is modeled as a process conditioned on the review conformity and grader properties (i.e., reliability and bias). In the following, we first describe how a machine learning model is embedded into PGPR and then describe the grading process and its integration into our framework.

3.2.1 Learning Conformity.

3.2.1 Learning Conformity. We model review conformity z_i with a Gaussian distribution:

$$z_i \sim \mathcal{N}(\mu_i, \sigma_i), \tag{1}$$

where μ_i and σ_i are the mean and the variance of the distribution, respectively. μ_i is predicted from the review features $\boldsymbol{x}_i$ through a neural network of arbitrary architecture.

$$\mu_i = \text{softmax}(f^{\mathcal{W}}(\boldsymbol{x}_i)), \tag{2}$$

where the function $f^{W}(\boldsymbol{x}_i)$ models the output of the network layers preceding the softmax layer, parameterized by $\mathcal{W}$ shared across all reviews. The variance σ_i of the Gaussian distribution is automatically learned through our inference algorithm (described in Section 3.3). Unlike normal supervised settings, we do not have the ground truth of review conformity μ_i; instead, we are given a set of review grades, which we model next.

3.2.2 Modeling Review Grades. We model the grading process by considering two important properties of graders, namely reliability and bias. In practice, we would like to have a measure of *confidence* in estimating the reliability and bias of the graders grading different numbers of reviews: we should be more confident in estimating the reliability and bias of graders who grade 50 reviews than those who grade 5 reviews only. To quantify the confidence in our inference, we adopt a Bayesian treatment when modeling both grader properties by introducing prior distributions.

Specifically, we denote the grader reliability by r_g ($g \in \mathcal{G}$) and model it with a Gamma distribution: a higher value indicates a better ability to provide accurate grades.

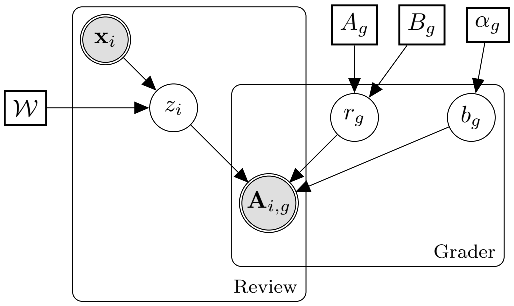


Figure 1: Graphical representation of PGPR. Double circles represent observed variables, while single circles represent latent variables. Squares represent model parameters. Edges represent conditional relationships in text classification. On the left-hand side, a machine learning model parameterized by $\mathcal{W}$ predicts the conformity z_i of a review. Each review is represented with a feature vector $\boldsymbol{x}_i$. On the right-hand side, a grader is represented with her reliability distribution r_g with parameters A_g and B_g and her bias b_g with α_g as a prior. The grader assigns a review with grade $\boldsymbol{A}_{i,g}$.

$$r_g \sim \Gamma(A, B), \tag{3}$$

We consider grader bias as the tendency of a grader to give high or low conformity scores to reviews. We denote the grader bias by b_g ($g \in \mathcal{G}$) and model it using a Gaussian distribution.

$$b_g \sim \mathcal{N}(m, \frac{1}{\alpha}). \tag{4}$$

3.2.3 Integrating Machine Learning with Peer Grading. We define the likelihood of a grader g giving a score $\boldsymbol{A}_{i,g}$ to review r as a probability conditioned on the grader's reliability r_g, the bias b_g, and the latent conformity of the review z_i.

$$p(A_{i,g}|z_i, r_g, b_g) = \mathcal{N}(z_i + b_g, \frac{1}{r_g}) \tag{5}$$

The conditional probability in Eq. (5) formalizes the following intuitions: *i)* a grader with a bias $b_g > 0$ (or $b_g < 0$) is likely to overestimate (or underestimate) the conformity of a review, whereas a grader with a bias $b_g \approx 0$ has a more accurate estimation of review conformity; and *ii)* a grader with a high reliability r_g is likely to give a conformity score with a small deviation from the true conformity.

3.3 Variational Inference for PGPR

Learning the parameters of PGPR resorts to maximizing the following likelihood function:

$$p(\boldsymbol{A}) = \int p(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}|\boldsymbol{X}; \mathcal{W})\, d\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}, \tag{6}$$

where $\boldsymbol{z}$ is the latent conformity scores for all the reviews, and $\boldsymbol{r}$ and $\boldsymbol{b}$ are the latent reliability scores and biases for all graders. $\boldsymbol{X}$ represents the feature matrix of all reviews and $\mathcal{W}$ is the set of machine learning parameters.

Since Eq. (6) contains more than one latent variable, it is computationally infeasible to optimize [44]. Therefore, we consider the log of the likelihood function, i.e.,

$$\log p(\boldsymbol{A}) = \underbrace{\int q(\boldsymbol{z},\boldsymbol{r},\boldsymbol{b}) \frac{p(\boldsymbol{A},\boldsymbol{z},\boldsymbol{r},\boldsymbol{b}|X;\mathcal{W})}{q(\boldsymbol{z},\boldsymbol{r},\boldsymbol{b})} d\boldsymbol{z},\boldsymbol{r},\boldsymbol{b}}_{\mathcal{L}(\mathcal{W},q)} + \underbrace{\int q(\boldsymbol{z},\boldsymbol{r},\boldsymbol{b}) \frac{q(\boldsymbol{z},\boldsymbol{r},\boldsymbol{b})}{p(\boldsymbol{z},\boldsymbol{r},\boldsymbol{b}|\boldsymbol{A},X;\mathcal{W})} d\boldsymbol{z},\boldsymbol{r},\boldsymbol{b},}_{KL(q||p)} \tag{7}$$

where $KL(\cdot)$ is the Kullback Leibler divergence between two distributions. The log likelihood function in Eq. (7) is composed of two terms. Using the variational expectation-maximization algorithm [44], we can optimize the objective function iteratively in two steps: 1) the E-step, where we minimize the KL-divergence to approximate $p(\boldsymbol{z},\boldsymbol{r},\boldsymbol{b}|\boldsymbol{A},X;\mathcal{W})$ with the variational distribution $q(\boldsymbol{z},\boldsymbol{r},\boldsymbol{b})$; and 2) the M-step, where we maximize the first term $\mathcal{L}(\mathcal{W},q)$ given the newly inferred latent variables. In the following, we describe both steps.

E-step. Using the mean-field variational inference approach [9], we assume that $q(\boldsymbol{z},\boldsymbol{r},\boldsymbol{b})$ factorizes over the latent variables:

$$q(\boldsymbol{z},\boldsymbol{r},\boldsymbol{b}) = \prod_{i\in\mathcal{I}} q(z_i) \prod_{g\in\mathcal{G}} q(r_g) \prod_{g\in\mathcal{G}} q(b_g). \tag{8}$$

To minimize the KL divergence, we choose the following forms for the factor functions:

$$q(z_i) = \mathcal{N}(\mu_i,\sigma_i), q(r_g) = \Gamma(A_g,B_g), q(b_g) = \mathcal{N}(m_g,\frac{1}{\alpha_g}), \tag{9}$$

where $\mu_i,\sigma_i,A_g,B_g,m_g,\alpha_g$ are variational parameters used to perform the optimization and minimize the KL-divergence.

