Aligned Information Elicitation for Text

by

Shengwei Xu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Information)
in The University of Michigan
2026

Doctoral Committee:

Professor Grant Schoenebeck, Chair
Professor Elizabeth Bondi-Kelly
Professor Yiling Chen
Professor Qiaozhu Mei

Shengwei Xu
shengwei@umich.edu
ORCID iD: 0000-0002-8058-0820

© Shengwei Xu 2026

ACKNOWLEDGEMENTS

I want to start by thanking my advisor, Professor Grant Schoenebeck. Grant has supported me in so many ways during my PhD—not only in research, but also in career and life more broadly. He was always generous with his time, advice, and encouragement, and I learned a lot from the way he approaches research and mentoring. I feel very lucky to have had him as my advisor.

I sincerely thank my dissertation committee, Professor Elizabeth Bondi-Kelly, Professor Yiling Chen, and Professor Qiaozhu Mei, for their time, feedback, and support. I really appreciate their thoughtful comments and the care they gave to this work.

I also want to thank my undergraduate advisor, Professor Yuqing Kong, as well as Professor Paul Resnick, Professor Tracy Xiao Liu, Professor Zhiyi Huang, Professor Jason Hartline, and other mentors who have guided me along the way. Yuqing first brought me into EconCS research, and both Yuqing and Grant played a big role in helping me find my interests and see the fun in doing research. I still remember the first paper I worked on with them, on League of Legends and Werewolf. Those early experiences made research feel exciting and enjoyable to me, and they had a lasting influence on the path I chose. I also learned a lot from Paul, Tracy, Zhiyi, Jason, and others through their advice, encouragement, and example, and I am very grateful for their help and support.

I am grateful to Yizhi Zhang, Noah Burrell, Biaoshuai Tao, Sanzeed Anwar, David Gamba, and other members of the Useful Theory Innovations Lab for the many conversations, ideas, and lots of help along the way. I also want to thank my cohort and friends at UMich. They made graduate school feel much less lonely and much more fun, and they made Ann Arbor feel like home.

I have been fortunate to work with wonderful peers. I am especially grateful to Zhihuan Huang, Yuxuan Lu, Zhaohua Chen, and Yifan Wu for many thoughtful discussions, joint problem-solving sessions, and the shared excitement of pursuing difficult questions together.

Finally, I owe my deepest thanks to my fiancé, Fangyuan Yang, and my parents, for your love, patience, and unwavering support. Your confidence in me sustained me through the uncertainties and long stretches of doctoral research. This dissertation would not have been possible without you.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

LIST OF FIGURES vi

LIST OF TABLES viii

LIST OF APPENDICES xii

ABSTRACT xiii

CHAPTER

1 Introduction 1

1.1 Contributions 4

2 Spot Check Equivalence: An Interpretable Metric for Information Elicitation Mechanisms 6

2.1 Introduction 6
2.2 Model 9
2.2.1 Information Elicitation Context 9
2.2.2 Motivational Proficiency 12
2.2.3 Measure of Performance Measurements 14
2.2.4 Using Spot-checking as a Reference: Spot Check Equivalence 15
2.3 Unification of Sensitivity and Measurement Integrity 17
2.4 Computing Spot Check Equivalence 18
2.4.1 Computation with Ground Truth 19
2.4.2 Computation without Ground Truth 20
2.5 Effectiveness of the Spot Check Equivalence: Agent-Based Model Experiment 20
2.5.1 Setup 20
2.5.2 Measurement-Integrity-based Spot Check Equivalence. 25
2.5.3 Sensitivity-based Spot Check Equivalence 28
2.5.4 Spot Check Equivalence of Peer Prediction 29
2.6 Does Peer Prediction Make Things Worse? 30
2.6.1 Comparing Spot-checking Mechanisms and Spot-checking Peer-prediction Mechanisms 31
2.7 Related Work 33
2.7.1 Discussion of the truthful report assumption. 34

2.8 Conclusion and Discussion . . . 34

3 Eliciting Informative Text Evaluations with Large Language Models . . . 35

3.1 Introduction . . . 35
3.1.1 Our Contribution . . . 38
3.2 Related Work . . . 40
3.3 Preliminaries . . . 42
3.3.1 Model . . . 42
3.3.2 Mechanism Design Goal . . . 45
3.3.3 Miller et al. [2005]'s Peer Prediction Mechanism . . . 46
3.4 Peer Prediction Mechanisms for Textual Signals . . . 47
3.4.1 Main Assumption: the LLM-Prediction . . . 47
3.4.2 The Generative Peer Prediction Mechanism (GPPM) . . . 48
3.4.3 The Generative Synopsis Peer Prediction Mechanism (GSPPM) . . . 49
3.4.4 Theoretical Results: GPPM and GSPPM are ε-Potent . . . 50
3.5 Estimating the Posterior Prediction via LLMs . . . 52
3.5.1 TOKEN: Implementation by LLM Token-Prediction . . . 53
3.5.2 JUDGMENT: Implementation by LLM Judgment-Prediction . . . 55
3.6 Evaluation . . . 56
3.6.1 Reports Degradations . . . 57
3.6.2 LLM-generated Reviews. . . . 58
3.6.3 Statistics Metrics . . . 59
3.7 Experiments . . . 59
3.7.1 Experiment Setup . . . 60
3.7.2 Result 1: GPPM Effectively Penalizes Report Degradations. . . . 62
3.7.3 Result 2: Both GPPM and GSPPM Differentiate Three Quality Levels — human, GPT-4, and GPT-3.5 . . . 63
3.7.4 Result 3: GSPPM Penalizes LLM-Generated Peer Review More than GPPM. . . . 65
3.7.5 Empirical Distribution of the Performance Score Change . . . 65
3.8 Limitation, Future Work, and Conclusion . . . 66

4 Benchmarking LLMs' Judgments with No Gold Standard . . . 70

4.1 Introduction . . . 70
4.1.1 Our Contributions . . . 71
4.1.2 Related Work . . . 73
4.2 Preliminaries . . . 74
4.3 Generative Estimator for Mutual Information (GEM) . . . 76
4.3.1 Theoretical Guarantees: Benchmarking LLMs with No Gold Standard 76
4.3.2 Filtering Out Shortcuts . . . 77
4.4 Experiment: Validating GEM's Effectiveness . . . 79
4.4.1 Result 1: Positive Correlation with Human Annotation . . . 80
4.4.2 Result 2: Better Sensitivity to Degradation . . . 82
4.4.3 Result 3: Better Robustness against Manipulation . . . 83
4.5 GRE-bench: Benchmarking LLMs with GEM . . . 84

4.6 Conclusion and Discussion . . . 87

5 Beyond Correlation: Systematically Designing and Testing Aligned Text Evaluation Metrics with f-Mutual Information . . . 89

5.1 Introduction . . . 89
5.2 Problem Setup . . . 91
5.3 Test Principles . . . 92
5.4 f-MI Metric Design Framework . . . 93
5.4.1 f-Mutual Information . . . 94
5.4.2 MI Estimation . . . 96
5.4.3 Representation . . . 99
5.4.4 Prediction Mechanism . . . 100
5.4.5 Prior Work as Instances of the Framework . . . 101
5.5 Experiment . . . 102
5.5.1 Experiment Design . . . 103
5.5.2 Results . . . 103
5.6 Related Work . . . 108

