Eliciting Informative Text Evaluations with Large Language Models

YUXUAN LU[*†], Peking University, China
SHENGWEI XU[*‡], University of Michigan, USA
YICHI ZHANG[‡], University of Michigan, USA
YUQING KONG[†], Peking University, China
GRANT SCHOENEBECK[‡], University of Michigan, USA

Peer prediction mechanisms motivate high-quality feedback with provable guarantees. However, current methods only apply to rather simple reports, like multiple-choice or scalar numbers. We aim to broaden these techniques to the larger domain of text-based reports, drawing on the recent developments in large language models (LLMs). This vastly increases the applicability of peer prediction mechanisms as textual feedback is the norm in a large variety of feedback channels: peer reviews, e-commerce customer reviews, and comments on social media.

We introduce two mechanisms, the Generative Peer Prediction Mechanism (GPPM) and the Generative Synopsis Peer Prediction Mechanism (GSPPM). These mechanisms utilize LLMs as predictors, mapping from one agent's report to a prediction of her peer's report. Theoretically, we show that when the LLM prediction is sufficiently accurate, our mechanisms can incentivize high effort and truth-telling as an (approximate) Bayesian Nash equilibrium. Empirically, we confirm the efficacy of our mechanisms through experiments conducted on two real datasets: the Yelp review dataset and the ICLR OpenReview dataset. We highlight the results that on the ICLR dataset, our mechanisms can differentiate three quality levels — human-written reviews, GPT-4-generated reviews, and GPT-3.5-generated reviews in terms of expected scores. Additionally, GSPPM penalizes LLM-generated reviews more effectively than GPPM.

CCS Concepts: • **Applied computing** → *Economics.*

Additional Key Words and Phrases: Information Elicitation, Peer Prediction, Large Language Models (LLMs)

ACM Reference Format:
Yuxuan Lu, Shengwei Xu, Yichi Zhang, Yuqing Kong, and Grant Schoenebeck. 2024. Eliciting Informative Text Evaluations with Large Language Models. In *Conference on Economics and Computation (EC '24), July 8–11, 2024, New Haven, CT, USA.* ACM, New York, NY, USA, 31 pages. https://doi.org/10.1145/3670865.3673532

[*]Both authors contributed equally to the paper and are listed in alphabetical order.
[†]Supported by National Natural Science Foundation of China award number 62372007.
[‡]Supported by United States National Science Foundation award number 2313137.

Authors' Contact Information: Yuxuan Lu, Peking University, Beijing, China, yx_lu@pku.edu.cn; Shengwei Xu, University of Michigan, Ann Arbor, USA, shengwei@umich.edu; Yichi Zhang, University of Michigan, Ann Arbor, USA, yichiz@umich.edu; Yuqing Kong, Peking University, Beijing, China, yuqing.kong@pku.edu.cn; Grant Schoenebeck, University of Michigan, Ann Arbor, USA, schoeneb@umich.edu.

EC '24, July 8–11, 2024, New Haven, CT, USA

ACM ISBN 979-8-4007-0704-9/24/07
https://doi.org/10.1145/3670865.3673532

1 Introduction

Consider the following review for an academic paper:

> *"I didn't get much out of reading this paper. Their methods do not seem very rigorous. I don't think the conclusions are supported very well."*

The above review is not very informative: its initial critique is too general, and the issues with the methods and conclusions should have been explained more thoroughly. If the peer review process only gathers reviews of this quality, it will struggle to make useful and fair publication decisions. The problem has been exacerbated by large language models (LLMs), which greatly reduce the cost of generating reviews that closely mimic human-written reviews but often lack substantial insight [Liang et al., 2023].

Here are two reviews of a randomly selected paper submitted to the ICLR[1] 2020. An interesting exercise is to determine which was generated by an LLM and which was written by a human.

... I lean towards rejecting this paper however, because I am not convinced of the results' significance. We already know how to learn symmetric functions (see Exercise 3.26 in Mohri et al., 2018). The authors' results show that we can inject this knowledge into a neural network at initialization, and then run SGD without making things too much worse. I do not see how these ideas might apply to more substantial learning problems where our prior knowledge is less precise. ...	... The paper makes a valuable theoretical contribution to the understanding of neural network initialization, particularly in the context of symmetric functions. The empirical validation is a strong point, although the experiments could be expanded to provide a more comprehensive evaluation. The paper's focus on a single hidden layer network is both a strength, in terms of theoretical tractability, and a weakness, in terms of practical relevance. To move forward, the authors should consider extending their analysis to more complex architectures and providing a broader empirical evaluation. ...

Fig. 1. An example from our study: Two reviews of a submission at ICLR2020, the left one by a human reviewer, and the right one by GPT-4.

Due to LLMs, it is no longer possible to filter out low-quality reviews by their length, lack of any relation to the paper at hand, or poor grammatical constructions. In the above example, we can observe that the AI-generated review looks informative and effectively summarizes the paper. However, it lacks the depth and insight of the expert human review.

The need to obtain high-quality subjective human evaluation extends beyond academia to other domains, including business, the arts, and more. For example, if customer feedback on Amazon and Yelp is inundated with shallow reviews or those generated by LLMs, consumers will struggle with making well-informed decisions [Resnick et al., 2000, Tadelis, 2016]. The problem of incentivizing high-quality reviews is more important and daunting than ever.

One approach is to provide well-designed incentives for the reviewer that, in particular, reward high-quality reviews more than low-quality reviews [Srinivasan and Morgenstern, 2021]. However, because reviews are subjective, their correctness cannot be directly verified. This inherent challenge complicates the evaluation of review quality.

[1]The International Conference on Learning Representations (ICLR), a top-tier machine learning conference, makes all of its peer review data openly accessible on OpenReview (https://openreview.net).

One straightforward idea is to ask other people to judge the quality of the reviews. But then we face a new challenge: how to motivate these new judges? Moreover, an automated approach that does not introduce additional participants and procedures is preferable.

Prior work has proposed the peer prediction mechanism, a powerful tool to elicit subjective information [Miller et al., 2005]. The high-level idea is to determine the reward of a person according to the "correlation" between her report and a peer's report. The underlying intuition is that better, more insightful reports will naturally align more closely with one another. In their setting, when a person puts in the effort to understand a task, she gains a private signal such as "good" or "bad". She can then choose whether to report this signal honestly. Miller et al. [2005] prove that in their setting, truth-telling is an equilibrium—if a participant believes other people will invest effort and tell the truth, she should also do this.

However, implementing the above mechanism requires knowledge of the prior: the joint distribution of the private signals. In the original peer prediction mechanism [Miller et al., 2005], agents are asked to report their private signals such as "good" or "bad". The mechanism then predicts a peer's report, such as 70% "good", by getting a posterior based on an agent's report and the prior. The agent is then rewarded for the accuracy of this posterior prediction. Assuming the mechanism has perfect knowledge of the prior, this incentivizes truth-telling because only an honest report can lead to the optimal posterior.

The mechanism's required knowledge of the prior has been seen as a major impediment to real-world implementation of peer-prediction mechanisms. Significant advances, that follow two main approaches, have enabled overcoming this limitation in several settings. Both these approaches circumvented the requirement of knowing the prior by learning, not the prior itself, but a proxy, usually the relationship between the agent reports, from agent reports themselves.

The first approach, often called the multitask setting, involves assigning agents to multiple a priori similar tasks. This allows learning the structure of agent reports and enables measuring the amount of information in common between agent responses [Dasgupta and Ghosh, 2013, Kong, 2020, Kong and Schoenebeck, 2019, Liu et al., 2023, Shnayder et al., 2016, Zhang and Schoenebeck, 2023a]. In certain settings such as multiple choice questions, even a small number of tasks may suffice [Burrell and Schoenebeck, 2021, Kong, 2024, Schoenebeck and Yu, 2020]. The second approach called the signal-prediction framework, pioneered by Prelec [2004], which is independent and concurrent to Miller et al. [2005], involves eliciting second-order predictions, that is asking how they believe other agents will respond, for example, "I think 70% of my peers will answer 'good' " [Chen et al., 2021, Radanovic and Faltings, 2014, Schoenebeck and Yu, 2023, Witkowski and Parkes, 2012].

However, because both these approaches rely on learning from agent responses or predictions, they work better when the space is simple—either categorical (such as a multi-choice question) or numerical (such as a rating between 0 and 10). Otherwise, the structure is too involved to learn in the multitask setting, and the forecasts can not be communicated efficiently for forecast elicitation.

However, reducing to such a simple space often loses the rich information within the textual judgments. For example, in peer review, the decision of the editor/area chair often relies more on the arguments and justifications in the textual reviews rather than merely on numerical ratings. Furthermore, on online platforms, the inflation of ratings makes them less reliable and distinguishable, while textual reviews tend to be more stable [Filippas et al., 2018].

Given these limitations and the recent success of large language models (LLMs), our research question is: **can we develop automated mechanisms that effectively incentivize high-quality, informative textual feedback by rewarding it more than generic or low-quality content?**

Intuitively, eliciting textual feedback is inherently more difficult than eliciting numerical or categorical responses. However, the recent rise of powerful Large Language Models (LLMs) has surprisingly flipped this script. LLMs, more or less, estimate the probability distribution of the entirety of human language. Thus, our goal is to instead run the original peer-prediction mechanism by using LLMs to gain access to the prior. We use the LLM's ability to analyze the structure of textual responses and predict the probability of one text given another text (LLM-prediction). This directly addresses the "knowing prior" problem and eliminates the need for multiple tasks or second-order predictions (common for categorical/numerical responses). It is somewhat paradoxical that moving to this much *more* complex domain actually may make the entire task easier! In essence, LLMs make eliciting textual responses easier than simpler formats, as textual responses offer more complexity that LLMs can leverage, while simpler formats lack this richness.

Directly employing LLMs may reward superficial similarities, such as matching speaking styles, or reviews that offer no more than a reiteration of the paper's abstract, which may benefit LLM-generated reviews. Ultimately, the goal is to encourage reviewers to delve deeper, providing unique perspectives. A related question is **can we distinguish the valuable, unique human expert reviews from the coherent yet potentially superficial reviews generated by AI?**

To answer the above research question and not reward superficial similarities, when using LLMs to compute the correlation, it is important to effectively condition out "shortcut" information such as language styles and information contained in any synopsis of the reviewed item. We borrow the term from "shortcut learning", where a machine learning algorithm learns the undesired information from data that is strongly correlated with labels on the training data but lacks generality [Geirhos et al., 2020]. In our setting, for example, a human-written review can have a high correlation with an LLM-generated review because they mentioned several particular terms in the paper. However, such "shortcut" information may lead to unintended rewards for shallow reviews (e.g., LLM-generated reviews) and noise caused by different language styles. By conditioning out "shortcut" information, we aim to filter out these superficial aspects and focus on rewarding reviews that demonstrate a deeper level of engagement.

1.1 Our Contribution

We apply the LLM-prediction to peer prediction and propose two mechanisms—the Generative Peer Prediction mechanism (GPPM) and the Generative Synopsis Peer Prediction mechanism (GSPPM). At a high level, the former rewards a review based on how much it helps predict the contents of another review. The latter, however, rewards a review based on how much *more* it helps predict the contents of another review than a mere synopsis of the item to be reviewed, such as the abstract of a paper, thereby conditioning out the "shortcut" information derived from superficial information contained in the synopsis.

We first use theory to present the main idea of our method, where we consider a model with three layers of effort: high, low, and no effort. We show that when the LLM prediction is sufficiently accurate, both mechanisms can incentivize high effort and truth-telling as an (approximate) Bayesian Nash equilibrium. By conditioning on a synopsis of the item to be reviewed, GSPPM can further shrink the gap of expected scores between low-effort and no-effort reporting while preserving the gap between high-effort and low-effort reporting.

We then propose two implementations for getting the LLM-prediction—Token and Judgment. Token predicts the content of a response by using the 'predicting next word' mechanism of a language model, while Judgment first summarizes one textual response into individual judgments and then estimates a prediction for each judgment by querying an LLM. We highlight that for a robust implementation, it is necessary to preprocess the responses. We propose a straightforward yet effective heuristic preprocessing method. This involves using an LLM to rephrase and summarize the initial responses, aiming to standardize the language style and remove superficial content, and consequently, mitigate the impact of the "shortcut".

Moreover, we conduct experiments on the mechanisms using an ICLR dataset and a Yelp review dataset and observed the following results:

Result 1: GPPM can effectively penalize report degradations. We apply three report degradation methods, which degrade the information of an agent's report. In both ICLR and Yelp datasets, we observe that the expected score computed by the GPPM significantly decreases after all degradations.

Result 2: Both GPPM and GSPPM can differentiate three quality levels—human, GPT-4, and GPT-3.5. In the ICLR dataset, we replace an agent's review with a GPT-4-generated review and a GPT-3.5-generated review respectively, representing a decreasing level of effort. We observe that the expected scores computed by both GPPM and GSPPM significantly decrease. Furthermore, the decrease of the GPT-3.5-generated review is larger than that of the GPT-4-generated review.

Result 3: GSPPM penalizes LLM-generated peer review more than GPPM. We find that the GSPPM applies a more significant expected score penalty on the LLM-generated peer reviews, including both GPT-4 and GPT3.5, compared to GPPM, which indicates its improved capacity to distinguish high-quality reports from low-quality reports.