In the following, we give the update rules for each of the latent variables. We first give the update rules for review conformity z_i by the following lemma.[1]

Lemma 3.1. *(Incremental Update for Review Conformity) The conformity distribution $q(z_i)$ follows a Gaussian distribution and can be incrementally computed using the grade, the grader reliability, and the review conformity from the previous iteration:*

$$q(z_i) \sim \mathcal{N}(\frac{W}{V},\frac{1}{V}), \tag{10}$$

where:

$$\begin{cases} W = \sum_g \frac{A_g}{B_g}(\boldsymbol{A}_{i,g} - m_g) + \frac{\mu_i}{\sigma_i^2}, \\ V = (\sum_g \frac{A_g}{B_g} + \frac{1}{\sigma_i^2}). \end{cases}$$

Next, we show the updating rules of grader's reliability and bias.

Lemma 3.2. *(Incremental Update for Grader Reliability) The update of the grader reliability $q(r_g)$ follows a Gamma distribution with parameters that can be incrementally updated using the conformity of reviews she graded, her bias and her reliability from the previous iteration:*

$$q(r_g) \sim Gamma(X,Y), \tag{11}$$

where:

$$\begin{cases} X = A_g + \frac{|\mathcal{I}_g|}{2}, \\ Y = B_g + \frac{1}{2}(\frac{|\mathcal{I}_g|}{\alpha_g} + \sum_i[\boldsymbol{A}_{i,g}^2 + \sigma_i^2 + 2\mu_i(m_g - \boldsymbol{A}_{i,g}) - 2\boldsymbol{A}_{i,g}m_g]). \end{cases}$$

Lemma 3.3. *(Incremental Update for Grader Bias) The bias of the graders $q(b_g)$ follows a Gaussian distribution with parameters that can be incrementally updated using the review conformity, the grader reliability and her bias from the previous iteration:*

$$q(b_g) \sim \mathcal{N}(\frac{L}{K},\frac{1}{K}), \tag{12}$$

where:

$$\begin{cases} K = \frac{A_g|\mathcal{I}_g|}{B_g} + \alpha_g, \\ L = \alpha_g m_g + \frac{A_g}{B_g}\sum_i(\boldsymbol{A}_{i,g} - \mu_i). \end{cases}$$

M-step. Given the conformity of a review, the grader reliability and bias inferred in the E-step, the M-step maximizes the first term of Eq. (7) to learn the parameter $\mathcal{W}$ of the machine learning model:

$$\begin{aligned} &\mathcal{L}(\mathcal{W},q) \\ &= \int q(z_i,r_g,b_g) \log p(\boldsymbol{A}_{i,g},z_i,r_g,b_g|\boldsymbol{x}_i;\mathcal{W})\, dz_i,r_g,b_g + C \\ &= \int q(z_i,r_g,b_g) \log[p(\boldsymbol{A}_{i,g}|z_i,r_g,b_g)p(z_i|\boldsymbol{x}_i;\mathcal{W})]\, dz_i,r_g,b_g + C \\ &= \underbrace{\int q(z_i,r_g,b_g) \log p(\boldsymbol{A}_{i,g}|z_i,r_g,b_g)\, dz_i,r_g,b_g}_{\mathcal{M}_1} \\ &+ \underbrace{\int q(z_i) \log p(z_i|\boldsymbol{x}_i;\mathcal{W}) dz_i}_{\mathcal{M}_2} + C \end{aligned} \tag{13}$$

where $C = \mathbb{E}_{q(z_i,r_g,b_g)} log(\frac{1}{q(z_i,r_g,b_g)})$ is a constant. Only the second part of $\mathcal{L}(\mathcal{W},q)$, i.e., $\mathcal{M}_2$, depends on the model's parameters. $\mathcal{M}_2$ is exactly the inverse of the cross-entropy between $q(z_i)$ and $p(z_i|\boldsymbol{x}_i;\mathcal{W})$, which is widely used as the loss function for many classifiers. $\mathcal{M}_2$ can, therefore, be optimized using back-propagation.

3.4 Algorithm

The overall optimization algorithm is given in Algorithm 1. We start by initializing the parameters of each probability distribution and of the machine learning model. Then, we iterate between the E step (rows 3-7) and the M step (rows 8-9). The E step consists of updating the variational distributions of the review conformity $q(z_i)$, the grader reliability $q(r_g)$ and her bias $q(b_g)$. The M step consists of updating the parameters $\mathcal{W}$ of the machine learning model using back-propagation. The convergence is reached when the review conformity $q(z_i)$ is no longer modified by the grader reliability and bias. Note that when some reviews with ground truth conformity are available, the machine learning model can be trained first to obtain an initialization of $\mathcal{W}$, which will then

[1] Proofs for all the lemmas are given in the appendix.

Algorithm 1: Learning PGPR Parameters

```
Input  : Grader-Review matrix $A$, Review features matrix $X$
Output : Parameters of the PGPR framework:
         $\mu_i, \sigma_i, A_g, B_g, m_g, \alpha_g, \mathcal{W}$
1 Initialize PGPR parameters ;
2 while $\log p(A)$ has not converged do
3     for $i \in \mathcal{I}$ do
4         update $q(z_i)$ using Lemma 3.1;
5     for $g \in \mathcal{G}$ do
6         update $q(r_g)$ using Lemma 3.2;
7         update $q(b_g)$ using Lemma 3.3;
8     for $i \in \mathcal{I}$ do
9         Update $\mathcal{W}$ using back-propagation;
```

be updated further by Algorithm 1. Once the learning algorithm terminates, the machine learning model of PGPR can be taken out to assess the conformity of any review.

The iterations in rows 3-4 require a time complexity of $|\mathcal{I}|$ and the iterations through all graders yield a time complexity of $|\mathcal{G}|$. The overall complexity of our algorithm is $O(\#iter(|\mathcal{I}|+|\mathcal{G}|+C_W)$ where $\#iter$ is the total number of iterations needed until convergence and C_W is the complexity to learn the parameters of the machine learning model.

4 TASK DESIGN FOR GRADING REVIEWS

In this section, we present our design for the review grading task, which is used to collect data for evaluating our proposed framework. Due to the privacy concern, submissions and review information in most venues are not publicly available. Fortunately, we have access to such an information in two venues, on which we conduct a small-scale experiment with expert reviewers to evaluate the effectiveness of peer grading in measuring review conformity. Evaluating our proposed PGPR framework, however, requires more grading than those we can collect from expert reviewers. We conduct a larger-scale crowdsourcing study, in which we collect worker grading to approximate the grading from expert reviewers and use those grading for evaluating PGPR.

This section focuses on the task design of grading reviews for both expert and crowd scenarios. We present an analysis on the effectiveness of grading from both expert reviewers and crowd workers in the next section. In the following, we first identify a set of criteria for review conformity assessment and then describe the setup of the grading task.

4.1 Criteria for Review Conformity

We compile a list of eight criteria for review conformity from the literature, a set of review guidelines published by journals and conferences [1, 18, 36], and guidelines from publishers such as Springer [41] or Nature Research [32]. Those criteria are grouped into the following three categories.

- **Clarity.** The clarity of a review resides in three main aspects. 1) *Structure*: it is often imposed that the review should contain a summary of the paper, the decision, and supporting arguments for this decision. 2) *Length*: a review should be of adequate length to provide sufficient information for the meta-reviewer to understand the reviewer's recommendation [34]. 3) *Justification*: a review should include supporting arguments of the decision, by including pointers to prior work as well as references to specific parts of the paper on which the score is based [18].