6 Conclusion and Future Directions . . . 110

6.1 Future Directions . . . 111

APPENDICES . . . 113

BIBLIOGRAPHY . . . 171

LIST OF FIGURES

FIGURE

2.1 Information Elicitation Context 9
2.2 An Agent's Perspective of an Information Elicitation Context 11
2.3 Theoretical Analysis Overview: Spot Check Equivalence based on Measurement Integrity, Sensitivity and motivational proficiency (negative expected total payment) are equivalent. 17
2.4 Measurement Integrity v.s Total Payment of Borda-count payment scheme: the x-axis is the Measurement Integrity and the y-axis is the total payment needed to elicit that equilibrium within the tournament payment scheme. The horizontal line shows the agents' cost to exert the effort level, which implies the minimal payment to satisfy Individual Rationality. 26
2.5 Convergence speed of the Measurement Integrity and the total payment: the x-axis is the number of the samples, and the y-axis is the estimated Measurement Integrity and the estimated total payment of the Borda-count payment scheme at effort level $\xi = 0.6$ respectively. 27
2.6 Sensitivity v.s Total Payment of Borda-count payment scheme: the x-axis is the $\Delta\mu/\sigma$ and the y-axis is the total payment needed to elicit that equilibrium within the tournament payment scheme. The horizontal line shows the agents' cost to exert the effort level, which implies the minimal payment to satisfy the Individual Rationality. 28
2.7 Spot Check Equivalence on different effort levels 29
2.8 #tasks per agent vs. the Spot Check Equivalence based on Measurement Integrity: the x-axis is the number of tasks assigned to each agent and the y-axis is the Spot Check Equivalence calculated by Measurement Integrity. 30

3.1 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. 36
3.2 An Overview of Our Information Elicitation Model 43
3.3 An Example of the Three-Level Effort Model 43
3.4 An Example of the Synopsis-determined Low-effort Signals 45
3.5 A simplified example of a prompt for TOKEN in academic peer review scenario. In Appendix B.4, we present all exact prompts used in our experiment. 54

3.6 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 $-\log_{10}$(p-value) > 1.3 (equivalent to p-value < 0.05) is regarded as significant difference. 63
3.7 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, a significance score $-\log_{10}$(p-value) > 1.30 (equivalent to p-value < 0.05) is regarded as significant difference. 64
3.8 Empirical Distribution of the Performance Score Change of Random Signal Replacement Evaluation. 66
3.9 Empirical Distribution of the Performance Score Change of Sentence-Level Degradation Evaluation. 66
3.10 Empirical Distribution of the Performance Score Change of Judgment-Level Degradation Evaluation. 67
3.11 Empirical Distribution of the Performance Score Change of LLM-Generated Review Evaluation. This result corresponds to Table 3.3. 67

4.1 Our model 74
4.2 Example of Hierarchical Information Structure in Peer Reviews 78
4.3 An Overview of Our Generative Estimator for Mutual Information 78
4.4 Meaningless Elongation 84
4.5 Results of GRE-bench based on three evaluation metrics with 90% confidence intervals vs. model parameter sizes[1]indicated by color. The grey line represents the average human baseline, with the 90% confidence interval shaded in grey. . 85

B.1 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 $-\log_{10}$(p-value) > 1.30 (equivalent to p-value < 0.05) is regarded as significant difference. We defer the comprehensive statistics metrics to Table B.2 and Table B.3. 123
B.2 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 $-\log_{10}$(p-value) > 1.30 (equivalent to p-value < 0.05) is regarded as significant difference. The comprehensive statistics metrics are in Table 3.3 and Table B.1. 123

C.1 Results of GRE-bench based on finetuned GEM and GEM-S with 90% confidence intervals vs. model parameter size. The grey line represents the average human baseline, with the 90% confidence interval shaded in grey. 161

LIST OF TABLES

TABLE

2.1 IEC parameter setups for ABM simulations 22
2.2 f-Mutual Information 24

3.1 A high-level comparison of three implementations. 53
3.2 Table illustrating the use of different LLM models in various tasks 61
3.3 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. 64

4.1 Spearman's correlation coefficient (ρ) and p-values between various evaluation metrics and instructor-annotated grades. Significant positive correlations ($p < 0.05$) are bolded. 81
4.2 Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after degradations and manipulations of various evaluation metrics. Significant score decreases ($p < 0.05$) after degradations are highlighted in bold green, implying the evaluation metric can effectively penalize the degradation. Significant score increases ($p < 0.05$) after manipulations are highlighted in bold red, implying the evaluation metric is not robust against the manipulation. We provide 95% confidence interval of SMD values in Appendix C.4.1. 83
4.3 Spearman's correlation coefficients between GRE-bench and other popular benchmarks. 86

5.1 Common f-divergences, along with a convenient population-optimal critic $T^*(x, y)$ and the corresponding conjugate term $f^*(T^*(x, y))$, written in terms of the density ratio $r(x, y)$. 98
5.2 Sample-level score and quantity to estimate for each (F, E) pair. 99
5.3 Existing methods as instances of the design tuple (F, E, R, Π). 101
5.4 Overview of evaluation datasets. n is the number of candidate reports; #ref is the typical number of positive references per task. "Human rating" indicates whether item-level human quality annotations are available for candidate reports. 103
5.5 Coverage of the design space. Each cell is labeled by the name of the corresponding scoring method, where applicable. NEW denotes combinations introduced or evaluated in this work. 104

5.6 Human-rating correlation results. Rows are scoring methods, grouped by distribution model D and its granularity; within each group, rows sweep the information measure $F \in \{\mathrm{KL}, \mathrm{TV}\}$ and estimator $E \in \{\text{Direct}, f\text{-var.}\}$. Columns are datasets; entries are Spearman rank correlations with human ratings. For SummEval we report the mean over all dimensions. Best result within each group is bolded; best among f-MI metrics per column and best overall per column are underlined. 105
5.7 Sensitivity to degradation. For each (metric, dataset) pair, we apply m degradation strategies (header) and count those on which the metric fails to produce a significant score decrease at $p < 0.05$; lower is better. A ✓ marks zero failures; otherwise, letter codes mark the strategies the metric missed (legend below). . . 106
5.8 Manipulation resistance. For each (metric, dataset) pair, we apply m manipulation strategies (header) and count those the metric fails to resist, i.e., that produce a significant score increase at $p < 0.05$; lower is better. A ✓ marks zero failures; otherwise, letter codes flag the strategies the metric failed to resist (legend below). 106

B.1 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. 124
B.2 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. 124
B.3 Statistics Metrics of Sentence-Level Degradation Evaluation for GPPM. 124
B.4 Statistics Metrics of Judgment-Level Degradation Evaluation for GPPM. 124
B.5 Statistics Metrics of Random Report Replacement Evaluation (Judgement-Llama-2) 125
B.6 Statistics Metrics of Sentence-Level Degradation Evaluation (Judgement-Llama-2) 125
B.7 Statistics Metrics of Random Report Replacement Evaluation (Cluster) . . . 128
B.8 Statistics Metrics of Judgment-Level Degradation Evaluation (Cluster) 128