We further note that our mechanisms can serve not only to assess the quality of reviews which can inform decision-making, but also to incentivize effort from agents. This can be interpreted by considering GPT-3.5, GPT-4, and human-written reviews as representing three levels of effort. Thus, by rescaling the scores of the mechanisms into payments, we can reward high-effort reviews much more than low-effort reviews.

Omitted Material. Due to space constraints, a more comprehensive literature review, detailed theoretical analysis (Section 3), prompts used in mechanism implementations (Section 4), and experiment details (Section 6) are provided in the full version at: https://arxiv.org/abs/2405.15077.

2 Preliminaries

This section introduces the classic information elicitation model and the preliminaries that guide the design of our method.

2.1 Model

In our setting, a set of items (e.g., papers or restaurants) are reviewed by a set of agents, where each item is assigned to multiple agents for review. We reduce the problem to the setting where there is only one item to be reviewed by two agents and emphasize that our method can be applied to any item and any pair of agents. Let $I = \{1, 2\}$ be the set of agents reviewing the same item.

Item and Signal. Agents' judgments of the item are influenced by the inherent characteristics of the item and other related background knowledge used to generate the judgment. Let $Z \in \mathcal{Z}$ denote an item such that observing $Z = z$ is sufficient for an agent to form her judgments about the item. Suppose Z is sampled from an unknown common prior π.

Given an item $Z = z$, each agent receives a subjective signal $X_i \in \mathcal{X}$ when evaluating the item. We use x_i to denote a potential value for X_i. Similar to prior literature [Dasgupta and Ghosh, 2013, Miller et al., 2005], we adopt the common assumption that signals are i.i.d. conditioned on the item, i.e., $\Pr[X_i \mid Z = z]$ is identical for any agent i. Furthermore, the signals related to the same item are expected to be correlated in some meaningful way. We thus adopt the following assumption, which is required to guarantee that truthful information is elicitable [Crémer and McLean, 1985].

Assumption 2.1 (Stochastic Relevance). *For any $x_i, x_i' \in \mathcal{X}$, $x_i \neq x_i'$, there exists $x_j \in \mathcal{X}$ such that*

$$\Pr[X_j = x_j \mid X_i = x_i] \neq \Pr[X_j = x_j \mid X_i = x_i'].$$

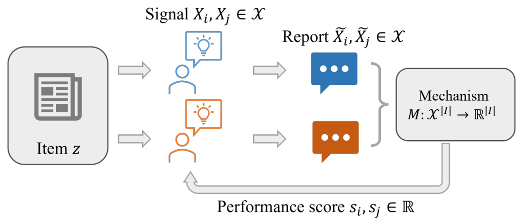


Fig. 2. An Overview of Our Information Elicitation Model

Reporting Strategy. We assume the report of agent i, denoted as $\tilde{X}_i \in \mathcal{X}$, shares the same domain as the signal. Agents can truthfully report their signals or manipulate their signals as reports. However, only reports (not signals) are observable to the mechanism. Let $\sigma_i : \mathcal{X} \to \Delta_{\mathcal{X}}$ denote the reporting strategy of agent i so that $\tilde{X}_i = \sigma_i(X_i)$. Let τ denote the truthful reporting strategy such that $\tau(X_i) = X_i$ for any signal.

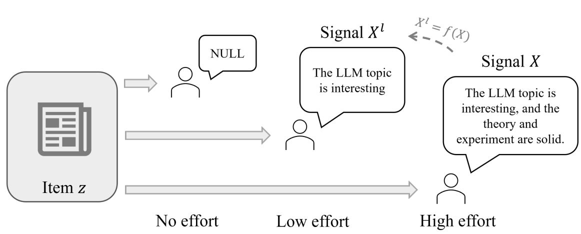


Fig. 3. An Example of the Three-Level Effort Model

Effort Model. We assume that agents can obtain signals with different qualities if they exert different levels of effort. We focus on a three-level effort model: exerting no effort with a cost of 0, low effort with a cost of $c_l \geq 0$, and high effort with a cost of $c_h > c_l$. We use c_i to denote the agent i's cost.

For each agent i, if she exerts high effort, she observes a high-quality signal $X_i = x_i$. If she exerts low effort, she observes a low-quality signal $X_i^l = f(x_i)$, which is part of the high-quality signal. Here, f is a deterministic function, so $\Pr[X_i^l = f(x_i) \mid X_i = x_i] = 1$. However, given X_i^l, the high-quality signal X_i is uncertain. If an agent exerts no effort, they observe an uninformative signal Null all the time. In this case, $\Pr[X = x \mid \text{Null}] = \Pr[X = x]$ for all random variables X. The above model implies that the conditional entropy[2] $H(X_i^l \mid X_i) = 0$, and $H(X_i \mid X_i^l) \geq 0$.

This effort model, called the hierarchical effort model [Kong and Schoenebeck, 2018], suggests that the high-effort signals are strictly more informative than the low-effort signals. For example, in the setting of peer review, a low-effort reviewer might focus solely on surface-level aspects such as the writing quality of the paper. In contrast, a high-effort reviewer can assess not only the writing quality but also other aspects, such as the novelty of the idea, the soundness of the method, and the validity of the experiments.

[2]The conditional entropy $H(X \mid Y)$ measures the average amount of uncertainty in $X \in \mathcal{X}$ given the value of another random variable $Y \in \mathcal{Y}$, i.e., $H(X \mid Y) = -\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} P(x, y) \log\left(\frac{P(x,y)}{P(y)}\right)$.

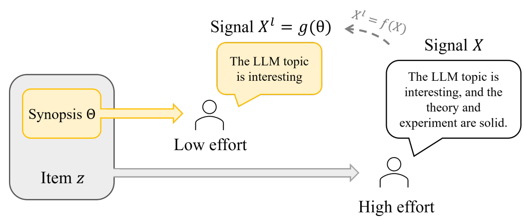


Fig. 4. An Example of the Synopsis-determined Low-effort Signals

Note that our model can also capture a binary effort setting which is assumed in some previous work [Gao et al., 2016, Miller et al., 2005], by assuming $X_i^l = f(x_i) = \text{Null}$ for any $x_i \in \mathcal{X}$. In this setting, the agent either receives an informative signal or a Null signal.

Synopsis. In some applications, we are able to find a commonly known synopsis $\Theta = \theta$ of the item z. We use Σ to denote the space of the synopsis. The synopsis may determine the low-effort signals. For example, in peer review, when a low-effort reviewer writes a review based solely on the introduction of the paper, the paper introduction can be regarded as a commonly known synopsis that fully determines the low-effort signal.

Definition 2.2 (Synopsis-determined Low-effort Signals). *We say the low-effort signal is synopsis-determined if $X_i^l = g(\theta)$ where g is a deterministic function that maps the synopsis to the low-effort signal (Figure 4).*

In some cases, the low-effort signal X_i^l may contain more information about X_i than the synopsis θ, and the synopsis θ does not reveal more information about X_i than X_i^l. In the peer review example, a low-effort reviewer may write a review based on the whole introduction and a glance at the rest of the paper.

Definition 2.3 (Synopsis-covering Low-effort Signals). *We say the low-effort signal is synopsis-covering if $\Pr[X_i = x \mid X_i^l = x_i^l, \Theta = \theta] = \Pr[X_i = x \mid X_i^l = x_i^l]$ for any $x \in \mathcal{X}$.*

Note that Synopsis-determined and Synopsis-covering are not mutually exclusive. We will use these definitions in Corollary 3.6, but Propositions 3.4 and 3.5 do not require them.

Peer Prediction Mechanism and Agents' Incentive. A peer prediction mechanism $M : \mathcal{X}^{|I|} \to \mathbb{R}^{|I|}$ takes all agents' reports as input and outputs a *performance score* s_i to each agent i, i.e., $s_i = M(\tilde{X}_i, \tilde{X}_j)$. Then, agent i is paid according to a linear function $p_i = \alpha \cdot s_i + \beta$, where $\alpha > 0$ and β are constant parameters. The utility of agent i is thereby $u_i = \alpha \cdot s_i + \beta - c_i$.

Each agent aims to maximize her expected utility by choosing a reporting strategy σ_i and an effort c_i. We focus on pure effort strategy c_i and mixed reporting strategy σ_i. Let tuple (σ_i, c_i) be agent i's strategy and $\{(\sigma_i, c_i)\}_{i \in I}$ be the strategy profile of all agents. Under a peer prediction mechanism, an agent's performance score and utility depend on the other agent's strategy. Therefore, we sometimes write agent i's expected utility as a function of the strategy profile of both agents, i.e., $U_i((\sigma_i, c_i), (\sigma_j, c_j))$.

2.2 Mechanism Design Goal

Definition 2.4. *A strategy profile $\{(\sigma_i, c_i)\}_{i \in I}$ is an ϵ-BNE if, for any agent i and for any alternative strategies (σ_i', c_i'), we have:*

$$U_i((\sigma_i, c_i), (\sigma_j, c_j)) \geq U_i((\sigma_i', c_i'), (\sigma_j, c_j)) - \epsilon.$$

In other words, no agent can gain more than ϵ in expected utility by unilaterally deviating from her strategy in an ϵ-BNE. At a high level, our goal is to design a mechanism that (approximately) maximizes an agent's expected performance score if she exerts high effort and reports truthfully. We call such a mechanism (ϵ-)*potent.*

Definition 2.5 (Potent Mechanism)**.** *A peer prediction mechanism M is ϵ-potent if there exists a linear payment scheme with parameters α, β, such that exerting high effort ($c_i = c_h$) and reporting truthfully ($\sigma_i = \tau$) is an ϵ-Bayesian Nash equilibrium.*

2.3 Miller et al. [2005]'s Peer Prediction Mechanism

Miller et al. [2005] propose the first peer prediction mechanism. We refer to this as the original peer prediction mechanism. The original mechanism scores an agent based on how well her report predicts a randomly selected peer's report. They use the log scoring rule (LSR) [Cooke, 1991] to quantify the quality of the prediction.

Definition 2.6 (Log Scoring Rule (LSR))**.** *Given a set of outcomes, $\mathcal{Y}$, and a prediction over the outcomes $p \in \Delta\mathcal{Y}$, the log scoring rule maps the prediction and an outcome $y \in \mathcal{Y}$ to a score* $\mathrm{LSR}(p, y) = \log(p(y))$. *Furthermore, for $q \in \Delta\mathcal{Y}$, let* $\mathrm{LSR}(p, q) = \sum_{y \in \mathcal{Y}} q(y)\,\mathrm{LSR}(p, y)$ *denote the expected log score when the outcome is sampled from the distribution q.*

The Log Scoring Rule is *strictly proper* [Gneiting and Raftery, 2007, Selten, 1998], i.e.,

$$\mathrm{LSR}(p, q) < \mathrm{LSR}(p, p),\ \forall p, q \in \Delta\mathcal{Y} \text{ and } p \neq q,$$

meaning that reporting the true belief of the outcome maximized the score. The idea of Miller et al. [2005]'s peer prediction mechanism is thus very straightforward: scoring agent i based on how well her report can predict her peer's report according to a proper scoring rule.

Definition 2.7 (Original Peer Prediction Mechanism)**.** *Given agent i's report $\tilde{x}_i$, and the peer agent j's report $\tilde{x}_j$. The performance score of agent i is*

$$\mathrm{LSR}(\Pr[X_j \mid X_i = \tilde{x}_i], \tilde{x}_j) = \log \Pr[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i].$$

Proposition 2.8 (Proposition 1 of [Miller et al., 2005])**.** *In the binary effort model, if the common prior π and* $\Pr[X_i \mid Z = z]$ *are known, the above mechanism is potent.*

Intuitively, this mechanism incentivizes effort and truth-telling because only the prediction induced by the informative true signal can maximize the expected score under a strictly proper scoring rule.

3 Peer Prediction Mechanisms for Textual Signals

As discussed, the original peer prediction mechanism requires knowledge about the prior distribution over the items and signals. However, the conditional distribution over signals $\Pr[X_j = x_j \mid X_i = x_i]$ can often be complex and difficult to learn from historical data, especially when eliciting textual signals. To address the textual settings, we propose leveraging Large Language Models (LLMs) to create an estimator of this distribution. We design two new mechanisms — the Generative Peer Prediction Mechanism (GPPM) and Generative Synopsis Peer Prediction Mechanism (GSPPM). Assuming a sufficiently accurate LLM estimator, we theoretically prove that both mechanisms are (approximately) potent and show that GSPPM can better differentiate exerting high effort from exerting low effort by reducing the difference between exerting low effort and exerting no effort.

3.1 Main Assumption: the LLM-Prediction

We first integrate LLMs into the information elicitation model and discuss the key assumption that builds the theoretical foundations of our mechanisms. Given a prompt ψ and a response $\tilde{x}$, a pre-trained LLM can produce a prediction indicating the likelihood of the response being $\tilde{x}$. We denote the distribution of responses generated by an LLM with a prompt ψ as $\Pr_{\mathrm{LLM}(\psi)}$ and refer to it as the *LLM-prediction*. Thus, $\Pr_{\mathrm{LLM}(\psi)}[\tilde{x}]$ denotes the probability that response $\tilde{x}$ is predicted by LLM via prompt ψ.