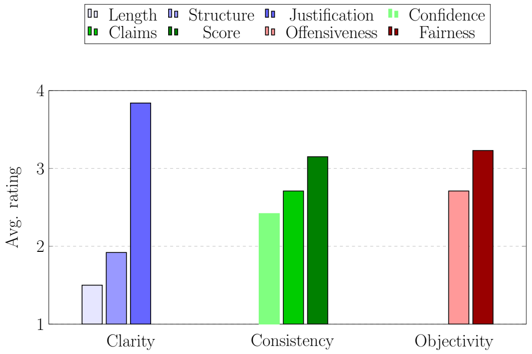


Figure 2: Ranking of review criteria.

- **Consistency.** The consistency of a review is defined by three aspects. 1) *Score*: the recommended score should be supported by at least one or two justifications. 2) *Claims*: there should be no contradiction between the summary and the stated weak or strong claims. 3) *Confidence*: the reviewer should make a clear acknowledgement when certain aspects of a paper are beyond her expertise [18].
- **Objectivity.** A review should be fair and provide constructive critiques. 1) *Fairness*: A review should not be biased towards irrelevant factors such as assigning a low score because of missing references from the reviewer's own work only[2]. 2) *Offensiveness*: A review should cover the technical work rather than giving personal statements and/or offensive terms [18].

To understand the importance of those criteria, we initially conducted an online survey with 38 expert reviewers from two international venues: SEMANTICS (SEM) (2019 edition) and the International Workshop on Decentralizing the Semantic Web (DSW) (2017 and 2018 editions). We asked the expert reviewers to rate the importance of each individual criterion and the three categories on a 5-point Likert scale and show the results in Figure 2. We observe that clarity ranked the highest (by 28 expert reviewers) and, in particular, that justification is viewed as the most important aspect of a high-conformity review. Consistency is equally important to objectivity. While many agree that objectivity is not a deterministic aspect, half of the experts admit having received or read unfair reviews while few have received offensive ones. In fact, 24 expert reviewers rank fairness as equally or more important than offensiveness. These results indicate that the experts consider review fairness as an important concern.

4.2 Task Design

For each of the reviews we consider in our work, we ask participants to provide ratings for each of the eight conformity criteria, grouped in three sections corresponding to the three categories introduced above. In the crowdsourcing scenario, we recruit from Amazon

[2] https://www.seas.upenn.edu/ nenkova/AreaChairsInstructions.pdf

MTurk workers with a "Master" qualification, i.e. workers who have demonstrated high degree of success in performing a wide range of tasks across a large number of requesters. The task starts by explaining how a scholarly review is presented, the criteria (on a category level), followed by a positive and a negative example. Then, we show workers a review and ask them to rate each criterion from 1 to 4 with 4 being the best rating. We set the range to be 1-4 instead of 1-5 as we found in a preliminary study that workers tend to favor 3 in the latter case. Each rating question is accompanied with an information box that explains the aspect to rate. For questions regarding justification, fairness and offensiveness, we ask workers to provide a snippet from the review as a rationale justifying their grading decision [28]. The rationale can be used as an explanation for the conformity score assigned to the review. For attention check, we ask workers to identify the recommendation decision from the review; results of workers who fail at recognizing review decisions are excluded. After getting their ratings, we ask the workers to enter feedback in free text. Each review is rated by three different workers. The task takes approximately 12 min to complete. Workers who completed the task received a reward of 1.8 USD.

In the expert scenario, the task is simplified to include only the rating for each of the criteria. The peer grading of scholarly reviews is implicit in the current peer review systems: each reviewer is supposed to read the reviews from other reviewers and decide whether to keep her original recommendation or not; however, they are typically not required to express their opinion about other reviews explicitly. We assume explicit peer grading can stimulate reviewers to look into other reviews and promote the quality of the discussions afterwards. We show in the next section through an experiment with real expert reviewers that the peer grading is effective when used to weight the reviewers' recommendation scores in score aggregation, which approximates meta-decisions better than existing aggregation methods, e.g., weighted average by reviewers' self-indicated confidence.

5 EXPERIMENTAL RESULTS

This section presents the results of our empirical evaluation[3]. We first conduct a preliminary analysis to understand the effectiveness of expert and worker grading, then evaluate the performance of our PGPR framework by comparing it against the state of the art. Finally, we perform an in-depth analysis of PGPR's main properties. We answer the following questions:

- Q1: How effective is expert and worker grading in assessing review conformity? (Section 5.2).
- Q2: How effective is our proposed human-AI approach in predicting review conformity? (Section 5.3).
- Q3: How effective is our framework in leveraging peer grading compared to majority voting? (Section 5.4).
- Q4: How effective is peer grading in improving the conformity prediction over time when more reviews with ground truth decisions become available? (Section 5.5).

5.1 Experimental Setup

5.1.1 Datasets. We collect data from the ICLR conference, which provides open access to reviews and evaluation scores for all submissions through OpenReview[4]. We collected reviews for all submissions to the ICLR conference from 2017 until 2019. Our ICLR dataset contains in total 2917 submissions and 8838 reviews. 1194 papers have at least one review that is misaligned with the meta-decision. In our study, we are mainly interested in those cases as they require some additional effort when reaching a final decision. Key statistics on the collected dataset are reported in Table 2.

Table 2: Description of the ICLR Datasets. #Misalign. sub. is the number of submissions to which there is at least a review with decision misaligned with meta-decision; #Misalign. reviews is the overall number of not-aligned reviews.

Edition	#sub.	#Misalign. sub.	#Misalign. reviews
2017	506	169	530
2018	846	355	1072
2019	1565	670	2060

5.1.2 Active Selection of Reviews for Grading. We leverage active learning to select a subset of the most informative reviews from the ICLR-2018 and 2019 datasets for grading: for each year, we apply the model trained in the previous year to all reviews in the current year, and select the reviews on which the model prediction is most uncertain (measured by the entropy of the predicted probability) for crowdsourcing. We select the top-30% (321) reviews and top-5% (103) reviews from ICLR-2018 and ICLR-2019, respectively, and show in our experiments that those numbers are sufficient for the model to converge to optimal performance. We refer to the selected reviews as "uncertain" reviews and the rest as "certain" ones. We investigate in our experiments the performance of PGPR on both categories as well as the impact of the number of graded uncertain reviews on model training. In total, we crowdsourced a subset of 444 reviews in 2018 and 2019 and collected 1093 grades from 64 crowd workers on those selected reviews.

5.1.3 Data Split. To simulate the real-world application of PGPR, we evaluate it on different editions of the ICLR conference as follows: for each year (2018 or 2019), we assume the reviews and the ground truth from previous years are known, while for the current year only the reviews are available without the ground truth. For a subset of the reviews in the current year, we collect grading from workers. The training data, therefore, contains reviews and decisions from the previous years, and some reviews with crowd labels from the current year. We take reviews with the ground truth of the current year and equally split it into validation and test sets.

5.1.4 Label Extraction. We consider the ground truth of a review conformity as a binary variable indicated by the alignment between a reviewer decision and the meta-reviewer decision: when both the reviewer and the meta-reviewer decide to accept or reject a paper, the ground truth for the review is set to 1, otherwise to 0. Our model predicts for each review a value between 0 and 1 describing the probability of the review being conform. The higher the value, the higher the likelihood of the review to be conform. For the grades collected from crowd workers, we map it to the interval $[0, 1]$ using the function $t(x) = (x - 1)/4$, so that the range of valid grading matches the range of our model's predictions.

[3] Source code and data are available at https://github.com/eXascaleInfolab/pgpr.