C.1 Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after degradations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score decreases ($p < 0.05$) after degradations are highlighted in bold green, implying the evaluation metric can effectively penalize the degradation. 149
C.2 Standardized Mean Differences (SMD) of scores $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ after manipulations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score increase ($p < 0.05$) after manipulations are highlighted in bold red, implying the evaluation metric are not robust against the manipulation. 150
C.3 Spearman's correlation coefficients between GRE-bench and other popular benchmarks. 151

C.4 Results of GRE-bench based on GEM, GEM-S (abstract), and GEM-S (ASSW) with Llama3.1 70B. In the column of parameter size, the symbol $\sim$ indicates that this size is estimated. For mix-of-expert (MoE) models, we show the total parameter size instead of the activated parameter size. 152
C.5 Results of GRE-bench based on GEM, GEM-S (abstract), and GEM-S (ASSW) with Llama3.1 8B. In the column of parameter size, the symbol $\sim$ indicates that this size is estimated. For mix-of-expert (MoE) models, we show the total parameter size instead of activated parameter size. 153
C.6 Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after degradations of GEM and GEM-S metrics (preprocessing with Llama-3.2 90B) with 95% confidence intervals in parentheses. Significant score decreases ($p < 0.05$) after degradations are highlighted in bold green, implying the evaluation metric can effectively penalize the degradation. 154
C.7 Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after manipulations of GEM and GEM-S metrics (preprocessing with Llama-3.2 90B) with 95% confidence intervals in parentheses. Significant score increases ($p < 0.05$) after manipulations are highlighted in bold red, implying the evaluation metrics are not robust against the manipulation. 154
C.8 Results of GRE-bench based on GEM, GEM-S (abstract), and GEM-S (ASSW) with Llama3.2 90B preprocessing. In the column of parameter size, the symbol $\sim$ indicates that this size is estimated. 155
C.9 Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after degradations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score decreases ($p < 0.05$) after degradations are highlighted in bold green, implying the evaluation metric can effectively penalize the degradation. 157
C.10 Standardized Mean Differences (SMD) of scores $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ after manipulations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score increases ($p < 0.05$) after manipulations are highlighted in bold red, implying the evaluation metrics are not robust against the manipulation. 158
C.11 Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after degradations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score decreases ($p < 0.05$) after degradations are highlighted in bold green, implying the evaluation metric can effectively penalize the degradation. 160
C.12 Standardized Mean Differences (SMD) of scores $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ after manipulations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score increase ($p < 0.05$) after manipulations are highlighted in bold red, implying the evaluation metric are not robust against the manipulation. 160

D.1 Degradation strategies. 162
D.2 Manipulation strategies. 163
D.3 Degradation perturbation method results. Each cell reports paired standardized change $d_z \pm$ 95% CI half-width. Red values mark failed degradation criteria ($p < 0.05$). 167

D.4 Manipulation perturbation method results. Each cell reports paired standardized change $d_z \pm$ 95% CI half-width. Red values mark failed manipulation criteria ($p < 0.05$). 168

LIST OF APPENDICES

APPENDIX

A Proofs for Spot Check Equivalence 113

B Appendix for Generative Peer Prediction 117

C Appendix for Generative Mutual Information 137

D Appendix for Aligned Scoring Mechanisms 162

ABSTRACT

Automated evaluation of textual information is central across the AI lifecycle, from collecting training data to benchmarking model outputs. The core goal is alignment with underlying quality: a higher score should mean a genuinely better report. In this dissertation, the quality of interest is semantic informativeness—whether a report conveys useful, truthful, task-relevant information.

Achieving this goal requires two forms of alignment. Statistical alignment means that scores correlate with human-annotated quality ratings, a property often studied in natural language generation (NLG) evaluation. Strategic alignment means that untruthful or low-effort reports receive lower expected scores than truthful, informative ones, a property often studied in information elicitation.

Conventional reference-based scoring pursues alignment by comparing a candidate response to a gold-standard reference—a trusted answer, expert judgment, or canonical output. Yet many important tasks lack such references. In open-ended settings like academic peer review, free-form explanation, and subjective judgment, gold-standard references may be unavailable or prohibitively expensive to obtain at scale.

This dissertation studies how to elicit and evaluate textual information without gold-standard references. Its central claim is that aligned elicitation and evaluation without references requires strategic and statistical alignment to be developed jointly—a metric that correlates with human judgments but can be gamed is unreliable, and a mechanism that discourages bad strategies but whose ex-post scores do not track quality is not an adequate evaluator. The dissertation therefore combines ideas from information elicitation, information theory, and large language models to design aligned metrics and mechanisms for text when gold-standard references are unavailable.

The dissertation makes four contributions. First, I introduce Spot Check Equivalence (SCE), an interpretable metric for comparing information-elicitation mechanisms. Using SCE, I show that strategic and statistical alignment are often closely coupled: in some settings, high statistical alignment implies high strategic alignment. Second, I extend peer prediction from categorical and numerical signals to text, proposing the Generative Peer Prediction Mechanism (GPPM)—a strategically aligned mechanism for eliciting truthful textual information. Third, I adapt GPPM into the Generative Estimator for Mutual

Information (GEM), an NLG evaluation metric in which a targeted modification improves statistical alignment while preserving strategic alignment. Building on GEM, I propose GRE-bench, which benchmarks LLMs' ability to write peer reviews using a continuously refreshed stream of open-access review data, helping mitigate data contamination. Fourth, generalizing these mechanisms, I develop a systematic design-and-test framework for aligned text evaluation metrics based on f-mutual information. The design framework unifies prior methods and reveals a broader design space, while the test framework evaluates metrics across multiple principles and tasks, enabling controlled factorial comparison of existing and new methods.

Together, these contributions bridge information-elicitation theory and modern language-model evaluation, showing how automated text evaluation can be made more reliable when gold-standard references are unavailable.

CHAPTER 1

Introduction

Automated evaluation of information quality is becoming a central challenge for modern AI. Across the AI lifecycle, from collecting training data, to training models, to benchmarking model outputs, researchers and practitioners need ways to assess whether the textual information they rely on is actually informative and reliable. When such assessment is missing or noisy, systems may be trained on poor data, models may appear strong on paper while failing in practice, and downstream scientific, business, or policy decisions may be based on unreliable signals.

This problem has become more important with the rise of large language models (LLMs). Synthetic LLM-generated text makes simple filters and heuristic quality checks far less effective for data collection. Recent work shows that training on recursively generated or low-quality data may lead to model degradation [Shumailov et al., 2024]. At the same time, AI development is increasingly expensive and high-stakes, so evaluation procedures must be not only accurate, but also resistant to gaming once they become optimization targets.

A central obstacle is that, in many important settings, gold-standard references do not exist or are prohibitively expensive to obtain. One can ask crowd workers to label data or judge model outputs, but ensuring the reliability of those judgments is itself a problem of ensuring data quality. Moreover, many tasks are inherently non-verifiable: there may be no single correct answer for a reference to capture, as in open-ended explanation, creative generation, or academic peer review.