Sometimes, the prompt itself depends on some input y (e.g., the review from a different agent), in which case we write the prompt as $\psi(y)$. If the input to the prompt itself is a random variable Y, to be consistent with the classic information elicitation model, we use $\Pr_{\mathrm{LLM}(\psi)}[X_j = \cdot \mid Y = y]$ to denote the LLM-prediction $\Pr_{\mathrm{LLM}(\psi(y))}[\cdot]$. We are particularly interested in two predictions. First, $\Pr_{\mathrm{LLM}(\psi)}[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i]$ is the LLM-prediction of agent j's report while integrating agent i's report into the prompt. Second, $\Pr_{\mathrm{LLM}(\psi)}[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i, \Theta = \theta]$ is the LLM-prediction of agent j's report given agent i's report $\tilde{x}_i$ and a commonly known synopsis θ.

Our theoretical results require the following (strong) assumption about the fidelity of the LLM-prediction to that of the model:

Assumption 3.1 (LLM-Prediction). *We assume that for an information elicitation task of interest, there exist sets of prompts $\{\psi(x)\}_{x\in\mathcal{X}}$, $\{\psi'(x,\theta)\}_{x\in\mathcal{X},\theta\in\Sigma}$, and $\epsilon, \epsilon' \geq 0$ such that for any pair of signal $x_i \in \mathcal{X}$ and synopsis $\theta \in \Sigma$:*

$$D_{KL}\left[\Pr[X_j = \cdot \mid X_i = x_i] \,\middle\|\, \Pr_{\mathrm{LLM}(\psi)}[X_j = \cdot \mid X_i = x_i]\right] \leq \epsilon,$$

$$D_{KL}\left[\Pr[X_j = \cdot \mid X_i = x_i, \Theta = \theta] \,\middle\|\, \Pr_{\mathrm{LLM}(\psi')}[X_j = \cdot \mid X_i = x_i, \Theta = \theta]\right] \leq \epsilon',$$

where $D_{KL}[P \parallel Q]$ denotes the KL-divergence[3] *between two distributions. Furthermore, this fact is common knowledge for all agents.*

Assumption 3.1 implies that the prediction of an (idealized) LLM can accurately estimate the underlying information structure of the high-effort signals, allowing us to leverage the vast knowledge embedded within LLMs to predict the probability of a new review x_j given an existing review x_i. Such an LLM provides a data-driven way of computing this distribution. In Section 4, we will detail various *implementations* for computing an approximation to $\Pr_{\mathrm{LLM}(\psi(x_i))}$.

We now turn toward defining our two mechanisms by assuming that we have access to some way of computing the LLM-prediction.

3.2 The Generative Peer Prediction Mechanism (GPPM)

To define our Generative Peer Prediction Mechanism (GPPM), we combine the idea of an LLM prediction with Miller et al.'s mechanism in the textual setting.

Definition 3.2 (Generative Peer Prediction Mechanism (GPPM)). *Given the peer's report $\tilde{x}_j$, the performance score of agent i with report $\tilde{x}_i$ is*

$$\mathrm{LSR}(\Pr_{\mathrm{LLM}(\psi)}[X_j \mid X_i = \tilde{x}_i], \tilde{x}_j) = \log \Pr_{\mathrm{LLM}(\psi)}[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i]$$

If Assumption 3.1 holds and the peer's report $\tilde{x}_j$ is of high effort and truthful, GPPM will be able to successfully differentiate between different effort levels. We defer the formal theory to Section 3.4 while providing the high-level intuitions of why GPPM is potent here.

[3]The KL-divergence between two distributions over the same probability space is $D_{\mathrm{KL}}(P\|Q) = \sum_x P(x)\log\left(P(x)/Q(x)\right)$.

First, GPPM should be able to reward an effortful signal more than a no-effort signal. This is because even a low-effort signal X_i^l helps predict some of the terms in X_j. For example, if X_i^l merely makes it clear that the reviewed paper proposes a new machine learning algorithm, then the words "loss function" and "training" will (correctly) be much more likely.

In addition, we anticipate that the GPPM should be able to reward a high-effort signal more than a low-effort signal. Intuitively, this is because some insights/critiques can only be predicted by high-effort signals. Consider the peer review example again, a low-effort signal may report common features of a machine learning paper but could overlook specific details of the particular paper such as an elegant proof and a potential broader impact of the method.

3.3 The Generative Synopsis Peer Prediction Mechanism (GSPPM)

Intuitively, the above GPPM pays both high and low-effort signals. However, in some applications where low-effort signals may be easily generated by LLMs, such as academic peer review shown in fig. 1, we only want the high-effort signals. This raises the question: Is it possible to further penalize the reporting of low-effort cheap signals?

We propose the Generative Synopsis Peer Prediction Mechanism (GSPPM). The idea is to condition the LLM-prediction on the commonly known synopsis, such as a summary of the reviewed item. As the low-effort signal can be fully characterized by the synopsis, reporting the low-effort signal brings no extra information on predicting agent j's report when the synopsis is conditioned out. In the above peer review example, if the abstract of the reviewed paper has already been inputted into the LLM as prompts, then a low-effort signal simply reiterating the abstract's contents would be redundant.

Definition 3.3 (Generative Synopsis Peer Prediction Mechanism (GSPPM)). *Suppose we have a synopsis θ of the item being reviewed. Given the peer's report being $\tilde{x}_j$, the performance score for agent i with report $\tilde{x}_i$ is*

$$\mathrm{LSR}(\Pr_{\mathrm{LLM}}[X_j \mid X_i = \tilde{x}_i, \Theta = \theta], \tilde{x}_j) = \log \Pr_{\mathrm{LLM}}[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i, \Theta = \theta].$$

In practice, we hypothesize that GSPPM can generally decrease the scores of low-effort signals to that of no-effort signals. This is because the low-effort signal is unlikely to offer additional insights beyond those already present in the synopsis.

For the same reason, we hypothesize that GSPPM can outperform GPPM in distinguishing between low-effort and high-effort signals. Because the scores of low-effort signals are pushed closer to the baseline of no-effort, the reduction in entropy by providing more specific insights that appear in both X_i and X_j should be more salient. In other words, we hypothesize that it will improve the signal-to-noise ratio by making insights more prominent and vocabulary alignment less important. This intuition is very similar to that in Kong and Schoenebeck [2018], which also used conditioning to motivate high-effort signals above low-effort signals.

3.4 Theoretical Results: GPPM and GSPPM are ε-Potent

Here, we provide formal theoretical guarantees of our mechanisms under Assumption 3.1. Due to space constraints, we present only the main propositions here, deferring all lemmas, intuitions, and proofs to Appendix B. For a comprehensive discussion, please refer to the full version.

We first present several important notations before introducing our propositions. We use $I(X_i; X_j)$ to denote the Shannon mutual information [Shannon, 1948] between two signals, X_i and X_j. It provides a quantitative measure of the information shared between them.

$$I(X_i; X_j) = \sum_{x_i, x_j \in \mathcal{X}} \Pr[X_i = x_i, X_j = x_j] \log \frac{\Pr[X_i = x_i, X_j = x_j]}{\Pr[X_i = x_i] \Pr[X_j = x_j]}$$

Furthermore, we use $I(X_i;X_j \mid X_i^l)$ to denote the conditional mutual information.

$$I(X_i;X_j \mid X_i^l)$$
$$= \sum_{X_i^l \in \mathcal{X}} \Pr[X_i^l = x_i^l] \sum_{x_i,x_j \in \mathcal{X}} \Pr[X_i = x_i, X_j = x_j \mid X_i^l = x_i^l] \log \frac{\Pr[X_j = x_j, X_i = x_i \mid X_i^l = x_i^l]}{\Pr[X_i = x_i \mid X_i^l = x_i^l] \Pr[X_j = x_j \mid X_i^l = x_i^l]}$$

Then, we have the following propositions for GPPM and GSPPM respectively.

Proposition 3.4. *For the Generative Peer Prediction Mechanism (GPPM), when Assumption 3.1 holds with parameter $\epsilon \geq 0$. When $I(X_i;X_j \mid X_i^l) > 0$, GPPM is $\alpha\epsilon$-potent, where*

$$\alpha = \max\left(\frac{c_h - c_l}{I(X_i;X_j \mid X_i^l)}, \frac{c_h}{I(X_i;X_j)}\right).$$

Proposition 3.5. *For the Generative Synopsis Peer Prediction Mechanism (GSPPM), when Assumption 3.1 holds with parameter $\epsilon' \geq 0$. When $I(X_i;X_j \mid \Theta, X_i^l) > 0$, GSPPM is $\alpha\epsilon'$-potent, where*

$$\alpha = \max\left(\frac{c_h - c_l}{I(X_i;X_j \mid \Theta, X_i^l)}, \frac{c_h}{I(X_i;X_j \mid \Theta)}\right).$$

These two propositions show that better LLM-prediction approximations will lead to better incentive properties. On the other hand, to achieve any desired incentive property, there's a corresponding threshold for the error of the LLM-prediction approximation.

Additionally, we compare GPPM and GSPPM. Let $\mathrm{Gap}(h, l)$ denote the difference in the expected score of agent i between exerting high effort and exerting low effort while both agents report truthfully. Let $\mathrm{Gap}(h, \mathrm{Null})$ be the analogous notation for the gap between exerting high effort and exerting no effort. We have the following corollary.

Corollary 3.6. *If the low-effort signals are synopsis-determined (definition 2.2) and synopsis-covering (2.3), and $\epsilon' = \epsilon$, $\mathrm{Gap}(h, l)$ has the same lower bound, $I(X_i;X_j \mid \Theta) - \epsilon$, in both GPPM and GSPPM. In contrast, $\mathrm{Gap}(h, \mathrm{Null})$ has a smaller lower bound $I(X_i;X_j \mid \Theta) - \epsilon < I(X_i;X_j) - \epsilon$ in GSPPM than in GPPM.*

Corollary 3.6 suggests that compared with GPPM, GSPPM shrinks the gap between no-effort and low-effort, while preserving the gap between low-effort and high-effort. This property of GSPPM offers two advantages over GPPM: 1) in practice, it is harder for agents to "cheat" the mechanism by submitting a low-effort signal and getting a partial payoff, which consequently further incentivizes high-effort signals; 2) it reduces the noise caused by low-effort signals and produces more reliable scores, better differentiating between low-effort and high-effort.

Although this comparison concerns the lower bounds of the gaps, we will confirm this theoretical insight by empirically showing that GSPPM can distinguish low-effort and high-effort reports better. We demonstrate this by using GSPPM and GPPM to score human-written and LLM-generated reviews in Section 6.4.

4 Estimating the Posterior Prediction via LLMs

Although our mechanisms, at this point, may appear to be straightforward generalizations from prior work, implementing them in practice with textual reports presents distinct challenges. In this section, we present the implementations of our mechanisms, which crucially involve estimating the underlying distribution $\Pr[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i]$ via LLMs. We introduce two heuristic implementation methods, each leveraging the capabilities of the LLM in different ways and degrees.

- The first implementation, denoted as Token, leverages the LLM by directly accessing its output layer to obtain the log probability feedback of the next token, which requires access to the neural network (NN) of the LLM. It has two variants:
  - Token-Raw: We directly use the log probability to predict agents' raw reports.
  - Token-Preprocess: We first use the LLM to preprocess agents' reports and use the log probability to predict the pre-processed reports. The goal is to standardize the language styles and extract essential information.
- The second implementation, denoted as Judgment, uses the LLM to first distill each report into a set of "judgments" and further apply the LLM chatbot to estimate the likelihood of each judgment with textual response. This implementation is particularly useful when it is hard to access the output layer of the LLM, since logprob feedback usually cannot be obtained from commercial LLM APIs, such as GPT-4 Chat Completion.

Note that both implementations can be used for *zero-shot* estimation, meaning that estimating $\Pr[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i]$ only requires $\tilde{x}_i$ and $\tilde{x}_j$ without any historical data. We alternatively offer a non-zero-shot implementation that estimates judgment distributions by clustering historical data. We primarily focus on the zero-shot implementations in the main body, while we defer the discussions of the clustering implementation to Appendix C.

All implementation details including the prompt design, along with example inputs and outputs, are available in the full version.

4.1 Token: Implementation by LLM Token-Prediction

As discussed, the idea of Token is to use the log-probability (*logprob*) feedback of an LLM to predict a given report $\tilde{x}_j$. This is possible because LLMs are fundamentally designed to estimate the likelihood of a subsequent token in a sequence based on a distribution induced by the prompt. This inherent capability is deeply embedded in their pre-training datasets. Consequently, with access to an *open-source* LLM, such as Llama-2 [Touvron et al., 2023] or ChatGLM [Du et al., 2022, Zeng et al., 2023], it becomes feasible to compel the LLM to generate a specific output and report the logprob for each output token.

Formally, we view the textual signal $X = \{X^{(k)}\}_{k\in[n]}$ as a sequence of n tokens, where $X^{(k)}$ denotes the k-th token of X. Again, we denote $\tilde{x}^{(k)}$ as the value of the k-th token in a report $\tilde{x}$.