[4] https://openreview.net/

5.1.5 Neural Architecture and Features. The inputs of our machine learning model are hand-engineered features along with embeddings of the sentences in a review. For the hand-engineered features, we extract for each review the decision score, the confidence score, and their difference with the decision and confidence scores of the other reviews on the same paper. We also compute the review's length, the number of citations within the review, and the number of keywords referring to a paper's content (e.g., equation, section, figure). For the textual embeddings, we represent each sentence as a fixed-size vector by leveraging the pre-trained language model SciBERT [6]. These inputs are fed to the machine learning component of our framework consisting of a multi-input model we call "Mix-model". It includes both an attention-based model for the review's embedding and a logistic regression for the review's statistical features. We concatenate the output of the attention-based model and logistic regression and use a fully connected layer with tanh activation followed by a linear layer; the output is generated by a softmax function (Eq. 2).

5.1.6 Comparison Methods. We compare our approach against the most applicable techniques for review's conformity assessment. We first compare against classification methods designed for the scholarly domain: 1) MILNET [45], a Multiple Instance Learning (MIL) neural model used to classify scholarly reviews (originally for sentiment analysis). 2) SciBERT [6], a self-attention-based neural language model pre-trained on scientific text consisting of publications from the computer science and biomedical domains. 3) DoesMR [17], a Logistic Regression model that takes hand-engineered features from scholarly reviews for prediction. In addition, we compare against models developed for non-scholarly review tasks, including a general-purpose language model and two models originally developed for predicting the helpfulness of product reviews: 4) ALBERT [26], a pre-trained language model for various NLP tasks, taking into account inter-sentence coherence to capture fine-grained information in documents including reviews. 5) PCNN [33], a convolutional neural model with context encoding. 6) RAHP [47], an attention-based model relying on a bidirectional LSTM to capture the sequential dependencies in text. For DoesMR, in addition to the original features, we include all features used by our method, such as the number of citations within the review and the number of keywords referring to a paper's content. All other methods use only textual data and hence cannot leverage hand-engineered features.

We also compare PGPR with its variant Mix-model that only consists of the machine learning component. Note that in Mix-model, the attention-based model used for the review's embedding is the same model used to evaluate SciBERT and the logistic regression used for the hand-engineered features is similar to DoesMR. All the comparison methods are trained using the same training data, i.e., historical reviews with decisions and new reviews with worker grading, which are aggregated by majority voting.

5.1.7 Parameter Settings. For all the comparison methods, we tune the hyperparameters on the validation set. This includes the learning rate searched in {1e-5, 1e-4, 1e-3, 1e-2, 1e-1}, and the batch size in {8, 16, 32, 64}. For RAHP and PCNN, we vary the dimension of the embedding vector in {50, 100, 200, 300}. We train the models for a maximum of 500 epochs and take the versions that achieve the best performance on the validation set. For PGPR, after concatenating the output from the attention-based model and logistic regression, we use a fully connected layer with tanh activation and ten neurons.

Table 3: Accuracy of approximating meta-decisions with average review scores and weighted average by self-reported confidence, expert grading, and worker grading.

Method	SEM	DSW	ICLR
Average	0.33	0.60	0.69
Confidence-weighted	0.50	*NA*	0.70
Grade-weighted (Experts)	0.83	0.80	*NA*
Grade-weighted (Workers)	*NA*	0.80	0.73

5.1.8 Evaluation Metrics. We measure the effectiveness of expert and worker grading in assessing review conformity by the accuracy of approximating meta-decisions with the grading-weighted average of reviewers' recommendation scores. Given a set of reviews $\mathcal{R}$ on the same paper, we denote the recommendation score of a review $r \in \mathcal{R}$ to the paper by s_r and the average grading the review receives by g_r. The aggregated score of $\mathcal{R}$ is given by:

$$s_{\mathcal{R}} = \frac{\sum_{r \in \mathcal{R}} g_r s_r}{\sum_{r \in \mathcal{R}} g_r}. \tag{14}$$

To measure the performance of PGPR and our baselines, we use accuracy, precision, recall and F1-score over the positive class. Higher values indicate better performance.

5.2 Preliminary Analysis on Peer Grading

We verify the effectiveness of peer grading on review conformity by expert reviewers and by crowd workers. We use the grading to weight reviewers' recommendation scores in score aggregation, and compare to other aggregation methods. We compute the accuracy of approximating meta-decision with the aggregation result.

5.2.1 Grading Reviews by Experts. For our first experiments, we select seven and five borderline papers from SEM and DSW, respectively. We only consider the borderline papers on which reviewers have some disagreement over their recommendations. Reviews from DSW papers are publicly available through OpenReview. For SEM, as the reviews are not publicly available, we contacted the reviewers to get their consent before sharing them with their peers. For both venues, we asked the original reviewers of the same paper to grade each other's reviews. 21 reviewers were involved for SEM providing one review each and 12 reviewers were involved for the DSW papers providing in total 16 reviews. Results are shown in Table 3. We observe that the grade-weighted average of the reviews' recommendations is better at approximating meta-decisions than other means of aggregating review scores. The result verifies that peer grading is a better indicator of review conformity than self-reported confidence scores and can be leveraged to better approximate meta-decisions than existing aggregation methods.

5.2.2 Grading Reviews by Crowd Workers. For this experiment, we use the DSW and ICLR datasets. We do not consider the reviews from SEM since those reviews are not public. Results are shown

Table 4: Performance (Accuracy, Precision, Recall and F1-score) comparison with baseline methods. The best performance is highlighted in bold; the second best performance is marked by '*'.

Method	ICLR-2018				ICLR-2019			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
MILNET	0.533	0.580	0.770	0.660	0.528	0.560	0.860*	0.670
DoesMR	0.678*	0.710*	0.782	0.740*	0.747*	0.752*	0.838	0.792*
SciBERT	0.540	0.678	0.434	0.524	0.583	0.604	0.778	0.680
ALBERT	0.548	0.652	0.516	0.570	0.567	0.590	0.782	0.670
PCNN	0.523	0.624	0.508	0.562	0.516	0.570	0.645	0.605
RAHP	0.593	0.612	0.784*	0.688	0.501	0.570	0.515	0.540
PGPR	**0.781**	**0.822**	**0.810**	**0.810**	**0.799**	**0.770**	**0.917**	**0.840**

in Table 3. We observe that for both venues, the weighted average leveraging worker grading better approximates meta-decisions than the weighted average by self-reported confidence scores or the average without the weighting. Worker grading achieves comparable results to expert grading on DSW reviews. To further compare worker grading with expert grading in ICLR, we derive the peer grading according to the agreement between the reviews' recommendation scores: the mutual grading between two reviewers is set to 4 if they gave the same score; if two reviewers have the same decision (e.g., an accept) with different scores, then we set their mutual grading to 3; if two reviewers have different decisions with a small difference between their scores (e.g., a weak accept and a weak reject), we set their mutual grading to 2; otherwise the mutual grading is set to 1. We calculate the average grading to the same review by workers and experts and observe that on 67% of the reviews, worker grading is similar to expert grading (difference < 1). We also observe that workers and experts have a higher agreement on assigning high grades rather than low ones and that workers tend to be more "generous" in grading reviews. Overall, those results are aligned with related work showing that crowd workers in carefully-designed tasks can provide satisfying outcomes on domain-specific problems [8, 24].

5.3 Comparison with the State of the Art

Table 4 summarizes the performance of PGPR against all the comparison methods on both ICLR-2018 and ICLR-2019. We make several observations.