This dissertation studies how to **elicit and evaluate textual information without gold-standard references**. The central objective is to align evaluation scores with underlying quality, where quality, throughout this dissertation, is understood in terms of *semantic informativeness*, i.e., whether a report conveys useful, truthful, task-relevant information.

Two research traditions have approached this problem from different directions, each emphasizing a different notion of alignment:

- **Strategic alignment.** The information-elicitation literature designs peer-prediction and related mechanisms in which truthful reporting is incentivized even when ground truth is unavailable [Miller et al., 2005, Dasgupta and Ghosh, 2013, Kong and Schoenebeck, 2019, Kong, 2020, 2024b]. The core idea is to use correlations among peer reports to reward informative, truthful reports and reduce incentives to manipulate. In this dissertation, I use the term strategic alignment to emphasize that the scoring rule aligns participants' incentives with honest and effortful reporting. Some mechanisms provide truthfulness in equilibrium, while later work establishes stronger dominant-truthfulness guarantees: any untruthful strategy, which is less preferred, yields a lower expected score [Kong, 2020, 2024b].

- **Statistical alignment.** The natural language generation (NLG) evaluation literature develops metrics intended to agree with human judgments on benchmark datasets, and I refer to this property as statistical alignment. In this sense, a metric is statistically aligned when it tracks human ratings on observed data samples.

Although these traditions address closely related problems, interestingly, they have largely remained separate. As Burrell and Schoenebeck [2021] point out, the information elicitation literature has focused primarily on strategic alignment, with far less attention to the statistical alignment of the resulting scores, despite its importance for fairness. Conversely, in NLG evaluation, high correlation on a fixed benchmark is insufficient: it does not guarantee robustness under distribution shift or strategic behavior. Recent evidence demonstrates that LLM-as-a-Judge can be gamed: applying superficial tricks can inflate scores without improving underlying quality [Xu et al., 2025, Raina et al., 2024]. While LLMs themselves are non-strategic entities, the benchmarks provide quantitative targets that create incentives for engineers developing new models. This dynamic exemplifies Goodhart's Law: "When a measure becomes a target, it ceases to be a good measure."

Strategic alignment also connects to the core question of generalization in machine learning: alignment should persist when the distribution shifts. The phenomenon of *shortcut learning* [Geirhos et al., 2020] provides a compelling example, models that exploit spurious correlations in training data fail to generalize, analogous to how agents might exploit loopholes in evaluation metrics that lack strategic alignment.

A further challenge is that existing information-elicitation mechanisms do not directly generalize to textual reports. Classical peer prediction mechanisms rely on the joint distribution of agents and peers' private signals. However, previous methods to estimate this joint distribution, such as learning the joint distribution from multiple tasks [Shnayder et al., 2016] or directly eliciting second-order predictions [Prelec, 2004], only work on structured categori-

cal or numerical responses. Text, by contrast, is rich, high-dimensional, and semantically nuanced, which creates new challenges for information elicitation and evaluation.

Thesis statement. This dissertation argues that strategic and statistical alignment should not be treated in isolation. Aligned elicitation or evaluation of textual information without gold-standard references requires both: scores should encourage honest, informative behavior, and they should also track underlying quality in the data distributions that matter. To this end, I develop methods that combine ideas from information elicitation theory with tools from large language models to assess semantic informativeness in settings where references are missing, noisy, or themselves of unverified quality.

Scope and relationship to broader evaluation goals. The form of quality studied in this dissertation is what I call semantic informativeness: whether an evaluation score faithfully reflects how much useful, truthful, task-relevant information a report conveys. Semantic informativeness is only one dimension of evaluation, not a complete account of all socially important properties of AI systems.

Recent work emphasizes broader desiderata: Holistic evaluation frameworks consider dimensions such as accuracy, calibration, robustness, fairness, bias, toxicity, and efficiency [Liang et al., 2022]. Alignment research on general-purpose assistants emphasizes helpfulness, honesty, and harmlessness [Askell et al., 2021]. Truthfulness benchmarks focus specifically on factual honesty [Lin et al., 2022]. Fairness research shows that fairness is not a single property, but a family of desiderata tied to different harms, affected groups, and deployment contexts [Gallegos et al., 2024]. Pluralism asks whether a model can represent, preserve, or be calibrated to multiple legitimate human perspectives rather than collapsing them into a single consensus [Sorensen et al., 2024]. All these concerns are important, and this dissertation does not aim to provide a universal metric for all of them.

The narrower focus on semantic informativeness is nonetheless one of the foundational targets. For many knowledge-intensive tasks, informativeness is a precondition for usefulness: a response cannot be genuinely helpful if it carries little or no task-relevant signal. Likewise, mechanisms that strategically reward truthful, effortful reporting can help discourage random guessing and some forms of hallucination [Wu and Hartline, 2024]. At the same time, pluralism reminds us that disagreement is not always noise, especially in open-ended tasks such as peer review, where different stakeholders may reasonably prioritize different values. The methods in this dissertation therefore address a specific core property while leaving room for future integration with broader evaluation goals.

1.1 Contributions

This dissertation makes four main contributions.

Contribution 1: Interpretable metric for information elicitation mechanisms. First, I study statistical alignment within the general information elicitation setting. I propose an interpretable metric for information elicitation mechanisms, named Spot Check Equivalence (SCE), which measures the performance of a mechanism by comparing it to spot checking (random verification against ground truth).

Beyond truthfulness, previous literature has proposed at least three metrics to compare the performances of information elicitation mechanisms [Zhang and Schoenebeck, 2023b, Gao et al., 2016, Burrell and Schoenebeck, 2021]. However, different metrics lead to divergent and even contradictory results in various contexts. By introducing Spot Check Equivalence, I harmonize these divergent stories, showing that two of these metrics, sensitivity (related to the payment efficiency) [Zhang and Schoenebeck, 2023b] and measurement integrity (related to ex-post fairness) [Burrell and Schoenebeck, 2021], are actually the same within certain contexts, both capturing different aspects of the impact of statistical alignment. Furthermore, I explain why the third metric [Gao et al., 2016] diverges.

Contribution 2: Strategically-aligned mechanism for textual information elicitation. Second, I design strategically-aligned (i.e., truthful) information elicitation mechanisms specifically for textual information. I 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 their peer's report.

Theoretically, I show that when the LLM prediction is sufficiently accurate, the mechanisms can incentivize high effort and truth-telling as an (approximate) Bayesian Nash equilibrium. Empirically, I confirm the efficacy of the mechanisms through experiments conducted on two real datasets: the Yelp review dataset and the ICLR OpenReview dataset.

Contribution 3: Dual-aligned evaluation metrics for NLG evaluation. Third, I design evaluation metrics for NLG evaluation to achieve both strategic and statistical alignment without requiring gold-standard references. Building upon GPPM, which already ensures strategic alignment, I introduce a simple but necessary modification to improve statistical alignment as well, making it more suitable for NLG evaluation problems.