Given a prompt ψ and the first $k-1$ tokens, an open-source LLM can provide the conditional distribution for the subsequent token via *logprob* feedback, denoted as $\Pr_{\text{LLM}(\psi)}[X^{(k)} \mid X^{(l)} = \tilde{x}^{(l)}\ \forall l \in \{1, 2, \dots, k-1\}]$. With Bayes' rule, we can write the probability of the occurrence of a given report $\tilde{x}$ as:

$$\Pr_{\text{LLM}(\psi)}[X = \tilde{x}] = \Pr_{\text{LLM}(\psi)}\left[X^{(1)} = \tilde{x}^{(1)}\right] \prod_{k=2}^{n} \Pr_{\text{LLM}(\psi)}\left[X^{(k)} = \tilde{x}^{(k)} \,\middle|\, X^{(l)} = \tilde{x}^{(l)}\ \forall l \in \{1, 2, \dots, k-1\}\right].$$

4.1.1 Token-Raw. The most straightforward idea is to integrate the raw report of agent i into the prompt $\psi_{\text{token}}(\tilde{x}_i)$ and compute $\Pr_{\text{LLM}(\psi_{\text{token}}(\tilde{x}_i))}[X = \tilde{x}_j]$ with the log probability output by the LLM. We view this probability as an approximation for $\Pr[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i]$. Finally, agent i's performance score is

$$\text{Token-Raw}(\tilde{x}_i, \tilde{x}_j) = \log \Pr_{\text{LLM}(\psi_{\text{token}}(\tilde{x}_i))}[X = \tilde{x}_j].$$

4.1.2 Token-Preprocess. In practice, agents' raw reports can vary significantly in language style, such as vocabulary usage, sentence structure, and grammatical errors. Additionally, reports may contain superficial information, such as a summary of the paper, in cases of peer review.

Note that even low-quality reviews can be well correlated with superficial information or language style. However, we aim to reward agents based on the quality of the reports' semantics, rather than the correlation on superficial information or language style. Such information may provide "shortcuts" that confound the LLM predictions and consequently lead to unintended rewards. Therefore, we filter out the shortcut information, including language style and superficial information, before applying LLMs to predict the responses.

To address these issues, we propose a simple yet effective preprocessing technique that employs a uniform LLM (It may be the same as or different from the LLM generating logprob) to rephrase the text signal into a pre-set format. Note that the preprocessing step should be tailored for different tasks, considering the trade-off between retaining details and distilling the essential information.

Formally, the performance score of agent i using Token-Preprocess can be defined as:

$$\textsc{Token-Preprocess}(\tilde{x}_i, \tilde{x}_j) = \textsc{Token-Raw}(\textsc{Preprocess}(\tilde{x}_i), \textsc{Preprocess}(\tilde{x}_j)).$$

We will provide evidence comparing Token-Raw and Token-Preprocess and discuss this further in Appendix B.1. Generally, we find that Token-Preprocess is likely to effectively filter out such shortcut information and provide scores according to the semantic quality. Therefore, our main paper will mainly discuss the variant Token-Preprocess.

4.2 Judgment: Implementation by LLM Judgment-Prediction

Accessing the output layer of an LLM can sometimes be impossible, especially when the LLM is not open-source, e.g., the OpenAI GPT-4 API. To address this limitation, we propose an alternative method that uses an LLM as a black box: we first summarizes each report as a set of judgments and then predicts a report by estimating the probability of each judgment and taking the product.

Formally, suppose the set of all possible judgments is $J = \{w_1, w_2, \ldots, w_m\}$. Suppose each signal x (and report $\tilde{x}$) is a subset of judgments, i.e., $x \subset J$. We assume that the event of whether each judgment belongs to a report is independent, thus, we have

$$\Pr[\tilde{X}_j = \tilde{x}_j \mid \tilde{X}_i = \tilde{x}_i] = \prod_{w \in \tilde{x}_j} \Pr\left[w \in \tilde{x}_j \mid \tilde{X}_i = \tilde{x}_i\right] \cdot \prod_{w \in J \setminus \tilde{x}_j} \Pr\left[w \notin \tilde{x}_j \mid \tilde{X}_i = \tilde{x}_i\right].$$

In the zero-shot setting, it is hard to access the full universe of judgments J, making it infeasible to estimate the probability $\Pr\left[w \notin \tilde{x}_j \mid \tilde{X}_i = \tilde{x}_i\right]$. Therefore, we use $\prod_{w \in \tilde{X}_j} \Pr\left[w \in \tilde{X}_j \mid \tilde{X}_i = \tilde{x}_i\right]$ as a heuristic predictor of $\Pr[\tilde{X}_j = \tilde{x}_j \mid \tilde{X}_i = \tilde{x}_i]$.

Furthermore, we discretize the prediction space for LLMs' responses. In particular, given a review $\tilde{x}_i$ and a target judgment w, we ask the LLM to score how much w contradicts or supports $\tilde{x}_i$ with the score ranging from -3 (strong contradiction) to 3 (strong support). We view this score as the gain in log probability, i.e., $\log \Pr\left[w \in \tilde{X}_j \mid \tilde{X}_i = \tilde{x}_i\right] - \log \Pr\left[w \in \tilde{X}_j\right]$.

By taking the sum over the judgments in $\tilde{x}_j$, we obtain an estimate of the conditional log probability, which we use as the performance score, i.e.

$$\textsc{Judgment}(\tilde{x}_i, \tilde{x}_j) = \log \Pr[\tilde{X}_j = \tilde{x}_j \mid \tilde{X}_i = \tilde{x}_i] - \log \Pr[\tilde{X}_j = \tilde{x}_j].$$

Note that the performance score is not exactly the log of the conditional probability (the first term) as defined in definition 3.2. However, subtracting the second term which is independent of agent i's strategy will not disturb the incentive of agent i.

We emphasize that this method is not an accurate estimate of the goal probability $\Pr[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i]$ in general. However, it is a plausible and feasible solution given a black-box LLM. As we will see in the experiments, this heuristic simplification can still capture key information within agents' reports. We note that the performance of our method may be improved with future versions of LLMs that can output more calibrated and accurate predictions.

5 Evaluation

This section presents the methods we use to empirically evaluate the efficacy of our mechanisms. We are primarily interested in testing whether replacing the original high-quality reports with less informative low-quality reports leads to an expected performance score decrease with our mechanisms. We hypothesize that these low-quality reports can be viewed as reports by low-effort agents, and thus, a decrease in performance scores would indicate the empirical effectiveness of our mechanisms in eliciting high effort.

Toward this goal, we first introduce a general workflow and then propose several methods to create low-quality reports, including degrading the original reports and replacing the original reports with LLM-generated fictitious reports, which are regarded as lower quality than the original human expert reports. Note that these methods generating low-quality reports are not necessarily to be restricted in the effort model in Section 2.1 as we focus on testing the efficacy in a realistic scenario without relying on the theoretical assumptions.

Specifically, let z represent an item randomly selected from the dataset. We randomly draw two reports related to z, denoted as $\tilde{x}_i$ and $\tilde{x}_j$. Given the mechanism M, the computed score of agent i is $s_+ := M(\tilde{x}_i, \tilde{x}_j)$. We then replace $\tilde{x}_i$ with a low-quality report $\tilde{x}_i^l$ and recalculate the score $s_- := M(\tilde{x}_i^l, \tilde{x}_j)$. Our hypothesis is that the expected score will decrease.

To test this hypothesis, we repeat the process K times and apply a t-test to evaluate whether the mean decrease in score from s_+ to s_- is statistically significant, thereby confirming the sensitivity of our methods to manipulations. Our empirical distributions (Appendix B.2) show that the measured differences $(s_+ - s_-)$ are approximately normally distributed, confirming our data is suitable for the t-test. Details about the above workflow and the statistical tests are provided in the full version.

5.1 Reports Degradations

We regard the original responses from human agents as truthful and high-effort reports and create three degradation methods, which obviously degrade the information within agents' reports and, consequently, should lead to lower scores. We use these degradations as a "sanity check", implying that any mechanisms that fail to penalize these degradations are unlikely to be useful in practice.

Random Report Replacement We replace a report $\tilde{x}_i$ with a new report $\tilde{x}_{i'}$, which is randomly selected from a different randomly selected item z'. This process is denoted as $\tilde{x}_i^l = \tilde{x}_{i'}$. Note that $\tilde{x}_{i'}$ can also be viewed as a zero-effort signal as the information is likely to be irrelevant. For example, this method corresponds to the behaviors of malicious customers who upload irrelevant reviews in exchange for a restaurant's discount rewards.

Sentence-Level Degradation We degrade original reports by deleting every other sentence.

Judgment-Level Degradation We degrade the list of judgments by deleting every other judgment. This can only be performed with the Token-Preprocess and Judgment implementations, as the preprocessing step has already provided a well-structured list of the judgments.

The sentence-level and judgment-level degradation methods only depend on the agent's report. Thus, they can be viewed as not only creating a low-effort signal but also untruthfully reporting a high-effort signal. The same experiments here (Section 6.2) can also be used to test whether they can incentivize truth-telling.

5.2 LLM-generated Reviews.

Furthermore, we employ the LLMs to create synthetic text reports based on a given item, simulating the scenario of the creation of fictitious academic peer reviews. We conduct the experiment on the ICLR2020 OpenReview dataset[4]. Specifically, we provide the paper z as input to both GPT-3.5 and GPT-4 [Achiam et al., 2023, Brown et al., 2020], requesting them to generate comprehensive reviews following Liang et al. [2023]'s method.

We compare three types of reviews: the human-written review, the GPT-4-generated review, and the GPT-3.5-generated review. Given that GPT-4 is commonly considered a stronger AI than GPT-3.5, and both models are considered worse than human expert reviewers, we hypothesize that we can utilize the LLM-generated reviews to simulate low-quality human reviews, thereby simulating three levels of quality.

It is worth noting that, unlike the degradations discussed in section 5.1, the LLM-generated review does not only depend on the agent's signal x_i since the reviewed paper is input into the LLMs to create reviews. It can not be regarded as untruthfully reporting the original signal. Thus, our focus remains on assessing the effectiveness of differentiating various quality levels of reports across different mechanisms.

6 Experiments

This section presents an overview of the setup of our experiments and the empirical results. We provide more experiment setup details in the full version and all the codes in the Github repo.

6.1 Experiment Setup

6.1.1 Datasets. We use two datasets for our experiment.

- **Yelp Online Review Data (Yelp)** is publicly available online review data from Yelp. We construct our dataset by randomly sampling 1000 items (restaurants) from the entire dataset. For these 1000 items, we have 198,444 text reports (customer reviews) in total, i.e., averaged about 200 reviews per restaurant.
- **ICLR Peer Review Data (ICLR)** includes peer review data from the International Conference on Learning Representations (ICLR) 2020, accessed via the OpenReview API.[5] Given the typically longer and more informative nature of ICLR reviews compared to Yelp reviews, we choose a smaller sample size of 300 items (papers) randomly sampled from the entire ICLR dataset to manage computational demands efficiently. For these 300 items, we have 911 text reports (peer reviews) in total, i.e., averaged about 3 reviews per paper.

The Yelp dataset represents a crowdsourcing setting where reviews are completed by the public. On the other hand, the ICLR dataset can be viewed as an example of expert sourcing, involving reviews provided by experts in a highly specialized field.

6.1.2 LLMs.

- **GPT-4 / GPT-3.5** We employ the gpt-4-1106-preview [Achiam et al., 2023] model for preprocessing the reports on the ICLR dataset and the gpt-3.5-turbo-1106 [Brown et al., 2020] model for preprocessing the reports on the Yelp dataset.[6] For the Judgment implementation, we use gpt-4-1106-preview to predict judgments.
- **Llama-2** For the Token implementations, including both Token-Preprocess and Token-Raw (As discussed in Section 4.1, we defer the results for Token-Raw to Appendix B.1),

[4]As in 2020, the generative AI is not as widely used as now, we assume all the academic peer reviews in the dataset are written by humans.

[5]The reason for using the ICLR 2020 dataset is to exclude the chance that reviewers use LLMs to generate their reports, as we discussed in Section 5, in 2020, AI-generated reviews were rare.

[6]This selection is based on the nature of the texts in each dataset: Yelp reviews tend to be shorter and less complex, thus not requiring the advanced capabilities of a more powerful language model. In contrast, ICLR reviews are more intricate, justifying the use of the higher-capacity gpt-4-1106-preview model for effective rephrasing.

we use the llama-2-70b-chat [Touvron et al., 2023] model with 4-bit quantization to calculate log probabilities for token prediction. The open-source nature of Llama-2 allows for local execution and access to the log probability for each token in a text report.

6.1.3 *Mechanisms.*

GPPM. We test the GPPM on both the Yelp and the ICLR datasets with Token-Preprocess and Judgment implementations. We perform the report degradation evaluation (Section 5.1) and LLM-generated-review evaluation (Section 5.2).

GSPPM. We test the GSPPM on the ICLR dataset. This is because the Yelp dataset lacks detailed features of the items (restaurants) and thus there is not a suitable synopsis of each item. For the ICLR dataset, we consider the abstract of a paper as the commonly-known synopsis. In addition, we are primarily interested in comparing its performance differentiating high-quality and low-quality reports, thus, we only perform the LLM-generated-review evaluation (Section 5.2) to test the GSPPM.

Baseline. Additionally, we present a baseline mechanism (Definition 2.7) that uses only the numerical ratings from reports. This approach rebuilds the joint distribution of two ratings[7] based on historical data and assigns. Since it is not clear how to degrade numerical scores to the same degree as sentence/judgment-level degradations, this baseline is only applicable to the experiment of Random Report Replacement.