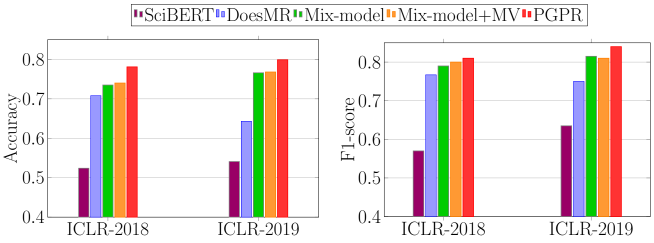
First, we observe that among the comparison methods, DoesMR outperforms the other embedding or deep neural network models. Recall that DoesMR relies on hand-engineered features from scholarly reviews. The result indicates the effectiveness of hand-engineered features as compared to automatically-learned representations in predicting review conformity. This is likely due to the similarity of the vocabulary used in most reviews, making review content alone not highly predictive of review conformity. In contrast, we find through DoesMR that hand-engineered features such as the relative strength of a review recommendation (and confidence) with respect to other reviews on the same paper are highly predictive of the review conformity. Second, we observe that methods developed for modeling scholarly reviews generally outperform those for modeling non-scholarly reviews. In particular, deep neural networks for predicting the helpfulness of product reviews, i.e., PCNN and RAHP, generally reach the lowest performance. These results indicate that models developed in other domains cannot be easily transferred to assess review conformity. Among the two pre-trained language models SciBERT and ALBERT, we observe that SciBERT, which is pre-trained on corpora including computer science publications, does not necessarily outperform ALBERT. Such a result indicates that language models pre-trained on scientific publications are not necessarily effective for modeling scholary reviews.

Most importantly, PGPR achieves the best performance on both datasets. Overall, it improves the second best method by 15.19% accuracy and 9.46% F1-score on ICLR-2018 and by 6.51% accuracy and 6.06% F1-score on ICLR-2019. Such a result underlines the effectiveness of our approach in integrating peer grading into model training. The relatively lower improvement on ICLR-2019 compared to that on ICLR-2018 is likely due to the larger historical data with ground truth available for training, which we investigate latter in our experiments.

5.4 Ablation Studies & Uncertain Reviews

The comparison between PGPR and machine learning baselines is shown in Figure 3. The Mix-model, which consists of the machine learning component of PGPR, outperforms both DoesMr and SciBERT by 11.5% and 40.9% accuracy and by 5.8% and 33.5% F1-score, respectively. These results show the complementary predictive power of hand-engineered features and embeddings. We observe that PGPR outperforms the Mix-model by 5.3% accuracy and by 2.8% F1-score on average on both datasets. This result indicates that using worker's grading improves substantially the model performance. We also observe that PGPR outperforms Mix-model additionally trained with workers grading (aggregated by majority voting), i.e., Mix-model+MV, by 4.8% accuracy and 2.47% F1-score. These results show that PGPR is better at utilizing worker's grading for conformity prediction by taking into account worker reliability.

Table 5 shows a breakdown comparison between the performance of Mix-model and PGPR using the uncertain (actively selected) and certain reviews. We observe that PGPR outperforms the Mix-model by 23.53% and by 5.63% on the uncertain reviews from ICLR-2018 and ICLR-2019, respectively. We also observe that PGPR has little improvement over Mix-model on the certain reviews. These results show that considering workers' grading is important in predicting the conformity of uncertain reviews accurately while



(a) Accuracy (b) F1-score

Figure 3: Comparison between PGPR and machine learning baselines measured by (a) Accuracy and (b) F1-score.

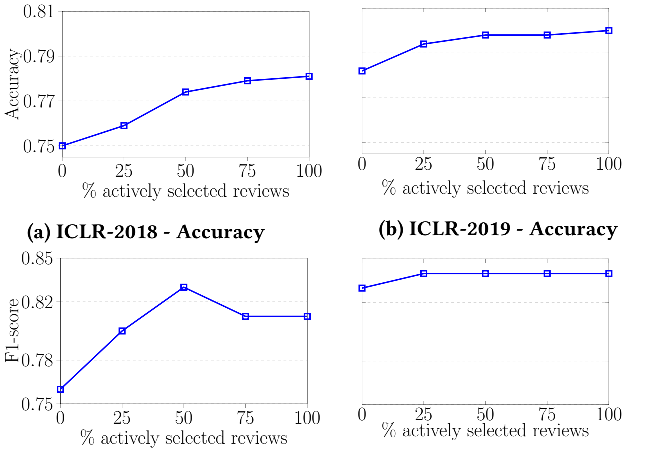
Table 5: Analysis of PGPR performance in terms of accuracy on certain and uncertain reviews.

Method	Dataset					
	ICLR-18			ICLR-19		
	all	certain	uncertain	all	certain	uncertain
Mix-model	0.752	0.845	0.510	0.793	0.801	0.764
PGPR	0.781	0.846	0.630	0.799	0.801	0.807

having little effect on certain ones. We also find that despite the importance of worker's grading in PGPR, the grading alone is not sufficient to predict the conformity of reviews. Using a majority aggregation of grading on the uncertain reviews leads to an accuracy of 0.61 and 0.73 on ICLR-2018 and ICLR-2019, respectively; i.e., less by 3.17% and 9.54% than our framework's performance. This result shows that combining workers grading with machine learning is crucial for an accurate prediction of review's conformity.

5.5 Grading Effect Over Time

The key advantage of our framework is leveraging peer grading for conformity prediction. In what follows, we study the impact of varying the amount of graded reviews on the performance of our framework. We measure the impact on PGPR performance by varying the percentage of the actively selected reviews. We split the graded reviews by s_{act} where we vary s_{act} between 0% and 100%, where s_{act} = 50% means that we use 50% of the graded reviews in addition to the historical data for training. The results are shown in Figure 4 where we use the same y-scale for ICLR-2018 and ICLR-2019 for ease of comparison. We observe that the performance of our framework increases along with the increase of s_{act} on the ICLR-2018 dataset while it gradually stabilizes with the increase of s_{act} on the ICLR-2019 data. This on one hand, confirms the effectiveness of integrating peer grading for model performance. On the other hand, using PGPR in subsequent editions of the same conference requires less grading from one year to the next, as it gradually "learns" the conformity standards of the conference. This property is highly desirable in real-world scenarios as with the increase of the number of submissions (and consequently the number of reviews) our model improves its prediction on the conformity of reviews while requiring fewer reviews to be graded.



(a) ICLR-2018 - Accuracy (b) ICLR-2019 - Accuracy

(c) ICLR-2018 - F1-score (d) ICLR-2019 - F1-score

Figure 4: Performance of PGPR over the two years (2018-2019) with an increasing number of actively graded reviews.

6 CONCLUSION

In this paper, we presented a human-AI approach that estimates the conformity of scholarly reviews by leveraging both human and machine intelligence. We introduced peer grading mechanisms that involve peer reviewers to grade each others' reviews anonymously and a Bayesian framework that seamlessly integrates peer grading with a machine learning model for review conformity assessment. The peer grading mechanism can be easily incorporated into current peer review systems without inducing much extra effort from the reviewers. The machine learning model trained by the Bayesian framework can continuously learn from new grading from peer reviewers over time. Through a crowdsourced, longitudinal study over a three years-worth dataset, we showed that our approach substantially improves the state of the art and that the machine learning in our framework can largely improve the performance over three consecutive years.

In future work, we plan to study transfer learning for our proposed framework such that it can be applied to detect low-conformity reviews in other conferences and journals.