I propose GEM (Generative Estimator for Mutual Information), which uses a generative model to estimate mutual information between candidate and reference responses. In

experiments on a human-annotated dataset, GEM demonstrates competitive correlations with human scores compared to GPT-4o-as-a-Judge, and outperforms all other baselines. Additionally, GEM is more robust against strategic manipulations, such as rephrasing or elongation, which can artificially inflate scores under a GPT-4o-as-a-Judge.

GEM broadens the scenarios where people can benchmark LLM generation performance, extending from traditional tasks like machine translation and summarization (where gold standards are readily available) to subjective tasks without clear gold standards, such as academic peer review. I present GRE-bench (Generating Review Evaluation Benchmark), which evaluates LLMs based on how well they generate high-quality peer reviews for academic research papers. Because GRE-bench is built upon GEM, it inherits its alignment properties. Additionally, GRE-bench circumvents data contamination problems (or data leakage) by leveraging the continuous influx of new open-access research papers and peer reviews each year.

Contribution 4: Systematic design and test framework for aligned text evaluation metrics with f-mutual information. Fourth, I study how aligned text evaluation metrics should be constructed and validated systematically. Existing mutual-information-based metrics for text evaluation, such as GPPM, GEM, and TVD-MI [Robertson and Koyejo, 2025], were proposed for overlapping goals but evaluated on different datasets and with different criteria, making direct comparison difficult. To address this, I introduce a systematic test framework organized around three test principles: correlation with human judgments, degradation sensitivity, and manipulation resistance.

I also develop a unified f-mutual-information design framework for constructing aligned text evaluation metrics. In this framework, each metric is specified by a design triple consisting of the information measure, the estimation method, and the text distribution model together with its scoring granularity. This decomposition recovers prior methods such as GEM, GPPM, and TVD-MI as special cases, while revealing a broader design space for new metrics.

Combining the test and design framework, I conduct controlled factorial experiments across peer review, summarization, and question answering tasks to isolate how individual design choices affect human correlation, degradation sensitivity, and manipulation resistance.

CHAPTER 2

Spot Check Equivalence: An Interpretable Metric for Information Elicitation Mechanisms

2.1 Introduction

Eliciting precise and valuable information from individuals is becoming paramount, especially with the rising demands for data labeling in the realms of AI and machine learning. Recent advancements, such as Large Language Models (LLMs), have proven the value of high-quality human-labeled data. For example, Meta is estimated to have invested upward of 25 million dollars in collecting preference data from human labelers to align Llama 2 with human preferences [Lambert, 2023]. This raises a pressing question: How can human agents be incentivized to provide high-quality information. E.g., without the proper incentives, human labelers for LLM alignment may not exert effort to distinguish between truthful LLM responses and merely authoritative-sounding responses (i.e. hallucinations), even when truthfulness is important for the task at hand.

Research from Amazon reveals that monetary compensation is the principal motivator for Amazon Mechanical Turk workers Chen et al. [2011], and indeed the primary solution is to monetarily reward agents in exchange for effortful and truthful labeling. Two distinct compensation strategies, *spot-checking* Jurca and Faltings [2005] and *peer prediction* Miller et al. [2005], each rate the quality of user feedback with a score. This score can then be transformed into a payment for an agent.

Spot-checking mechanisms reward agents by comparing their reports with the ground truth on a small fraction of gold standard questions. When the ground truth information is expensive or even infeasible to obtain, peer prediction mechanisms are proposed, which reward an agent based on the correlation between her reports and the reports of other agents.

To understand and compare the performance of mechanisms developed from these paradigms, there is a need for standard metrics similar to accuracy, recall, and F1 score used in supervised learning. Notice that all these metrics range from 0 to 1 where 1 is good/perfect

and 0 is bad. While several studies have proposed methods for comparing these mechanisms [Zhang and Schoenebeck, 2023b, Burrell and Schoenebeck, 2021, Gao et al., 2016], there remains a conspicuous gap for both a unified understanding of how these metrics relate, and, if possible, a unified interpretable metric.

To this end, we introduce the concept of *Spot Check Equivalence* (SCE), which uses a spot-checking mechanism as a benchmark to quantify the motivational proficiency of an arbitrary incentive mechanism. As introduced in Zhang and Schoenebeck [2023b], the motivational proficiency is the minimum cost of budget to induce a desired effort level in a symmetric equilibrium. Then, a SCE of 1 will indicate that a mechanism does as well as a certain spot-checking mechanism does when the spot-checking mechanism has access to the ground truth of every task. A SCE of 0 will indicate that a mechanism does as well as this same spot-checking mechanism when it has no access to the ground truth of any task (essentially, it is paying agents randomly). Note that accessing the ground truth might be costly, e.g., hiring an expert to get the ground truth might be much more expensive than hiring several non-expert crowd workers. Thus, SCE can quantify the considerable cost savings that might be achieved by employing a peer prediction mechanism over a straightforward spot-checking mechanism.

By sufficiently harshly punishing the agents for the checked low-quality reports, spot-checking mechanisms can effectively motivate effort, even when only a small fraction of the tasks are checked. However, in most real applications, the payoff should be non-negative which precludes this approach.

Gao et al. [2016, 2019] study a peer grading setting where agents are modeled as having a binary choice for the effort to exert: low versus high. In their setting, the goal is to minimize the fraction of random questions that must be spot-checked to incentivize agents to exert high effort (make choosing high effort a Nash equilibrium). They find that, in their model, combining spot-checking with peer prediction does not help reduce the spot-checking ratio required to achieve the desired incentive properties, i.e. peer prediction makes things worse. However, their results rest on several assumptions, which we will discuss later in Section 2.6.

Burrell and Schoenebeck [2021] propose a metric called *Measurement Integrity* to quantify the ex-post fairness of a peer prediction mechanism. Mechanisms with high Measurement Integrity can produce payments that are strongly correlated with the quality of the agents' reports. Their motivation and definitions look beyond incentives to fairness. They do not study spot-checking mechanisms, but it is clear that the more agents are spot-checked the more accurately their scores will reflect their true quality. For example, with ground truth for all the tasks, an agent's score could exactly reflect the true quality of her responses. Moreover, they model continuous effort but do not establish a clear link between Measurement Integrity

and the ability to incentivize effort.

Zhang and Schoenebeck [2023b] study incentivizing effort in a crowdsourcing setting where agents can choose their effort from a continuum. Their goal is to maximize the motivational proficiency by rescaling the scores output by the incentive mechanism into practical payments. They suggest a tournament-based payment scheme: first rank the agents based on their scores output by an incentive mechanism, and then pay a predetermined reward for each ranking. Under the tournament setting, they further propose a sufficient statistic of a mechanism's motivational proficiency, called the *Sensitivity*. Intuitively, the Sensitivity measures how responsive a score is to changes in an agent's effort. For example, a spot-checking mechanism that checks a larger fraction of tasks is more sensitive to changes in an agent's effort (intuitively, it has more chances to detect a change), and thus has a larger motivational proficiency. However, there is a lack of discussions on how to estimate Sensitivity in practice.