6.2 Result 1: GPPM Effectively Penalizes Report Degradations.

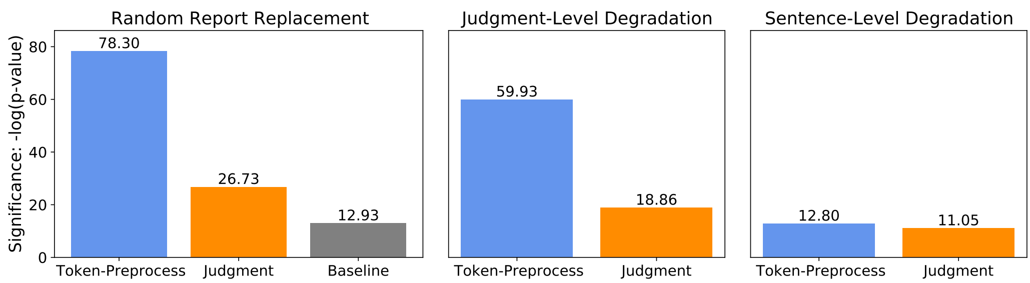
We now delve into our main results. We first describe the results of evaluating the GPPM with three report degradations defined in Section 5.1. We apply sample sizes of $K = 500$ and $K = 1000$ for the experiments on the ICLR and the Yelp datasets respectively. We use both Token-Preprocess and Judgment to compute the estimated conditional probability. We visualize the p-values in Figure 5 and defer the comprehensive statistics metrics to Appendix B (Table 3, 4 and 5).

GPPM significantly outperforms the baseline. We observe a positive $\bar{d}$ in all experiments: for three degradation methods, two datasets, and two implementations. Moreover, in the case of "random report replacement", although all tested mechanisms, including the baseline, exhibit a significance score ($-\log_{10}$(p-value)) well above the threshold of 1.30 (equivalent to p-values < 0.05), the significance score associated with the GPPM are significantly higher compared with the baseline. This observation matches our intuition that there exists a substantially larger amount of information within agents' textual responses, and our GPPM — with either implementation — can successfully extract it.

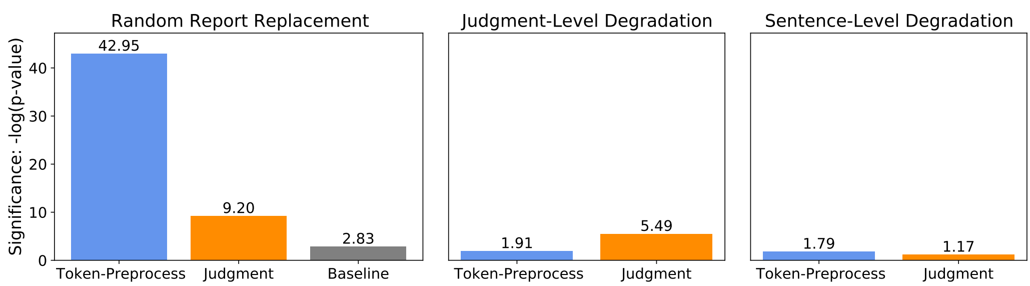
Token outperforms Judgment on the ICLR dataset. Compared with the Judgment implementation, we observe a higher significance score (equivalent to lower p-values) of Token-Preprocess implementations on the ICLR dataset. However, Token-Preprocess does not perform well on the "judgment-level degradation" test conducted on the Yelp dataset.

Additionally, one may be interested in the performance of Judgment with Llama-2. As GPT-4 has better inference capacity than Llama-2, the performance of Judgment with Llama-2 is worse than Judgment with GPT-4. We provide detailed results in Appendix B. Hence, the Token-Preprocess implementation can be considered superior to Judgment, as it consistently outperforms Judgment when both are applied with the same LLM model. However, Judgment is still valuable when there is no access to the LLM's log probability feedback.

[7] There are 4 possible ratings in ICLR dataset and 5 possible ratings in Yelp dataset.



(a) Results on ICLR Dataset



(b) Results on Yelp Dataset

Fig. 5. Report Degradation Evaluation Result for GPPM with Token-preprocess and Judgment Implementations, as well as the Baseline Mechanism: **Significance ($-\log_{10}$(p-value))** of the expected score difference $\bar{d} > 0$, **higher is better**. Typically, a significance score $-\log_{10}(\text{p-value}) > 1.3$ (equivalent to p-value < 0.05) is regarded as significant difference.

6.3 Result 2: Both GPPM and GSPPM Differentiate Three Quality Levels — human, GPT-4, and GPT-3.5

We now show the results of testing the GPPM and GSPPM with LLM-generated reviews introduced in Section 5.2. We focus on the ICLR dataset and use the Token-Preprocess implementation, given that the previous section shows that it is more efficient and performs better than Judgment on the ICLR dataset. We apply a sample size of $K = 500$. We visualize the p-values in Figure 6 and provide the statistics metrics in Table 1.

Mechanism	Review-Generating LLM	$\bar{d}$	$\sigma(d)$	$SE(\bar{d})$	t-statistic	p-value	$-\log_{10}$(p-value)
GPPM	GPT-3.5	10.020	19.290	0.863	11.603	5.0e-28	27.297
	GPT-4	2.904	20.489	0.916	3.166	8.2e-04	3.086
GSPPM	GPT-3.5	9.197	14.173	0.634	14.495	2.7e-40	39.562
	GPT-4	5.357	14.716	0.658	8.131	1.7e-15	14.770

Table 1. Statistics Metrics of LLM-Generated Review Evaluation for GPPM/GSPPM implemented with Token-Preprocess. $\bar{d}$ represents the mean of the score differences, $\sigma(d)$ represents the standard deviation of the score differences, and $SE(\bar{d}) = \sigma(d)/\sqrt{K}$ represents the standard error of the mean difference.

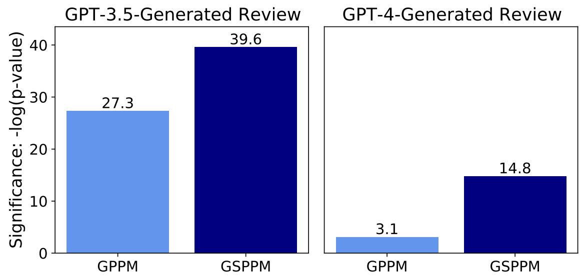


Fig. 6. LLM-generated Review Evaluation Result for GPPM and GSPPM with Token-preprocess Implementation: **Significance ($-\log_{10}$(p-value))** of the expected score difference $\bar{d} > 0$, **higher is better**. Typically, $-\log_{10}$(p-value) > 1.30 (equivalent to p-value < 0.05) is regarded as significant difference.

In Table 1, we observe that, with both mechanisms and the Token-Preprocess implementation, replacing human-written reviews with either GPT-3.5 or GPT-4-generated reviews leads to a statistically significant decrease in scores, as the significance score are all more than the threshold of 1.30 indicating p-values are all less than the threshold of 0.05.

Comparing the first row and the second row of Table 1, the scores are on the same scale as they are computed by the same mechanism. Thus, we observe that while the standard deviation $\sigma(d)$ remains similar, replacing the human-written reviews with GPT-3.5-generated reviews leads to a greater reduction in the expected score, $\bar{d}$, than replacing with GPT-4-generated signals. Similarly, comparing the third row and the fourth row, we have the same observation.

Therefore, with both GPPM and GSPPM, we can observe three score levels, from high to low: human-written, GPT-4-generated, and GPT-3.5-generated. This observation suggests that both our mechanisms can effectively differentiate among these three quality levels. Furthermore, as hypothesized in Section 5.2, GPT-generated reviews can be viewed as low-effort responses. As GPT-4 generates higher-quality reviews than GPT-3.5, we can thus infer three effort levels: high (human-written), medium (GPT-4-generated), and low (GPT-3.5-generated). Therefore, our results also show the effectiveness of our mechanisms in differentiating various effort levels.

6.4 Result 3: GSPPM Penalizes LLM-Generated Peer Peview More than GPPM.

We compare the efficacy of GSPPM and GPPM in differentiating between human-written and LLM-generated reviews. As shown in Corollary 3.6, we expect GSPPM to perform better than GPPM.

Note that the performance scores are no longer on the same scale. Therefore, we focus on the significance scores in Figure 6. We observe that the GSPPM has higher significance scores (lower p-values) in differentiating both GPT-3.5 and GPT-4-generated reviews, indicating its better performance at penalizing the LLM-generated reviews. Furthermore, note that the GPPM merely obtains a p-value of 8.2e-04 for the GPT-4-generated reviews, which is much larger than the GSPPM's p-value under the same condition, indicating a much lower significance.

Therefore, GSPPM has more significant score gaps among these three quality levels–human, GPT-4, and GPT-3.5. This is because much of the superficial information in the LLM-generated reviews is already contained in the synopsis (abstract). Therefore, GSPPM mitigates the ability of the LLMs to obtain a high score. Consequently, the more informative signals that require high effort to access but are necessary for a high-quality report tend to have a higher impact on the score computed by the GSPPM. As mentioned above, we take this as evidence that GSPPM can likely better differentiate high versus low-effort signals and thus better elicit high-effort reports than GPPM in the peer review scenario.

So far, we have been focused on the expected performance scores. However, in practice, the distribution of the performance matters. For example, risk-averse agents may be concerned with the frequency of receiving negative scores, and the variance in performance scores often reflects the fairness of the mechanism. Therefore, we visualize the empirical distribution of $d = s_+ - s_-$, the change of the performance score after applying the degradation in Appendix B.2.

7 Limitation, Future Work, and Conclusion

Limitations. As the first paper, to our best knowledge, exploring eliciting subjective textual data with LLMs, our work has several limitations, and each of them may lead to a future direction.

First, the theoretical effectiveness of the GPPM and GSPPM highly depends on how well the LLM prediction estimates the real underlying distribution (Assumption 3.1). The quality of this prediction can be influenced by many factors, including prompt engineering, the capacity of the LLMs, etc. The LLM predictions generated by state-of-the-art models like GPT-4 and Llama-2 may not offer perfect estimates. Nonetheless, we anticipate improvements in the efficacy of our mechanisms in the future, considering the rapid advancements in LLM technology.

Second, our empirical findings confirm that the GPPM and GSPPM can effectively penalize several degradations of the quality of agents' reports. However, the performance of our mechanisms in addressing more sophisticated manipulation strategies or even malicious strategies remains unstudied. Therefore, how to model and understand human agents' strategies in the textual world is an open question.

Third, we focused on integrating LLMs with Miller et al. [2005]'s mechanism. Future directions may apply our method to a broader set of classic peer prediction mechanisms, especially in the multi-task setting. It might be interesting to test whether the properties of these mechanisms in the classic setting can be generalized to the textual setting. Additionally, fine-tuning LLMs to learn the structure of agent responses in multi-task scenarios might be another promising approach.

Yet another approach is to further explore ways to use LLMs to map text into smaller dimensions. Independent work [Wu and Hartline, 2024], explores using known high-quality texts to determine such a mapping. Their approach shares a similar intuition to our Judgment implementation. Further discussion and comparison are provided in the full version.

Furthermore, in addition to the potent property, prior work has investigated other desiderata, including fairness [Burrell and Schoenebeck, 2021] and efficiency [Xu et al., 2024, Zhang and Schoenebeck, 2023b]. We view the comparisons of different text-elicitation mechanisms in terms of these properties as an interesting future work.

Future Work in Benchmarking LLMs with Peer Review Tasks. Building on our current research, there is significant potential for benchmarking LLMs in peer review tasks to mitigate data contamination and data leakage. More detailed discussions are available in the full version.

Conclusion. In summary, our research introduces a pioneering framework for eliciting high-quality textual judgment. To the best of our knowledge, our work is the first to design peer prediction mechanisms for eliciting high-quality textual reports. We propose two mechanisms, GPPM and GSPPM, which utilize the LLM-derived prediction, two implementations for estimating the LLM-derived prediction, and an evaluation workflow. The use of LLM prediction could extend to other peer prediction mechanisms, given that prediction is the foundation of most peer prediction mechanisms. Our empirical results demonstrate the potential of the GPPM and GSPPM to motivate quality human-written reviews over LLM-generated reviews.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, and Christopher Re. Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Noah Burrell and Grant Schoenebeck. Measurement integrity in peer prediction: A peer assessment case study. *arXiv preprint arXiv:2108.05521*, 2021.

Yi-Chun Chen, Manuel Mueller-Frank, and Mallesh M Pai. The wisdom of the crowd and higher-order beliefs. *arXiv preprint arXiv:2102.02666*, 2021.

Roger Cooke. *Experts in uncertainty: opinion and subjective probability in science.* Oxford University Press, USA, 1991.

Jacques Crémer and Richard P. McLean. Optimal selling strategies under uncertainty for a discriminating monopolist when demands are interdependent. *Econometrica*, 53(2):345–361, 1985. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1911240.

Anirban Dasgupta and Arpita Ghosh. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd international conference on World Wide Web*, pages 319–330, 2013.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022.

Apostolos Filippas, John Joseph Horton, and Joseph Golden. Reputation inflation. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 483–484, 2018.

Alice Gao, James R Wright, and Kevin Leyton-Brown. Incentivizing evaluation via limited access to ground truth: Peer-prediction makes things worse. *arXiv preprint arXiv:1606.07042*, 2016.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Yuqing Kong. Dominantly truthful multi-task peer prediction with a constant number of tasks. In *Proceedings of the fourteenth annual acm-siam symposium on discrete algorithms*, pages 2398–2411. SIAM, 2020.

Yuqing Kong. Dominantly truthful peer prediction mechanisms with a finite number of tasks. *J. ACM*, 71(2), apr 2024. ISSN 0004-5411. doi: 10.1145/3638239. URL https://doi.org/10.1145/3638239.

Yuqing Kong and Grant Schoenebeck. Eliciting expertise without verification. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 195–212, 2018.