ACKNOWLEDGMENTS

This work was supported by the the European Research Council under grant agreement No 683253 (GraphInt) and the Swiss National Funding under grant agreement No 169840 (ProvDS).

We thank the expert reviewers from SEMANTICS and the International Workshop on Decentralizing the Semantic Web for their participation in our study and for providing valuable feedback. We also thank the crowd workers for their contribution to our study.

REFERENCES

[1] ACM. 2018. Policy on Roles and Responsibilities in ACM Publishing. https://www.acm.org/publications/policies/roles-and-responsibilities. Accessed: 2020-02-26.

[2] Anastasia Ailamaki, Periklis Chrysogelos, Amol Deshpande, and Tim Kraska. 2019. The SIGMOD 2019 Research Track Reviewing System. *ACM SIGMOD Record* 48, 2 (2019), 47–54.

[3] Bruce Alberts, Brooks Hanson, and Katrina L. Kelner. 2008. Reviewing Peer Review. *Science* 321, 5885 (2008), 15–15.

[4] Ammar Ammar and Devavrat Shah. 2012. Efficient rank aggregation using partial data. *ACM SIGMETRICS Performance Evaluation Review* 40, 1 (2012), 355–366.

[5] Hannah Bast. 2020. How Objective is Peer Review? https://cacm.acm.org/blogs/blog-cacm/248824-how-objective-is-peer-review. Accessed: 2021-02-02.

[6] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: Pretrained Language Model for Scientific Text. In *EMNLP*. Association for Computational Linguistics, USA, 3613–3618.

[7] Phil Bernstein, Michael Brodie, Stefano Ceri, David DeWitt, Mike Franklin, Hector Garcia-Molina, Jim Gray, Jerry Held, Joe Hellerstein, HV Jagadish, et al. 1998. The Asilomar report on database research. *ACM Sigmod record* 27, 4 (1998), 74–80.

[8] Roi Blanco, Harry Halpin, Daniel M Herzig, Peter Mika, Jeffrey Pound, Henry S Thompson, and Thanh Tran Duc. 2011. Repeatable and reliable search system evaluation using crowdsourcing. In *SIGIR*. ACM, USA, 923–932.

[9] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *J. Amer. Statist. Assoc.* 112, 518 (2017), 859–877.

[10] Alejandro Uriel Carbonara, Anupam Datta, Arunesh Sinha, and Yair Zick. 2015. Incentivizing peer grading in MOOCs: an audit game approach. In *IJCAI*. AAAI Press, USA, 497–503.

[11] NeurIPS 2020 Program Chairs. 2020. Getting Started with NeurIPS 2020. https://medium.com/@NeurIPSConf/getting-started-with-neurips-2020-e350f9b39c28. Accessed: 2021-02-02.

[12] Hongbo Deng, Irwin King, and Michael R Lyu. 2008. Formal models for expert finding on dblp bibliography data. In *ICDM*. IEEE, USA, 163–172.

[13] Susan T Dumais and Jakob Nielsen. 1992. Automating the assignment of submitted manuscripts to reviewers. In *SIGIR*. ACM, USA, 233–244.

[14] Laura J Falkenberg and Patricia A Soranno. 2018. Reviewing reviews: An evaluation of peer reviews of journal article submissions. *Limnology and Oceanography Bulletin* 27, 1 (2018), 1–5.

[15] Hui Fang and ChengXiang Zhai. 2007. Probabilistic models for expert finding. In *ECIR*. Springer, Switzerland, 418–430.

[16] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *Journal of machine learning research* 4, Nov (2003), 933–969.

[17] Yang Gao, Steffen Eger, Ilia Kuznetsov, Iryna Gurevych, and Yusuke Miyao. 2019. Does My Rebuttal Matter? Insights from a Major NLP Conference. In *NAACL-HLT*. Association for Computational Linguistics, USA, 1274–1290.

[18] I Hames. 2013. COPE ethical guidelines for peer reviewers. *COPE Council* 1 (2013), 1–5.

[19] Seth Hettich and Michael J Pazzani. 2006. Mining for proposal reviewers: lessons learned at the national science foundation. In *KDD*. ACM, USA, 862–871.

[20] Jian Jin, Qian Geng, Qian Zhao, and Lixue Zhang. 2017. Integrating the trend of research interest for reviewer assignment. In *WWW Companion*. IW3C2, ACM, USA, 1233–1241.

[21] Maryam Karimzadehgan and ChengXiang Zhai. 2009. Constrained multi-aspect expertise matching for committee review assignment. In *CIKM*. ACM, USA, 1697–1700.

[22] Maryam Karimzadehgan, ChengXiang Zhai, and Geneva Belford. 2008. Multi-aspect expertise matching for review assignment. In *CIKM*. ACM, USA, 1113–1122.

[23] Ngai Meng Kou, U Leong Hou, Nikos Mamoulis, and Zhiguo Gong. 2015. Weighted coverage based reviewer assignment. In *SIGMOD*. ACM, USA, 2031–2046.

[24] Evgeny Krivosheev, Fabio Casati, and Boualem Benatallah. 2018. Crowd-based multi-predicate screening of papers in literature reviews. In *WWW*. ACM, USA, 55–64.

[25] Igor Labutov and Christoph Studer. 2017. JAG: a crowdsourcing framework for joint assessment and peer grading. In *AAAI*. AAAI Press, USA, 1010–1016.

[26] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *ICLR*. OpenReview.net, 16 pages.

[27] Cheng Long, Raymond Chi-Wing Wong, Yu Peng, and Liangliang Ye. 2013. On good and fair paper-reviewer assignment. In *ICDM*. IEEE, USA, 1145–1150.

[28] Tyler McDonnell, Matthew Lease, Mucahid Kutlu, and Tamer Elsayed. 2016. Why is that relevant? Collecting annotator rationales for relevance judgments. In *HCOMP*. AAAI Press, USA, 139–148.

[29] Fei Mi and Dit-Yan Yeung. 2015. Probabilistic graphical models for boosting cardinal and ordinal peer grading in MOOCs. In *AAAI*. AAAI Press, USA, 454–460.

[30] David Mimno and Andrew McCallum. 2007. Expertise modeling for matching papers with reviewers. In *KDD*. ACM, USA, 500–509.

[31] Ioannis Mitliagkas, Aditya Gopalan, Constantine Caramanis, and Sriram Vishwanath. 2011. User rankings from comparisons: Learning permutations in high dimensions. In *Allerton*. IEEE, New York City, USA, 1143–1150.

[32] Springer Nature. 2020. Focus on Peer Review. https://masterclasses.nature.com/focus-on-peer-review-online-course/16605550. Accessed: 2021-02-02.

[33] Iyiola E Olatunji, Xin Li, and Wai Lam. 2019. Context-aware helpfulness prediction for online product reviews. In *AIRS*. Springer, Switzerland, 56–65.

[34] ACM Transactions on Social Computing. 2019. Review Guidelines. https://dl.acm.org/journal/tsc/review-guidelines. Accessed: 2020-10-17.

[35] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*. ACL, USA, 1532–1543.

[36] IEEE Potentials. 2020. Reviewer Guidelines. https://www.ieee.org/content/dam/ieee-org/ieee/web/org/members/students/reviewer_guidelines_final.pdf. Accessed: 2020-02-26.

[37] Simon Price and Peter A Flach. 2017. Computational support for academic peer review: A perspective from artificial intelligence. *Commun. ACM* 60, 3 (2017), 70–79.