An apparent contradiction arises. Zhang and Schoenebeck [2023b] empirically show that when agents exert a reasonably high effort, peer prediction mechanisms have Sensitivity competitive with spot-checking mechanisms that randomly check 20% of the tasks. However, the aforementioned implication of Gao et al. [2016, 2019] would seem to predict that the spot-checking mechanisms are always superior to peer-prediction mechanisms.

Our contributions. First, we propose Spot Check Equivalence which uses the equivalent spot-checking ratio as an interpretable way to measure an information elicitation mechanism's performance under specified information elicitation contexts. We study the Spot Check Equivalence based on Measurement Integrity and Sensitivity, and demonstrate its effectiveness as a metric for motivational proficiency both theoretically and empirically.

Second, we unify Measurement Integrity (the metric for ex-post fairness) and Sensitivity (the metric that serves as a proxy for motivational proficiency). In particular, we prove that Spot Check Equivalence based on Measurement Integrity and Sensitivity are sometimes exactly the same. We also show why these results differ so much from Gao et al. [2016, 2019], and thus refute, or at least qualify, the titular statement that "Peer-prediction makes things worse."

Third, we present two approaches to compute Spot Check Equivalence, which are suitable for settings with and without ground truth data, respectively. Our method enables the comparison of the motivational proficiency of different mechanisms across various information elicitation contexts. Furthermore, our simulation results show that both approaches result in similar estimations of SCE, which implies the robustness of our methods.

2.2 Model

In this section, we will give a formal definition of the information elicitation context (Figure 2.1), and then formally define the Spot Check Equivalence.

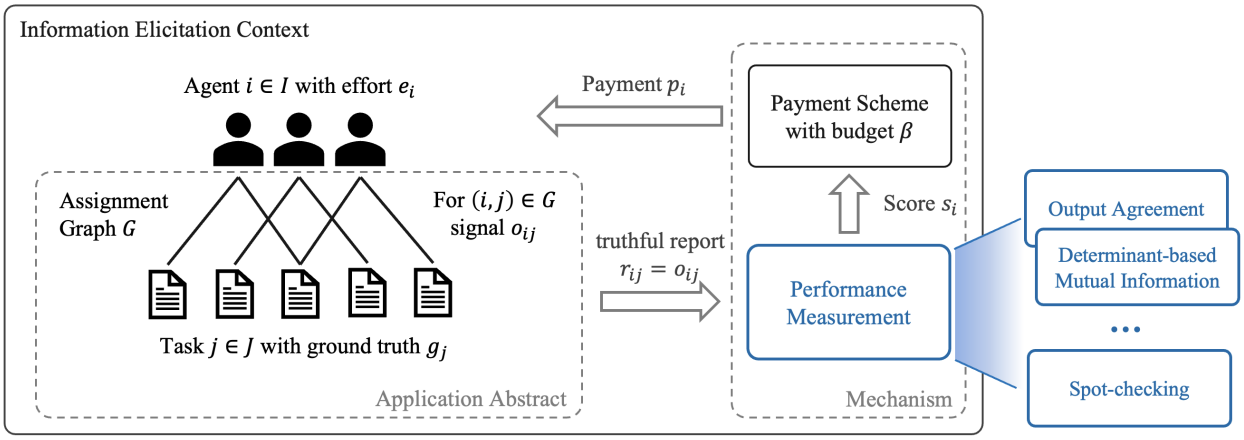


Figure 2.1: Information Elicitation Context

2.2.1 Information Elicitation Context

Formally, as shown in Figure 2.1, an information elicitation context (IEC) is defined as a tuple:

$$\text{Information Elicitation Context } (IEC) := (Agent, App, Mech)$$

where $Agent = (I, c, \mathbf{e})$ represents the agents and their properties, $App = (J, \mathcal{GT}, \omega, \Sigma, D)$ represents an information elicitation application abstraction, and $Mech = (M, P)$ represents a mechanism. We assume that the information elicitation context is common knowledge for all the agents.

Agent In $Agent = (I, c, \mathbf{e})$: I is the set of agents; $\mathbf{e} = [e_i]_{i \in I} \in [0,1]^{|I|}$ represents all agents' effort levels; the cost function $c : [0,1] \to \mathbb{R}^+ \cup \{0\}$ maps an effort level to a non-negative, increasing, and convex cost. Notice agents are homogeneous and share the same cost function.

Application Abstraction $App = (J, \mathcal{GT}, \omega, \Sigma, D)$ comprises the task set J, the ground truth space $\mathcal{GT}$, the prior of the ground truth $\omega = \Delta_{\mathcal{GT}}$, the signal space Σ, and the data-generating process $D = (D_{assign}, D_{signal})$.

D_{assign} describes how the tasks are assigned to the agents.

$$D_{assign} : \Delta_{\mathcal{G}}$$

where $\mathcal{G}$ represents the space over G, and $G = (I \cup J, E_G)$ represents a bipartite graph between I and J, indicating how the tasks are assigned to the agents.

D_{signal} describes how the signals are generated: the distribution of agent i's signal on task j conditioned on the effort $e_i \in [0, 1]$ and task j's ground truth $g_j \in \mathcal{GT}$, given the edge $(i, j) \in E_G$:

$$D_{signal} : [0, 1] \times \mathcal{GT} \to \Delta\Sigma$$

Agents' Report We assume agents truthfully report the signals they obtain from the application abstraction conditioned on their effort levels. Further discussion will be provided in Appendix 2.7.1. And we denote the agent i's report on task j as $r_{ij} \in \Sigma$.

Application Instance With the specified $Agent = (I, c, \mathbf{e})$ and $App = (J, \mathcal{GT}, \omega, \Sigma, D)$, we can generate an instance representing a realized information elicitation application:

- For the given I, J, we sample an assignment graph G according to D_{assign}.
- For each task $j \in J$, we independently sample its ground truth g_j from the prior ω.
- For each pair $(i, j) \in E_G$, we independently sample agent i's signal on task j from the distribution $D_{signal}(e_i, g_j)$, denoted as $o_{ij} \in \Sigma$.
- For each pair $(i, j) \in E_G$, as we assumed, the agent i's report $r_{ij} = o_{ij}$.

The mechanism, introduced presently, takes the application instance as input.

Performance Measurement The performance measurement M is a component of the mechanism $Mech = (M, P)$. It maps the agents' reports to their performance scores. Formally,

$$\text{(Peer Prediction) } M : \Sigma^{|E_G|} \to_{random} \mathbb{R}^{|I|}$$

$$\text{(Spot-checking) } M : \Sigma^{|E_G|} \times \mathcal{GT}^{|J_{\text{checked}}|} \to_{random} \mathbb{R}^{|I|}$$

Note that the spot-checking performance measurement can access the ground truth of the checked tasks $J_{\text{checked}} \subseteq J$.

We use $s_i \in \mathbb{R}$ to denote agent i's score, and $\mathbf{s} = [s_i]_{i \in I}$ to denote the vector of all agents' scores.