Yuqing Kong and Grant Schoenebeck. An information theoretic framework for designing information elicitation mechanisms that reward truth-telling. *ACM Transactions on Economics and Computation (TEAC)*, 7(1):1–33, 2019.

Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Smith, Yian Yin, Daniel McFarland, and James Zou. Can large language models provide useful feedback on research papers? a large-scale empirical analysis, 2023.

Yang Liu, Juntao Wang, and Yiling Chen. Surrogate scoring rules. *ACM Trans. Econ. Comput.*, 10(3), feb 2023. ISSN 2167-8375. doi: 10.1145/3565559. URL https://doi.org/10.1145/3565559.

Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

Drazen Prelec. A bayesian truth serum for subjective data. *science*, 306(5695):462–466, 2004.

Goran Radanovic and Boi Faltings. Incentives for truthful information elicitation of continuous signals. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 770–776. AAAI Press, 2014.

Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems. *Communications of the ACM*, 43 (12):45–48, 2000.

Grant Schoenebeck and Fang-Yi Yu. Learning and strongly truthful multi-task peer prediction: A variational approach. *arXiv preprint arXiv:2009.14730*, 2020.

Grant Schoenebeck and Fang-Yi Yu. Two strongly truthful mechanisms for three heterogeneous agents answering one question. *ACM Transactions on Economics and Computation*, 10(4):1–26, 2023.

Reinhard Selten. Axiomatic characterization of the quadratic scoring rule. *Experimental Economics*, 1:43–61, 1998.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Victor Shnayder, Arpit Agarwal, Rafael Frongillo, and David C Parkes. Informed truthfulness in multi-task peer prediction. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 179–196, 2016.

Siddarth Srinivasan and Jamie Morgenstern. Auctions and peer prediction for scientific peer review. *ArXivorg*, 2021.

Steven Tadelis. Reputation and feedback systems in online platform markets. *Annual Review of Economics*, 8:321–340, 2016.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Jens Witkowski and David C. Parkes. A robust bayesian truth serum for small populations. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI’12, page 1492–1498. AAAI Press, 2012.

Yifan Wu and Jason Hartline. Elicitationgpt: Text elicitation mechanisms via language models. *arXiv preprint arXiv:2406.09363*, 2024.

Shengwei Xu, Yichi Zhang, Paul Resnick, and Grant Schoenebeck. Spot check equivalence: an interpretable metric for information elicitation mechanisms. *arXiv preprint arXiv:2402.13567*, 2024.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. GLM-130b: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=-Aw0rrrPUF.

Yichi Zhang and Grant Schoenebeck. Multitask peer prediction with task-dependent strategies. In *Proceedings of the ACM Web Conference 2023*, WWW ’23, page 3436–3446, New York, NY, USA, 2023a. Association for Computing Machinery. ISBN 9781450394161. doi: 10.1145/3543507.3583292. URL https://doi.org/10.1145/3543507.3583292.

Yichi Zhang and Grant Schoenebeck. High-effort crowds: Limited liability via tournaments. In *Proceedings of the ACM Web Conference 2023*, WWW ’23, page 3467–3477, New York, NY, USA, 2023b. Association for Computing Machinery. ISBN 9781450394161. doi: 10.1145/3543507.3583334. URL https://doi.org/10.1145/3543507.3583334.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. ClusterLLM: Large language models as a guide for text clustering. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13903–13920, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.858. URL https://aclanthology.org/2023.emnlp-main.858.

A Omitted Proofs

We now discuss the potent properties of the GPPM and GSPPM. Intuitively, if a mechanism can lead to a positive gap between the performance scores, by linearly rescaling the performance scores as payments with parameters α and β, it can motivate effort and truth-telling as long as the gap between exerting high effort and exerting low (or zero) effort overcomes the gap between the cost of effort, and consequently, it is potent. Thus, we propose the following propositions.

Proposition A.1. *For the* Generative Peer Prediction Mechanism *(GPPM), when Assumption 3.1 holds with parameter $\epsilon \geq 0$, for any agent i, given the peer agent j exerting high effort and reporting truthfully, any untruthful reporting strategy $\hat{\sigma} \neq \tau$ or effort $c_i \in \{0, c_l\}$ (no-effort or low-effort) implying signal* Null *or X_i^l won't bring more than ϵ score increase. Specifically, we have*

$$\mathbb{E}[GPPM(\tau(X_i), X_j)] - \mathbb{E}[GPPM(\hat{\sigma}(X_i), X_j)] > -\epsilon.$$

(High effort, truthful v.s. High effort, untruthful)

$$\mathbb{E}[GPPM(\tau(X_i), X_j)] - \mathbb{E}[GPPM(\sigma(X_i^l), X_j)] \geq I(X_i; X_j \mid X_i^l) - \epsilon.$$

(High effort, truthful v.s. Low effort, either truthful or untruthful)

$$\mathbb{E}[GPPM(\tau(X_i), X_j)] - \mathbb{E}[GPPM(\sigma(\textsc{Null}), X_j)] \geq I(X_i; X_j) - \epsilon.$$

(High effort, truthful v.s. No effort, either truthful or untruthful)

Note that the mutual information $I(X_i; X_j \mid X_i^l)$ and $I(X_i; X_j)$ are non-negative.

Proposition A.2. *For the* Generative Synopsis Peer Prediction Mechanism *(GSPPM), when Assumption 3.1 holds with parameter $\epsilon' \geq 0$, for any agent i, given the peer agent j exerting high effort and reporting truthfully, any untruthful reporting strategy $\hat{\sigma} \neq \tau$ or effort $c_i \in \{0, c_l\}$ (no-effort or low-effort) implying signal* Null *or X_i^l won't bring more than ϵ score increase. Specifically, we have*

$$\mathbb{E}[GSPPM(\tau(X_i), X_j)] - \mathbb{E}[GSPPM(\hat{\sigma}(X_i), X_j)] > -\epsilon'.$$

(High effort, truthful v.s. High effort, untruthful)

$$\mathbb{E}[GSPPM(\tau(X_i), X_j)] - \mathbb{E}[GSPPM(\sigma(X_i^l), X_j)] \geq I(X_i; X_j \mid \Theta, X_i^l) - \epsilon'.$$

(High effort, truthful v.s. Low effort, either truthful or untruthful)

$$\mathbb{E}[GSPPM(\tau(X_i), X_j)] - \mathbb{E}[GSPPM(\sigma(\textsc{Null}), X_j)] \geq I(X_i; X_j \mid \Theta) - \epsilon'.$$

(High effort, truthful v.s. No effort, either truthful or untruthful)

Note that the mutual information $I(X_i; X_j \mid \Theta, X_i^l)$ and $I(X_i; X_j \mid \Theta)$ are non-negative.

To prove the above results, we analyze agent i's expected scores obtained by each strategy, given that agent j exerts high effort and reports truthfully. When agent i exerts a high effort and reports truthfully, we provide a lower bound of her expected score based on the LLM-Prediction Assumption (Assumption 3.1), which is the negative entropy of agent j's high-effort signal, conditioning on agent i's high-effort signal. For other strategies, we provide an upper bound of agent i's expected score based on the fact that the log scoring rule is proper. The expected score will be at most the the negative entropy of agent j's high-effort signal, conditioning on agent i's signal. Finally, because the high-effort signal is more informative than the low-effort signal, we prove that exerting high effort and reporting truthfully provides approximately the highest expected score. The gap between the negative conditional entropy can be interpreted as mutual information, as stated in the propositions.

Proof of Proposition A.1. For agent i, we analyze her expected score under each strategy given that agent j exerts high effort and reports truthfully.

If agent i also exerts high effort and reports truthfully, her expected score is

$$\begin{aligned}
&\sum_{x_j} \Pr[X_j = x_j \mid X_i = x_i] \log \Pr_{\mathrm{LLM}(\psi)}[X_j = x_j \mid X_i = x_i] && \text{(Assumption 3.1)}\\
\geq &\sum_{x_j} \Pr[X_j = x_j \mid X_i = x_i] \log \Pr[X_j = x_j \mid X_i = x_i] - \epsilon\\
= &- H(X_j \mid X_i = x_i) - \epsilon,
\end{aligned}$$

Taking the expectation of X_i, her expected score is

$$\begin{aligned}
&\sum_{x_i} \Pr[X_i = x_i] \sum_{x_i} \Pr[X_j = x_j \mid X_i = x_i] \log \Pr_{\mathrm{LLM}(\psi)}[X_j = x_j \mid X_i = x_i]\\
\geq &\sum_{x_i} \Pr[X_i = x_i] \left(\sum_{x_j} \Pr[X_j = x_j \mid X_i = x_i] \log \Pr[X_j = x_j \mid X_i = x_i] - \epsilon \right) && \text{(Assumption 3.1)}\\
= &\sum_{x_i, x_j} \Pr[X_i = x_i, X_j = x_j] \log \Pr[X_j = x_j \mid X_i = x_i] - \epsilon\\
= &- H(X_j \mid X_i) - \epsilon,
\end{aligned}$$

where $H(X_j \mid X_i)$ is the conditional entropy.

$$\begin{aligned}
&\sum_{x_i} \Pr[X_i = x_i] \sum_{x_j} \Pr[X_j = x_j \mid X_i = x_i] \log \Pr[X_j = x_j \mid X_i = x_i]\\
= &\sum_{x_i, x_j} \Pr[X_i = x_i, X_j = x_j] \log \Pr[X_j = x_j \mid X_i = x_i]\\
= &- H(X_j \mid X_i),
\end{aligned}$$

If agent i exerts high effort but reports non-truthfully $\sigma \neq \tau$, her expected score is

$$\begin{aligned}
&\sum_{x_j} \Pr[X_j = x_j \mid X_i = x_i] \log \Pr_{\mathrm{LLM}(\psi)}[X_j = x_j \mid X_i = \sigma(x_i)]\\
\leq &\sum_{x_j} \Pr[X_j = x_j \mid X_i = x_i] \log \Pr[X_j = x_j \mid X_i = x_i] && \text{(Log scoring rule is proper.)}\\
= &- H(X_j \mid X_i = x_i).
\end{aligned}$$

With the stochastic relevance assumption and the fact that LSR is strictly proper, the inequality is strict when $x_i \neq \sigma(x_i)$. Notice that there must exist x_i such that $x_i \neq \sigma(x_i)$ with positive probability as $\sigma \neq \tau$. Therefore, taking the expectation of X_i and $\sigma(X_i)$, her expected score is striclty less than $-H(X_j \mid X_i)$. Thus, if agent i also exerts high effort, reporting truthfully will be at least $-\epsilon$ better than reporting non-truthfully.

If agent i exerts low effort and observes X_i^l, and she reports truthfully or non-truthfully with $\sigma(X_i^l)$, then with an analogous derivation, her expected score is

$$\sum_{x_i^l} \Pr[X_i^l = x_i^l] \sum_{x_j} \Pr[X_j = x_j \mid X_i^l = x_i^l] \log \Pr_{\mathrm{LLM}(\psi)}[X_j = x_j \mid X_i = \sigma(x_i^l)] \leq -H(X_j \mid X_i^l).$$

This inequality is based on the fact that Log scoring rule is proper.

Notice that according to our hierarchical effort model, $H(X_j \mid X_i^l) < H(X_j \mid X_i)$ as X_i^l is determined by X_i. Therefore, the difference in the expected scores between exerting high effort and reporting truthfully compared to exerting low effort is at least $H(X_j \mid X_i^l) - H(X_j \mid X_i) - \epsilon = I(X_i; X_j \mid X_i^l) - \epsilon$.

With analogous analyses, the difference in the expected scores from investing in no-effort compared to high-effort is at least $I(X_i; X_j) - \epsilon = H(X_j) - H(X_j \mid X_i) - \epsilon \geq H(X_j \mid X_i^l) - H(X_j \mid X_i) - \epsilon = I(X_i; X_j \mid X_i^l) - \epsilon$.

□

Proof of Proposition A.2. In GSPPM, with analogous analyses, if agent i also exerts high effort and reports truthfully, her expected score is at least $-H(X_j \mid X_i = x_i, \Theta = \theta) - \epsilon'$. Taking the expectation over X_i and Θ, the expected score is $-H(X_j \mid X_i, \Theta) - \epsilon'$. If agent i also exerts high effort and reports non-truthfully, her expected score is strictly less than $-H(X_j \mid X_i = x_i, \Theta = \theta)$. Taking the expectation over X_i and Θ, the expected score is strictly less than $-H(X_j \mid X_i, \Theta)$.

If agent i exerts low effort and observes X_i^l, and she reports truthfully or non-truthfully with $\sigma(X_i^l)$ her expected score is

$$
\begin{aligned}
&\sum_{x_i^l} \Pr[X_i^l = x_i^l] \sum_{\theta} \Pr[\Theta = \theta | X_i^l = x_i^l] \sum_{x_j} \Pr[X_j = x_j \mid X_i^l = x_i^l, \Theta = \theta] \log \Pr_{\mathrm{LLM}(\psi)}[X_j = x_j \mid X_i = \sigma(x_i^l), \Theta = \theta] \\
&= \sum_{x_i^l, \theta} \Pr[X_i^l = x_i^l, \Theta = \theta] \sum_{x_j} \Pr[X_j = x_j \mid X_i^l = x_i^l, \Theta = \theta] \log \Pr_{\mathrm{LLM}(\psi)}[X_j = x_j \mid X_i = \sigma(x_i^l), \Theta = \theta] \\
&\leq \sum_{x_i^l, \theta} \Pr[X_i^l = x_i^l, \Theta = \theta] \sum_{x_j} \Pr[X_j = x_j \mid X_i^l = x_i^l, \Theta = \theta] \log \Pr[X_j = x_j \mid X_i^l = x_i^l, \Theta = \theta] \\
&= - H(X_j \mid X_i^l, \Theta)
\end{aligned}
$$

The inequality is based on the fact that LSR is proper.