[38] Nihar B. Shah and Zachary Lipton. 2020. SIGMOD 2020 Tutorial on Fairness and Bias in Peer Review and Other Sociotechnical Intelligent Systems. In *SIGMOD Conference*. ACM, USA, 2637–2640.

[39] Nihar B Shah, Behzad Tabibian, Krikamol Muandet, Isabelle Guyon, and Ulrike Von Luxburg. 2018. Design and analysis of the nips 2016 review process. *The Journal of Machine Learning Research* 19, 1 (2018), 1913–1946.

[40] Richard Snodgrass. 2006. Single-versus double-blind reviewing: An analysis of the literature. *ACM Sigmod Record* 35, 3 (2006), 8–21.

[41] SPRINGER. 2020. GUIDELINES FOR REVIEWERS. https://www.springer.com/authors/manuscript+guidelines?SGWID=0-40162-6-1261021-0. Accessed: 2020-02-26.

[42] Wenbin Tang, Jie Tang, and Chenhao Tan. 2010. Expertise matching via constraint-based optimization. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 1. IEEE, IEEE, USA, 34–41.

[43] Andrew Tomkins, Min Zhang, and William D Heavlin. 2017. Reviewer bias in single-versus double-blind peer review. *Proceedings of the National Academy of Sciences* 114, 48 (2017), 12708–12713.

[44] Dimitris G Tzikas, Aristidis C Likas, and Nikolaos P Galatsanos. 2008. The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine* 25, 6 (2008), 131–146.

[45] Ke Wang and Xiaojun Wan. 2018. Sentiment analysis of peer review texts for scholarly papers. In *SIGIR*. ACM, ACM, USA, 175–184.

[46] Wanyuan Wang, Bo An, and Yichuan Jiang. 2018. Optimal Spot-Checking for Improving Evaluation Accuracy of Peer Grading Systems. In *AAAI*. AAAI Press, USA, 833–840.

[47] Wenxuan Zhang, Wai Lam, Yang Deng, and Jing Ma. 2020. Review-guided Helpful Answer Identification in E-commerce. In *WWW*. ACM / IW3C2, USA, 2620–2626.

A APPENDIX

In this section, we present the proofs for our lemmas. We apply the same notational conventions as in the paper. We use the symbol $\propto$ to denote that two variables are proportionally related.

A.1 Proof of Lemma 3.1

Proof. To minimize the KL divergence, we assume the variational distribution follows the same distribution as the latent variable [44]. For $q(z_i)$, we obtain

$$q(z_i) \propto h_{q(r_g,b_g)}[p(z_i, r, b, \boldsymbol{A}_{i,*}, \mathcal{W})], \tag{15}$$

where $h_{q(r_g,b_g)}$ denotes the exponential of expectation $\exp\{\mathbb{E}_x[\log(\cdot)]\}$ with x being a variational distribution. According to the mean field approximation, the probability $p(z_i, r, b, \boldsymbol{A}_{i,*}, \mathcal{W})$ factorizes over $\mathcal{G}_i$ and Eq.(15) can be written as:

$$q(z_i) \propto \prod_{g\in\mathcal{G}_i} h_{q(r_g,b_g)}[p(z_i, r_g, b_g, \boldsymbol{A}_{r,g}, \mathcal{W})]. \tag{16}$$

By applying the chain rule on the probability $p(z_i, r, b, \boldsymbol{A}_{i,*}, \mathcal{W})$ and keeping only the terms that depend on z_i, we get:

$$q(z_i) \propto p(z_i|\boldsymbol{x}_i, \mathcal{W}) \prod_{g\in\mathcal{G}_i} \underbrace{h_{q(r_g,b_g)}[p(\boldsymbol{A}_{r,g}|z_i, r_g, b_g)]}_{\mathcal{T}_1} \tag{17}$$

The term $\mathcal{T}_1$ can be expressed using the probability density function of a Gaussian distribution of $p(\boldsymbol{A}_{r,g}|z_i, r_g, b_g)$ where the logarithm of the Gaussian distribution is given by

$$\log(\mathcal{N}(z_i + b_g, \frac{1}{r_g}) \propto \frac{1}{2}\log r_g - \frac{r_g}{2}(\boldsymbol{A}_{\boldsymbol{i,g}} - z_i - b_g)^2 \tag{18}$$

Then, we keep the terms dependent on z_i and apply $\mathbb{E}_{r_g,b_g}$:

$$\mathbb{E}_{r_g,b_g}[\log(\mathcal{N}(z_i + b_g, \frac{1}{r_g})] \propto \mathbb{E}_{r_g,b_g}[\frac{r_g}{2}] \times \mathbb{E}_{r_g,b_g}[(\boldsymbol{A}_{\boldsymbol{i,g}} - z_i - b_g)^2] \tag{19}$$

We expand the second term by the square factor and get:

$$(\boldsymbol{A}_{\boldsymbol{i,g}} - z_i - b_g)^2 = \boldsymbol{A}_{\boldsymbol{i,g}}^2 + z_i^2 + b_g^2 + 2z_i b_g - 2\boldsymbol{A}_{\boldsymbol{i,g}} z_i - 2\boldsymbol{A}_{\boldsymbol{i,g}} b_g \tag{20}$$

We eliminate the terms independent from z_i and apply $\mathbb{E}_{r_g,b_g}$:

$$\begin{aligned}\mathbb{E}_{r_g,b_g}[(\boldsymbol{A}_{\boldsymbol{i,g}} - z_i - b_g)^2] &= \mathbb{E}_{r_g,b_g}[z_i^2] + 2\mathbb{E}_{r_g,b_g}[z_i]\mathbb{E}_{r_g,b_g}[b_g] \\ &- 2\mathbb{E}_{r_g,b_g}[\boldsymbol{A}_{\boldsymbol{i,g}}]\mathbb{E}_{r_g,b_g}[z_i]\end{aligned} \tag{21}$$

Using the properties of b_g distribution and since $\boldsymbol{A}_{\boldsymbol{i,g}}$ and z_i do not depend on b_g, the terms in Eq.(21) are expressed as follows:

$$\mathbb{E}_{b_g}[\boldsymbol{A}_{\boldsymbol{i,g}}] = \boldsymbol{A}_{\boldsymbol{i,g}}, \mathbb{E}_{b_g}[z_i^2] = z_i^2, \mathbb{E}_{b_g}[z_i] = z_i, \mathbb{E}_{b_g}[b_g] = m_g \tag{22}$$

The first term in Eq.(19) is the mean of r_g's distribution, i.e., $\frac{A_g}{B_g}$. We replace the second term by the expressions in Eqs.(21)-(22):

$$\mathbb{E}_{r_g,b_g}[\log(\mathcal{N}(z_i + b_g, \frac{1}{r_g})] \propto \frac{A_g}{2B_g} \times (z_i^2 + 2z_i(m_g - \boldsymbol{A}_{\boldsymbol{i,g}})) \tag{23}$$

We now replace in Eq.(17) $p(z_i|\boldsymbol{x}_i, \mathcal{W})$ by the probability density function of z_i and the term $\mathcal{T}_1$ by its simplification in Eq.(23).