Payment Scheme The payment scheme P is the other component of the mechanism. It maps the agents' performance scores to the payoffs, which are directly related to their utilities. Formally,

$$\text{(Payment Scheme) } P : \mathbb{R}^{|I|} \to (\mathbb{R}^* \cup \{0\})^{|I|}$$

We use $p_i \in \mathbb{R}^* \cup \{0\}$ to denote payoff of agent i, and $\beta = \sum_{i \in I} p_i$ to denote the total payment among all the agents. And we denote the vector of all the agents' payoffs as $\mathbf{p} = [p_i]_{i \in I}$.

For intuition, we give two examples of payment schemes:

Definition 2.1 (Linear Payment Scheme)**.** *A linear payment scheme pays a payoff $p_i = a \cdot s_i + b$ to each agent i, where a, b are the constant parameters.*

Definition 2.2 (Tournament Payment Scheme)**.** *A tournament payment scheme first ranks the agents according to their scores and then pays the i-th ranked agent $\hat{p}_i$, where $\hat{p}_1, \hat{p}_2, ..., \hat{p}_{|I|}$ are constant parameters that monotonically decrease, i.e., $\hat{p}_i \geq \hat{p}_{i'}$ when $i \leq i'$. Without loss of generality, we assume $s_1 \geq s_2 \geq ... \geq s_{|I|}$, and thus $p_i = \hat{p}_i$.*

Report quality. Given an instance, we can define the quality of an agent's report. The quality function Q for one report is a deterministic loss function:

$$\text{(Quality Function) } Q : \Sigma \times \mathcal{GT} \to \mathbb{R}$$

Agent i's overall report quality q_i is defined as the average of her reports' qualities, i.e. $q_i = \sum_{j|(i,j) \in G} Q(r_{ij}, g_j)$. We denote the vector of all the agents' qualities as $\mathbf{q} = [q_i]_{i \in I}$.

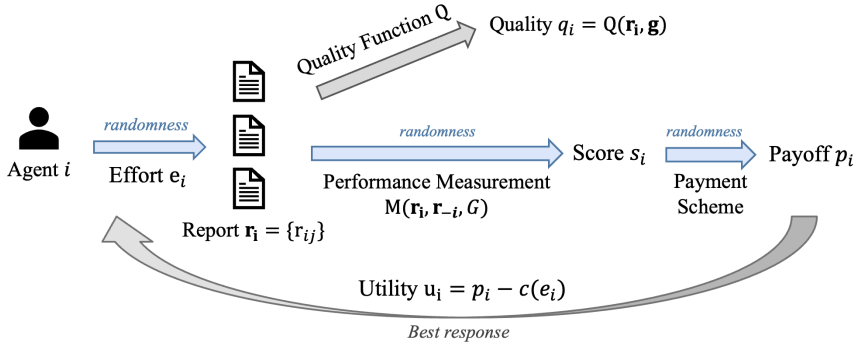


Figure 2.2: An Agent's Perspective of an Information Elicitation Context

Equilibrium We assume that the agents choose their effort level according to the following equilibrium.

Definition 2.3 (Symmetric local equilibrium)**.** *Given an IEC where all the agents exert effort $e_i = \xi$ and $\mathbb{E}[\sum_{i\in I} p_i] \geq |I| \cdot c(\xi)$ (Individual Rationality is satisfied), we say it is a symmetric local equilibrium if the derivative of every agent's utility is* 0*, i.e.*

$$\frac{\partial}{\partial e_i} u(e_i, e_{-i} = \xi)|_{e_i=\xi} = 0$$

Note that, at this equilibrium, $\frac{\partial}{\partial e_i} u(e_i, e_{-i} = \xi)|_{e_i=\xi} = 0$ is a necessary condition for ξ being a Nash equilibrium. Zhang and Schoenebeck [2023b] show empirical evidence that a local equilibrium is very likely to be a Nash equilibrium in the model we mainly discuss in our paper. In the rest of our paper, our discussion will focus on this equilibrium.

2.2.2 Motivational Proficiency

The motivational proficiency of a component (a mechanism, a performance measurement, or a payment scheme) within an information elicitation context IEC represents its ability to incentivize effort. To quantify it, we fix all the other components of the IEC and quantify the expected total payment for eliciting a fixed effort level at the equilibrium (Definition 2.3), lower expected total payment implies higher motivational proficiency.

Definition 2.4 (Motivational proficiency)**.** *We define the motivational proficiency of a component ($Mech$, M, or P) within an information elicitation context IEC where all the agents exert effort level ξ as the negative expected total payment needed to realize the symmetric local equilibrium (Definition 2.3) at effort level ξ when substituting the component into the information elicitation context.*

As we discussed in the introduction, Zhang and Schoenebeck [2023b] show that tournament payment schemes have higher motivational proficiency compared to linear payment schemes in certain settings where limited liability is needed. Therefore, in the following discussion, we will focus on the motivational proficiency of performance measurements in information elicitation contexts with a tournament payment scheme. In Section 2.5, we quantify the total payment in an information elicitation context with tournament payment.

We provide a simple example that illustrates the calculation of expected total payment in an information elicitation context with a winner-take-all tournament payment scheme. In the next section, we then demonstrate the Sensitivity and the Measurement Integrity in this example.

Example 2.5. *Consider a simple case where there are two agents (1 and 2) working on a large number of tasks. Each agent $i \in \{1, 2\}$ can choose to exert a non-negative effort level e_i,*

which incurs a cost $c(e_i)$. And we use a quadratic cost function, $c(e_i) = e_i^2$, which is one of the simplest functions that satisfies all the properties of a cost function.

Since the signals are independent conditioning on an agent's effort level and the spot-checking performance measurement checks each task u.a.r., we use normal distributions to approximate the quality and performance score: Agent i's report quality is $q_i \sim N(e_i, 1)$, and the performance score is $s_i \sim N(q_i, \sigma^2)$, where σ is monotonically decreasing with the spot-checking ratio. Consequently, we have that $s_i \sim N(e_i, \sigma^2 + 1)$

The expected utility of agent i is

$$u_i(e_i) = \Pr[i \text{ wins the tournament}] \cdot \text{payoff} - c(e_i)$$

$\Pr[i \text{ wins the tournament}] = \Pr[s_i - s_{-i} \geq 0]$, *where $-i$ denotes the other agent. Notice that $(s_i - s_{-i}) \sim N(e_i - e_{-i}, 2\sigma^2 + 2)$ because the difference of two normal random variables is also a normal random variable where the variances add. Therefore, we have*

$$\Pr[i \text{ wins the tournament}] = CDF_{N(e_{-i}, 2\sigma^2+2)}(e_i)$$

where $CDF_{N(e_{-i}, 2\sigma^2+2)}$ is the cumulative distribution function of the normal distribution $N(e_{-i}, 2\sigma^2 + 2)$.