Thus, if agent i also exerts low effort, her expected score is at most $-H(X_j \mid X_i^l, \Theta)$. Therefore, the difference in the expected scores between exerting high-effort compared to low-effort is at least $H(X_j \mid X_i^l, \Theta) - H(X_j \mid X_i, \Theta) - \epsilon' = I(X_i; X_j \mid \Theta, X_i^l) - \epsilon'$.

With analogous analyses, the difference in the expected scores from investing in no-effort compared to high-effort is at least $I(X_i; X_j \mid \Theta) - \epsilon' = H(X_j \mid \Theta) - H(X_j \mid X_i, \Theta) - \epsilon' \geq H(X_j \mid X_i^l, \Theta) - H(X_j \mid X_i, \Theta) - \epsilon' = I(X_i; X_j \mid \Theta, X_i^l) - \epsilon'$.

Therefore, whenever $I(X_i; X_j \mid \Theta) > \epsilon'$, exerting high effort and reporting truthfully is an ϵ'-Bayesian Nash equilibrium in GSPPM. □

With the above propositions, we prove Proposition 3.4 and Proposition 3.5.

Proposition 3.4. *For the Generative Peer Prediction Mechanism (GPPM), when Assumption 3.1 holds with parameter $\epsilon \geq 0$. When $I(X_i; X_j \mid X_i^l) > 0$, GPPM is $\alpha\epsilon$-potent, where*

$$
\alpha = \max\left(\frac{c_h - c_l}{I(X_i; X_j \mid X_i^l)}, \frac{c_h}{I(X_i; X_j)}\right).
$$

Proof of Proposition 3.4. Proposition A.1 lower bounds the gap of the *expected performance score* between (truth-telling, high effort) and any strategy of an agent in three cases. To prove that the mechanism is δ-potent, we have to show that there exist constants $\alpha > 0, \beta$ such that the gap of agent's *expected utility* between (truth-telling, high effort) and any strategy in the above three cases is always lower-bounded by $-\delta$.

Suppose agent i plays (σ_i, c_h) where σ_i is an arbitrary reporting strategy. In this case,

$$\begin{aligned}
&U_i((\tau, c_h), (\tau, c_h)) - U_i((\sigma_i, c_h), (\tau, c_h)) \\
=&\alpha \left(\mathbb{E}[\text{GPPM}(X_i, X_j)] - \mathbb{E}[\text{GPPM}(\sigma(X_i), X_j)]\right) \\
\geq& -\alpha\epsilon. && \text{(Proposition A.1)}
\end{aligned}$$

Suppose agent i plays (σ_i, c_l). In this case,

$$\begin{aligned}
&U_i((\tau, c_h), (\tau, c_h)) - U_i((\sigma_i, c_l), (\tau, c_h)) \\
=&\alpha \left(\mathbb{E}[\text{GPPM}(X_i, X_j)] - \mathbb{E}[\text{GPPM}(\sigma(X_i^l), X_j)]\right) - c_h + c_l \\
\geq&\alpha I(X_i; X_j \mid X_i^l) - \alpha\epsilon - (c_h - c_l). && \text{(Proposition A.1)}
\end{aligned}$$

Suppose agent i plays $(\sigma_i, 0)$. In this case,

$$\begin{aligned}
&U_i((\tau, c_h), (\tau, c_h)) - U_i((\sigma_i, 0), (\tau, c_h)) \\
=&\alpha \left(\mathbb{E}[\text{GPPM}(X_i, X_j)] - \mathbb{E}[\text{GPPM}(\sigma(\textsc{Null}), X_j)]\right) - c_h \\
\geq&\alpha I(X_i; X_j) - \alpha\epsilon - c_h. && \text{(Proposition A.1)}
\end{aligned}$$

Take these together, we have that when $\alpha = \max\left\{\frac{c_h - c_l}{I(X_i;X_j|X_i^l)}, \frac{c_h}{I(X_i;X_j)}\right\}$ we can obtain a unified lower bound $\delta = \alpha\epsilon$. This completes the proof.

□

Proposition 3.5. *For the Generative Synopsis Peer Prediction Mechanism (GSPPM), when Assumption 3.1 holds with parameter $\epsilon' \geq 0$. When $I(X_i; X_j \mid \Theta, X_i^l) > 0$, GSPPM is $\alpha\epsilon'$-potent, where*

$$\alpha = \max\left(\frac{c_h - c_l}{I(X_i; X_j \mid \Theta, X_i^l)}, \frac{c_h}{I(X_i; X_j \mid \Theta)}\right).$$

Proof of Proposition 3.5. With analogous analyses of proof of Proposition 3.4, we have Proposition 3.5. □

Corollary 3.6. *If the low-effort signals are synopsis-determined (definition 2.2) and synopsis-covering (2.3), and $\epsilon' = \epsilon$, Gap(h, l) has the same lower bound, $I(X_i; X_j \mid \Theta) - \epsilon$, in both GPPM and GSPPM. In contrast, Gap$(h, Null)$ has a smaller lower bound $I(X_i; X_j \mid \Theta) - \epsilon < I(X_i; X_j) - \epsilon$ in GSPPM than in GPPM.*

Proof of Corollary 3.6. When $X_i^l = X_j^l = g(\Theta)$, $\epsilon = \epsilon'$, given the fact that X_i^l and X_j^l contain partial information of X_i and X_j respectively, and the assumption that the synopsis does not reveal more information about X_i than X_i^l and X_j^l, the statements of the formulas for Gaps in Proposition A.1 and Proposition A.2 directly imply the results. □

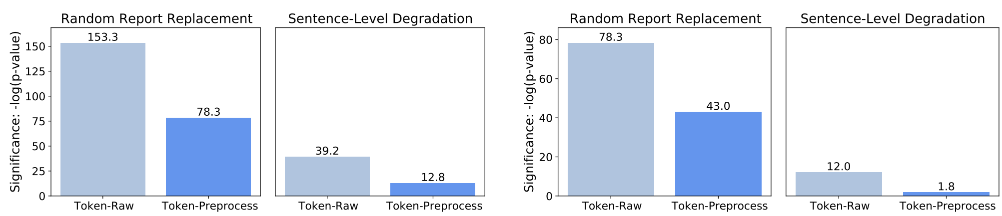
B Additional Results

In this section, we demonstrate the additional results of our experiments, which are omitted in the main text.

B.1 Token-Raw v.s. Token-Preprocess

In this subsection, we delve into the effectiveness of preprocessing by contrasting the performance of Token-Raw and Token-Preprocess. Intuitively, Token-Preprocess provides scores reflecting the semantic quality better since it removes ‘shortcut” information confounding the LLM prediction, such as the paper summary or the reviewer's language style. We present and explain our observations as follows.

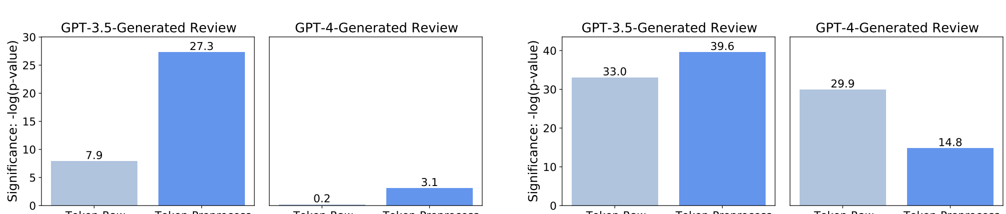
First, we observe that Token-Raw outperforms Token-Preprocess on two degradation tasks (Figure 7). We conjecture that the success of Token-Raw relies on the superficial information commonly found in different reviews of the same item. One example is the paper summary, which most reviewers write before considering the pros and cons. When we use one review to predict tokens in another review of the same item, such common superficial information can significantly increase the performance score. Therefore, as the random replacement degradation and sentence-level degradation remove or reduce the superficial information, Token-Raw can capture such a change more effectively than Token-Preprocess, since the latter removes the superficial information in both reviews.



(a) Results on ICLR Dataset (b) Results on Yelp Dataset

Fig. 7. Report Degradation Evaluation Result for GPPM with Token-raw and Token-preprocess Implementations: **Significance ($-\log_{10}$(p-value))** of the expected score difference $\bar{d} > 0$, **higher is better**. Typically, a significance score $-\log_{10}(\text{p-value}) > 1.30$ (equivalent to p-value < 0.05) is regarded as significant difference. We defer the comprehensive statistics metrics to Table 3 and Table 4.

Our conjecture is further confirmed by the fact that the performance of GPPM implemented with Token-Raw significantly drops when facing more complex tasks like distinguishing GPT reviews from human reviews, especially for GPT-4 generated reviews (Figure 8(a)). As GPT can successfully generate reviews with plausible superficial information, we conjecture this is because the superficial information can successfully fool Token-Raw.



(a) GPPM (b) GSPPM

Fig. 8. Evaluation Result for GPPM and GSPPM with Token-raw and Token-preprocess Implementations in LLM-generated Review: **Significance ($-\log_{10}$(p-value))** of the expected score difference $\bar{d} > 0$, **higher is better**. Typically, a significance score $-\log_{10}(\text{p-value}) > 1.30$ (equivalent to p-value < 0.05) is regarded as significant difference. The comprehensive statistics metrics are in Table 1 and Table 2.

Mechanism	Review-Generating LLM	$\bar{d}$	$\sigma(d)$	$SE(\bar{d})$	t-statistic	p-value	$-\log_{10}$(p-value)
GPPM	GPT-3.5	8.074	31.891	1.426	5.656	1.3e-08	7.884
	GPT-4	-0.511	31.689	1.417	-0.360	6.4e-01	0.193
GSPPM	GPT-3.5	13.776	23.687	1.059	12.992	9.5e-34	33.024
	GPT-4	12.912	23.557	1.054	12.244	1.3e-30	29.898

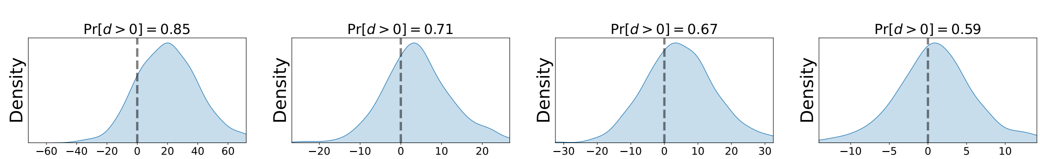
Table 2. Statistics Metrics of LLM-Generated Review Evaluation for GPPM/GSPPM implemented with Token-Raw. Note that GPPM with Token-raw has a p-value $0.64 > 0.05$, indicating a failure in differentiating GPT-4-generated review from human-written review.

Furthermore, when we condition on the synopsis (papers' abstract) using GSPPM, Token-Preprocess can again successfully distinguish GPT-generated reviews from human-written reviews. This is because the superficial information is primarily included in the synopsis, conditioned out by the mechanism. However, in Table 2, we observe that replacing human reviews with GPT-3.5 or GPT-4 reviews leads to a similar expected score decrease, indicating that GSPPM with Token-Raw cannot distinguish GPT-3.5 reviews from GPT-4 reviews as effectively as GSPPM with Token-Preprocess (as shown in Sections 6.3 and 6.4). We conjecture this is because the language style greatly influences LLMs' predictions on raw reviews.[8] As GPT-3.5 and GPT-4 have similar language styles but are very different from human reviewers, Token-Raw fails to separate the two types of GPT-written reviews well though it succeeds in separating GPT reviews from human reviews.

B.2 Empirical Distribution of the Performance Score Change

Here, we use kernel density estimation (KDE)[9] to visualize the empirical distribution of $d = s_+ - s_-$, the change of the performance score after applying the degradation. We observe that the score change follows a bell-shaped distribution. Furthermore, although the score changes are predominantly positive in all the cases, the probability sometimes approaches half, especially in the Yelp dataset. We hypothesize that this is because Yelp reviews tend to be shorter, more diverse, and less standardized, which greatly decreases the quality of the LLM predictions.

We emphasize that a positive score change in expectation is sufficient to incentivize high-quality reports. However, future research may seek to use the performance score as a metric for assessing data quality, in which case it is crucial to minimize the probability of getting a negative score change. We note that fine-tuning an existing LLM (which is not zero-shot) and the development of more advanced language models may contribute to potential improvements.

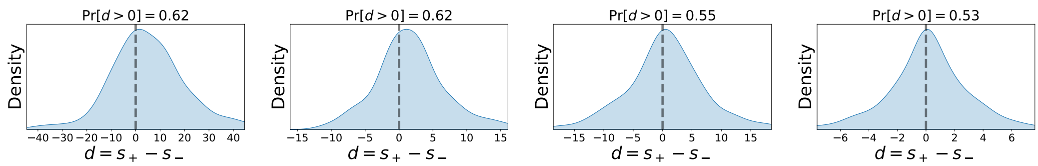


(a) ICLR+Token-Preprocess (b) ICLR + Judgment (c) Yelp + Token-Preprocess (d) Yelp + Judgment

Fig. 9. Empirical Distribution of the Performance Score Change of Random Signal Replacement Evaluation.