$$\begin{aligned} q(z_i) &\propto \mathcal{N}(\mu_i, \sigma_i) \prod_{g \in \mathcal{G}_i} \exp\{\frac{A_g}{2B_g} \times (z_i^2 + 2z_i(m_g - \boldsymbol{A}_{\boldsymbol{i,g}}))\} \\ &\propto \exp\{\frac{-1}{2}[(\sum_g \frac{A_g}{B_g} + \frac{1}{\sigma_i^2})z_i^2 - 2(\sum_g \frac{A_g}{B_g}(\boldsymbol{A}_{\boldsymbol{i,g}} - m_g) + \frac{\mu_i}{\sigma_i^2})]z_i\} \\ &\propto \mathcal{N}(\frac{W}{V}, \frac{1}{V}), \end{aligned}$$

where $W = \sum_g \frac{A_g}{B_g}(\boldsymbol{A}_{i,g} - m_g) + \frac{\mu_i}{\sigma_i^2}$ and $V = (\sum_g \frac{A_g}{B_g} + \frac{1}{\sigma_i^2})$, which concludes the proof. □

A.2 Proof of Lemma 3.2

Proof. Following the reasoning in Eq.(15)-(17) for r_g, we get:

$$q(r_g) \propto p(r_g|A_g, B_g) \prod_{i \in \mathcal{I}_g} \underbrace{h_{q(z_i,b_g)}[p(\boldsymbol{A}_{r,g}|z_i, r_g, b_g)]}_{\mathcal{T}_2} \tag{24}$$

To incrementally update the grader reliability, we simplify the term $\mathcal{T}_2$ in Eq.(24). First, we use Eq.(20) to expand the term $(\boldsymbol{A}_{\boldsymbol{i,g}} - z_i - b_g)^2$ and apply the expectation $\mathbb{E}_{z_i,b_g}(\cdot)$. Then, using the properties of the Gaussian distribution of z_i and b_g, we get:

$$\mathbb{E}_{z_i,b_g}[z_i] = \mu_i, \mathbb{E}_{z_i,b_g}[z_i^2] = \sigma_i^2, \mathbb{E}_{z_i,b_g}[b_g] = m_g, \mathbb{E}_{z_i,b_g}[b_g^2] = \frac{1}{\alpha_g} \tag{25}$$

The term $\mathbb{E}_{z_i,b_g}[(\boldsymbol{A}_{\boldsymbol{i,g}} - z_i - b_g)^2]$ can be simplified using the expressions in Eq.(25). We denote the simplification with M_i

$$M_i = \boldsymbol{A}_{\boldsymbol{i,g}}^2 + \sigma_i^2 + \frac{1}{\alpha_g} + 2(\mu_i m_g - \boldsymbol{A}_{\boldsymbol{i,g}}\mu_i - \boldsymbol{A}_{\boldsymbol{i,g}} m_g) \tag{26}$$

The expectation of Eq.(18) conditioned on z_i and b_g can be simplified using Eq.(26):

$$\mathbb{E}_{z_i,b_g}[\log(\mathcal{N}(z_i + b_g, \frac{1}{r_g})] \propto \frac{1}{2}\log r_g - \frac{r_g}{2}M_i \tag{27}$$

Now, we can replace the term $\mathcal{T}_2$ in Eq.(24) by its expression in Eq.(27) and the probability $p(r_g|A_g, B_g)$ by its density function.

$$\begin{aligned} q(r_g) &\propto \Gamma(A_g, B_g) \prod_{i \in \mathcal{I}_g} \exp\{\frac{1}{2}\log r_g - \frac{r_g}{2}M_i\} \\ &\propto \frac{1}{\Gamma(A_g)} B_g^{A_g} r_g^{A_g + \frac{|\mathcal{I}_g|}{2} - 1} \exp\{-(b_g + \frac{1}{2}\sum_{i \in \mathcal{I}_g} M_i) r_g\} \\ &\propto Gamma(X, Y) \end{aligned}$$

where $X = A_g + \frac{|\mathcal{I}_g|}{2}$ and $Y = B_g + \frac{1}{2}(\frac{|\mathcal{I}_g|}{\alpha_g} + \sum_i[\boldsymbol{A}_{i,g}^2 + \sigma_i^2 + 2\mu_i(m_g - \boldsymbol{A}_{i,g}) - 2\boldsymbol{A}_{i,g}m_g])$ which concludes the proof. □

A.3 Proof of Lemma 3.3

Proof. Following the reasoning in Eq.(15)-(17) for b_g, we get:

$$q(b_g) \propto p(b_g|m_g, \alpha_g) \prod_{i \in \mathcal{I}_g} \underbrace{h_{q(z_i,r_g)}[p(\boldsymbol{A}_{r,g}|z_i, r_g, b_g)]}_{\mathcal{T}_3} \tag{28}$$

To incrementally update worker's bias, we simplify the term $\mathcal{T}_3$ in Eq.(28). In order to do that, we use Eq.(20) to expand the term $(\boldsymbol{A}_{\boldsymbol{i,g}} - z_i - b_g)^2$ and apply the expectation $\mathbb{E}_{z_i,r_g}(\cdot)$. Then, we use the properties of the Gaussian distribution of z_i and the independence property of b_g with respect to z_i and r_g and get:

$$\mathbb{E}_{z_i,r_g}[z_i] = \mu_i, \mathbb{E}_{z_i,r_g}[z_i^2] = \sigma_i^2, \mathbb{E}_{z_i,r_g}[b_g] = b_g, \mathbb{E}_{z_i,r_g}[b_g^2] = b_g^2 \tag{29}$$

Using the expressions in Eq.(29) and by eliminating the terms that do not depend on b_j, the expectation $\mathbb{E}_{z_i,b_g}[(\boldsymbol{A}_{\boldsymbol{i,g}} - z_i - b_g)^2]$ can be simplified as follows.

$$\mathbb{E}_{z_i,b_g}[(\boldsymbol{A}_{\boldsymbol{i,g}} - z_i - b_g)^2] = b_g^2 + 2(\mu_i - \boldsymbol{A}_{\boldsymbol{i,g}})b_g \tag{30}$$

The expectation term $\mathbb{E}_{z_i,r_g}[\log(\mathcal{N}(z_i + r_g, \frac{1}{r_g})]$ is given by:

$$\mathbb{E}_{z_i,r_g}[\log(\mathcal{N}(z_i + r_g, \frac{1}{r_g})] \propto -\frac{A_g}{2B_g}(b_g^2 + 2(\mu_i - \boldsymbol{A}_{\boldsymbol{i,g}})b_g), \tag{31}$$

where $\frac{A_g}{B_g}$ is the mean of the reliability density function. We can now replace the term $\mathcal{T}_3$ in Eq.(28) by its expression in Eq.(31) and the probability $p(b_g|m_g, \alpha_g)$ by its density function.

$$\begin{aligned} q(b_g) &\propto \mathcal{N}(m_g, b_g) \prod_{i \in \mathcal{I}_g} \exp\{-\frac{A_g}{2B_g}(b_g^2 + 2(\mu_i - \boldsymbol{A}_{\boldsymbol{i,g}})b_g))\} \\ &\propto \exp(-\frac{1}{2}((\frac{A_g|\mathcal{I}_g|}{b_g} + \alpha_g)b_g^2 - 2[\alpha_g m_g + \frac{A_g}{b_g}\sum_i(\boldsymbol{A}_{\boldsymbol{i,j}} - \mu_i)]b_g)) \\ &\propto \mathcal{N}(\frac{L}{K}, \frac{1}{K}) \end{aligned}$$

where $K = \frac{A_g|\mathcal{I}_g|}{B_g} + \alpha_g$ and $L = \alpha_g m_g + \frac{A_g}{B_g}\sum_r(\boldsymbol{A}_{i,g} - \mu_i)$ which concludes the proof. □