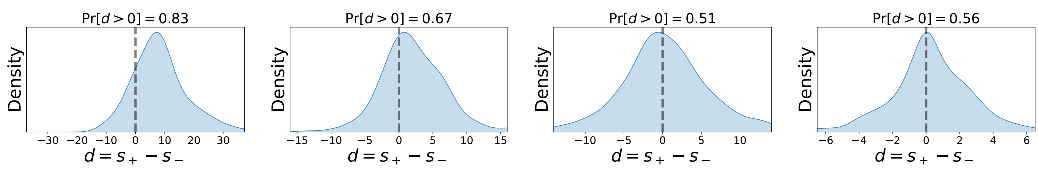
[8]Previous prompt engineering studies have provided evidence that the language style in the prompt can impact LLMs' output [Arora et al., 2022]

[9]We use seaborn.kdeplot (seaborn.pydata.org/generated/seaborn.kdeplot.html) with default parameters to plot the KDEs.



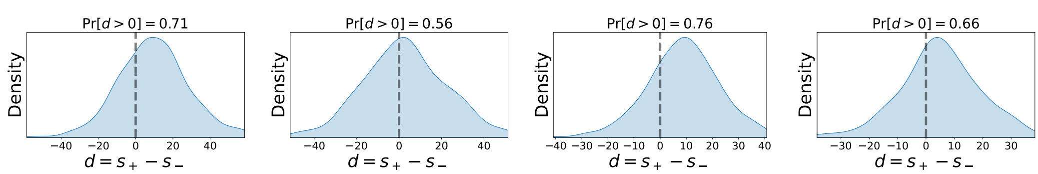
(a) ICLR+Token-Preprocess (b) ICLR + Judgment (c) Yelp + Token-Preprocess (d) Yelp + Judgment

Fig. 10. Empirical Distribution of the Performance Score Change of Sentence-Level Degradation Evaluation.



(a) ICLR+Token-Preprocess (b) ICLR + Judgment (c) Yelp + Token-Preprocess (d) Yelp + Judgment

Fig. 11. Empirical Distribution of the Performance Score Change of Judgment-Level Degradation Evaluation.



(a) GPPM with GPT-3.5-generated review (b) GPPM with GPT-4-generated review (c) GSPPM with GPT-3.5-generated review (d) GSPPM with GPT-4-generated review

Fig. 12. Empirical Distribution of the Performance Score Change of LLM-Generated Review Evaluation. This result corresponds to Table 1.

B.3 Statistics Metrics of Evaluations

In this subsection, we provide all statistics metrics omitted in the main text (Tables 3 to 5).

B.4 Evaluation Results of Implementation Judgment with Llama-2

As we discussed in Section 6.2, here we present the evaluation results for the implementation Judgment with Llama-2 (referred to as Judgement-Llama-2), detailed in Tables 6 and 7. Since Llama-2 is a weaker model compared to GPT-4 and exhibits issues with adherence to prompt instructions, the results are as expected that Judgement-Llama-2 yields worse results in both evaluations.

C Alternative Implementation Based on Clustering

In this section, we explore an alternative implementation named Cluster. This implementation employs clustering of judgments to estimate $\Pr[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i]$. The rationale behind the Cluster implementation is to address situations where LLMs lack specific task knowledge, resulting in inaccurate log probabilities or judgment predictions.

Dataset	Implementation	$\bar{d}$	$\sigma(d)$	$SE(\bar{d})$	t-statistic	p-value	$-\log_{10}$(p-value)
ICLR	Token-Raw	55.444	31.736	1.419	39.026	5.0e-154	153.305
	Token-Preprocess	20.784	20.483	0.916	22.666	5.0e-79	78.298
	Judgment	4.522	8.813	0.394	11.462	1.8e-27	26.735
	baseline	0.207	0.613	0.027	7.532	1.2e-13	12.928
Yelp	Token-Raw	9.975	15.323	0.485	20.576	5.2e-79	78.284
	Token-Preprocess	5.007	10.915	0.345	14.500	1.1e-43	42.955
	Judgment	1.055	5.438	0.172	6.131	6.3e-10	9.202
	baseline	0.018	0.195	0.006	2.982	1.5e-03	2.834

Table 3. Statistics Metrics of Random Report Replacement Evaluation for GPPM. $\bar{d}$ represents the mean of the score differences, $\sigma(d)$ represents the standard deviation of the score differences, and $SE(\bar{d})$ represents the standard error of the mean difference.

Dataset	Implementation	$\bar{d}$	$\sigma(d)$	$SE(\bar{d})$	t-statistic	p-value	$-\log_{10}$(p-value)
ICLR	Token-Raw	11.192	17.337	0.775	14.421	5.8e-40	39.233
	Token-Preprocess	5.117	15.261	0.683	7.490	1.6e-13	12.803
	Judgment	1.662	5.396	0.241	6.880	9.0e-12	11.046
Yelp	Token-Raw	1.778	7.873	0.249	7.137	9.2e-13	12.038
	Token-Preprocess	0.497	7.316	0.231	2.146	1.6e-02	1.794
	Judgment	0.148	3.131	0.099	1.494	6.8e-02	1.169

Table 4. Statistics Metrics of Sentence-Level Degradation Evaluation for GPPM.

Dataset	Implementation	$\bar{d}$	$\sigma(d)$	$SE(\bar{d})$	t-statistic	p-value	$-\log_{10}$(p-value)
ICLR	Token-Preprocess	8.860	10.487	0.469	18.872	1.2e-60	59.928
	Judgment	2.150	5.133	0.230	9.357	1.4e-19	18.857
Yelp	Token-Preprocess	0.399	5.606	0.177	2.252	1.2e-02	1.911
	Judgment	0.362	2.524	0.080	4.534	3.2e-06	5.489

Table 5. Statistics Metrics of Judgment-Level Degradation Evaluation for GPPM.

Dataset	Implementation	$\bar{d}$	$\sigma(d)$	$SE(\bar{d})$	t-statistic	p-value	$-\log_{10}$(p-value)
ICLR	Judgement-Llama-2	0.815	10.634	0.336	2.422	7.9e-03	2.103
Yelp	Judgement-Llama-2	0.513	5.474	0.173	2.962	1.6e-03	2.806

Table 6. Statistics Metrics of Random Report Replacement Evaluation (Judgement-Llama-2)

Dataset	Implementation	$\bar{d}$	$\sigma(d)$	$SE(\bar{d})$	t-statistic	p-value	$-\log_{10}$(p-value)
ICLR	Judgement-Llama-2	-0.246	11.152	0.353	-0.697	7.6e-01	0.121
Yelp	Judgement-Llama-2	0.407	4.451	0.141	2.890	2.0e-03	2.706

Table 7. Statistics Metrics of Sentence-Level Degradation Evaluation (Judgement-Llama-2)

Implementation. We present the pseudocode of the Cluster implementation in Algorithm 1. Similar to the Judgment implementation (Section 4.2), we reload each report $\tilde{x}$ as a set of judgments. In the preparation, we structure the dataset as a list of paired sets of judgments $D = \{(\tilde{x}_{i_t}, \tilde{x}_{j_t})\}_{t\in[N_D]}$, where each pair $(\tilde{x}_{i_t}, \tilde{x}_{j_t})$ indicates two reviews of the same item. Then, we leverage all the judgments in D to fine-tune a short-text embedder, applying the technique introduced by Zhang et al. [2023]. This embedder takes a judgment as input and generates a high-dimensional vector as its embedding, which allows us to employ the Minibatch K-means algorithm to build a clustering structure to classify judgments into a fixed number N_c of clusters[10]. Consequently, We introduce $\tilde{x}[k]$ to denote a binary cluster indicator for the existence of a judgment of cluster $k \in [N_c]$ in $\tilde{x}$, and $X[k]$ as the random variable indicating the existence of a judgment of cluster k in X.

To calculate the score of GPPM, it is necessary to estimate the conditional probability $\Pr[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i]$. We make two assumptions to simplify the estimation.

(1) First, we assume the clusters capture all the information a review has. This means the joint distribution between two reviews can be represented as the joint distribution between all $X_i[k]$ and $X_j[k]$, i.e.,

$$\Pr[X_i = \tilde{x}_i, X_j = \tilde{x}_j] = \Pr\left[X_i[1] = \tilde{x}_i[1], X_j[1] = \tilde{x}_j[1], \ldots, X_i[N_c] = \tilde{x}_i[N_c], X_j[N_c] = \tilde{x}_j[N_c]\right].$$

(2) Second, we assume indicators of different clusters are independent. That is, for any subset $\mathcal{K} \subseteq [N_c]$, such that,

$$\Pr\left[\bigcap_{k\in\mathcal{K}} X_i[k] = \tilde{x}_i[k], X_j[k] = \tilde{x}_j[k]\right] = \prod_{k\in\mathcal{K}} \Pr\left[X_i[k] = \tilde{x}_i[k], X_j[k] = \tilde{x}_j[k]\right]$$

With these two assumptions, we can compute the conditional probability as

$$\Pr[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i] = \frac{\Pr[X_i = \tilde{x}_i, X_j = \tilde{x}_j]}{\Pr[X_i = \tilde{x}_i]} = \frac{\prod_{k\in[N_c]} \Pr[X_i[k] = \tilde{x}_i[k], X_j[k] = \tilde{x}_j[k]]}{\prod_{k\in[N_c]} \Pr[X_i[k] = \tilde{x}_i[k]]}.$$

Furthermore, the numerator can be estimated by the empirical frequency,

$$\Pr[X_i[k] = \tilde{x}_i[k], X_j[k] = \tilde{x}_j[k]] \approx \frac{1}{|D|} \sum_{(\tilde{x}_{i_t}, \tilde{x}_{j_t})\in D} \mathbf{1}[\tilde{x}_{i_t}[k] = \tilde{x}_i[k], \tilde{x}_{j_t}[k] = \tilde{x}_j[k]].$$

Results. Tables 8 and 9 present the performance results of the Cluster implementation. Our results suggest that the performance of Cluster is dominated by Token-Preprocess and Judgment, suggesting a large space of improvement. We hypothesize that this is because the clustering step incurs a significant information loss. The development of better context-specific clustering algorithms can potentially improve the performance of Cluster.

Discussions and Limitations. We acknowledge several limitations with the Cluster implementation. First, it relies on historical data and thus is not zero-shot. Second, without non-trivial adaptations, Cluster is not compatible with the Generative Synopsis Peer Prediction Mechanism. Lastly, it relies on certain (strong) assumptions. In particular, Assumption (1) drops a lot of information within the textual data and only captures the information about what categories of judgments the review contains. Moreover, Assumption (2) overlooks the correlations between clusters, thereby weakening the predictive power of the judgment cluster information. These limitations suggest large spaces for improvements.

[10] We set $N_c = 30$ in the implementation.

ALGORITHM 1: CLUSTER: Peer Prediction Score by Clustering judgments

Initialization: `Initialize`(D);
Input: Review dataset $D = \{(\tilde{x}_{i_t}, \tilde{x}_{j_t})\}_{t \in [N_D]}$
begin
  // Each $(\tilde{x}_{i_t}, \tilde{x}_{j_t})$ contains two reviews of the same item.
  Train a short text embedder based on the set of judgments D;
  $N_c := 30$;
  Map judgments to $1, 2, \ldots, N_c$ using Minibatch K-means clustering with the short text embedder.;
  Denote $\tilde{x}[c]$ as a binary indicator for the existence of a judgment with label c in report $\tilde{x}$;
  for $c = 1$ **to** N_c **do**
    for $b_0 = 0$ **to** 1 **do**
      for $b_1 = 0$ **to** 1 **do**
        $p_{b_0,b_1}(c) := \frac{1}{|D|} \sum_{(\tilde{x}_{i_t}, \tilde{x}_{j_t}) \in D} \mathbf{1}[\tilde{x}_{i_t}[c] = b_0, \tilde{x}_{j_t}[c] = b_1]$;
      end
    end
  end
end

Query of GPPM score: `Query`$(\tilde{x}_i, \tilde{x}_j)$;
Input: textual reports $\tilde{x}_i$ and $\tilde{x}_j$
Output: Score for agent i: CLUSTER$(\tilde{x}_i, \tilde{x}_j)$
begin
  // $score \triangleq \log \Pr[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i]$
  $score := 0$;
  for $c = 1$ **to** N_c **do**
    $score := score + \log \frac{p_{\tilde{x}_i[c], \tilde{x}_j[c]}(c)}{p_{\tilde{x}_i[c],0}(c) + p_{\tilde{x}_i[c],1}(c)}$;
  end
  CLUSTER$(\tilde{x}_i, \tilde{x}_j) := score$;
end

Dataset	Implementation	$\bar{d}$	$\sigma(d)$	$SE(\bar{d})$	t-statistic	p-value	$-\log_{10}$(p-value)
ICLR	CLUSTER	0.084	0.397	0.013	6.659	3.6e-11	10.438
Yelp	CLUSTER	0.482	1.032	0.033	14.774	4.0e-45	44.396

Table 8. Statistics Metrics of Random Report Replacement Evaluation (CLUSTER)

Dataset	Implementation	$\bar{d}$	$\sigma(d)$	$SE(\bar{d})$	t-statistic	p-value	$-\log_{10}$(p-value)
ICLR	CLUSTER	0.015	0.219	0.007	2.203	1.4e-02	1.853
Yelp	CLUSTER	0.044	0.474	0.015	2.955	1.6e-03	2.796

Table 9. Statistics Metrics of Judgment-Level Degradation Evaluation (CLUSTER)