To achieve the symmetric local equilibrium at effort level ξ, we set the derivative of agent i's expected utility at $e_i = \xi$ equal to 0.

$$\frac{\partial}{\partial \xi} u_i(\xi) = pdf_{N(\xi, 2\sigma^2+2)}(\xi) \cdot \text{payoff} - 2\xi = 0$$

where $pdf_{N(\xi, 2\sigma^2+2)}$ is the probability density function of normal distribution $N(\xi, 2\sigma^2 + 2)$, which is the derivative of $\Pr[i \text{ wins the tournament}]$ at $e_i = \xi$. By simplification, we get, $pdf_{N(\xi, 2\sigma^2+2)}(\xi) = \frac{1}{2\sqrt{\pi \cdot (\sigma^2+1)}}$. Solving for the payoff, we have

$$\text{payoff} = 4\xi \cdot \sqrt{\pi \cdot (\sigma^2 + 1)}$$

In addition, we need to keep Individual Rationality, i.e. the expected utility for each agent should not be negative so that rational agents will not leave, which implies the total payoff to all agents should cover the total cost of all agents, i.e., $\text{payoff} \geq 2c(\xi) = 2\xi^2$. *Putting these together, the payoff should be*

$$\text{payoff} = \max\left(2\xi^2, 4\sqrt{\pi \cdot (\sigma^2 + 1)} \cdot \xi\right)$$

In the above example, we note that, σ monotonically decreases with the spot-checking ratio, and the total payoff monotonically increases with σ when IR is not binding. Therefore, we can see that a higher spot-checking ratio leads to a lower total payment when IR is not binding, and consequently, a higher motivational proficiency.

2.2.3 Measure of Performance Measurements

In addition to motivational proficiency, there are other measures of performance measurements, as we discussed in the introduction, including Sensitivity [Zhang and Schoenebeck, 2023b] and Measurement Integrity [Burrell and Schoenebeck, 2021]. We propose the general definition of a measure of performance measurement M.

Definition 2.6 (Measure of Performance Measurement)**.** *A measure f of performance measurement M within information elicitation context IEC maps the IEC with M to a real number, denoted as $f(IEC \leftarrow M) \in \mathbb{R}$, where the leftarrow means we apply M in the information elicitation context IEC.*

We now show the two examples of measure f, Sensitivity [Zhang and Schoenebeck, 2023b] and Measurement Integrity [Burrell and Schoenebeck, 2021].

Sensitivity Zhang and Schoenebeck [2023b] propose the Sensitivity as a proxy of the motivational proficiency of a performance measurement. They show that the motivational proficiency highly depends on the Sensitivity (Definition 2.7, Lemma 2.8), which measures how an agent's performance score changes when she deviates from the equilibrium effort level ξ. When all other agents exert effort ξ, we denote the mean of agent i's score as $\mu_s(e_i)$, and the standard deviation as $\sigma_s(e_i)$.

Definition 2.7 (Sensitivity Zhang and Schoenebeck [2023b])**.** *The Sensitivity of a performance measurement within an information elicitation context IEC at equilibrium effort level ξ is defined as*

$$\text{Sensitivity}(IEC \leftarrow M) = \delta(\xi) = \frac{\frac{\partial}{\partial e_i}\mu_s(e_i)|_{e_i=\xi}}{\sigma_s(\xi)}$$

Lemma 2.8 (Proposition 4.8 in Zhang and Schoenebeck [2023b])**.** *If the agent i's score s_i follows a normal distribution $N(\mu_s(e_i), \sigma_s(e_i)^2)$, the expected total payment to elicit effort ξ will (weakly) decrease in the Sensitivity $\delta(\xi)$ in a specific information elicitation context with any tournament payment scheme.*

To illustrate, we show the Sensitivity $\delta(\xi)$ in Example 2.5.

$$\delta(\xi) = \frac{\frac{\partial}{\partial e_i}\mu_s(e_i)|_{e_i=\xi}}{\sigma_s(\xi)} = \frac{1}{\sqrt{\sigma^2+1}}$$

We find that the total payment in Example 2.5, $\max\left(2\xi^2, 4\sqrt{\pi\cdot(\sigma^2+1)}\cdot\xi\right)$, is inversely proportional to the Sensitivity when IR is not binding, which aligns with the Lemma 2.8.

Measurement Integrity To measure the ex-post fairness of a performance measurement, Burrell and Schoenebeck [2021] propose the Measurement Integrity, which is defined as the expected correlation between the quality of the agents' reports and their performance scores.

Definition 2.9 (Measurement Integrity)**.** *Formally, the Measurement Integrity of a performance measurement M with respect to a quality function Q and a correlation function* corr*, within an information elicitation context IEC is*

$$\underset{Q,\text{corr}}{\text{MI}}\left(IEC \leftarrow M\right) = \mathbb{E}_{IEC}\left[\text{corr}\left(\mathbf{s},\mathbf{q}\right)\right]$$

In Example 2.5, we can see that the motivational proficiency highly depends on how an agent's performance score correlates with her report quality. In the example, the Pearson correlation coefficient between the agent i's report quality and score is as follows.

$$\rho(q_i, s_i) = \frac{\mathbb{E}[q_i\cdot s_i] - \mathbb{E}[q_i]\mathbb{E}[s_i]}{\sigma_{q_i}\cdot\sigma_{s_i}} = \frac{1}{\sqrt{\sigma^2+1}}$$

Note that the total payment is also inversely proportional to the Pearson correlation coefficient! And the Sensitivity has the same form as the correlation in this example.

This example intuitively shows that the correlation between the agents' report qualities and scores can be a proxy for a performance measurement's motivational proficiency. And both the report quality and score are accessible in real data.

Therefore, we can employ both Sensitivity [Zhang and Schoenebeck, 2023b] and Measurement Integrity [Burrell and Schoenebeck, 2021] as our proxy for motivational proficiency. In Section 2.3, we further formally unify these two proxies within our model.

2.2.4 Using Spot-checking as a Reference: Spot Check Equivalence

Even though we have metrics which can compare two performance measurements, there is still a need for a metric for a performance measurement whose value is per se. Our idea is to take any metric of performance measurements, f, and convert it to an interpretable metric

as follows: instead of using the actual value of f applied to some performance measurement, instead use the checking ratio of the spot-checking performance measurement who, when also evaluated by f, yields a value equivalent to that of the performance measurement.

First, we adopt the definition of spot-checking performance measurement from Gao et al. [2016], and assume it can access the ground truth, thus, it is an idealized spot-checking. Given that we exclusively focus on this particular spot-checking, we might omit the term 'idealized' in subsequent discussions for brevity.

Definition 2.10 (Spot-checking performance measurement (idealized))**.** *An (idealized) spot-checking performance measurement is denoted as a tuple $SC := (X, S, C)$, which checks X percent of all the tasks u.a.r.; then scores the agent i with $S(r_{ij}, g_j)$ for each checked task j, and score $C \in \mathbb{R}$ for each of the unchecked tasks.*

Intuitively, a higher checking ratio leads to less noise, so high effort is easier to notice, which is beneficial for both motivational proficiency and ex-post fairness. Thus, we can use the equivalent spot-checking ratio as a metric for both motivational proficiency and ex-post fairness of a performance measurement.

Formally, we can define the Spot Check Equivalence as follows.

Definition 2.11 (Spot Check Equivalence)**.** *For a performance measurement M within an information elicitation context $IEC = (Agent, App, Mech)$, at the symmetric local equilibrium $\mathbf{e} = \xi$, given a measure of performance measurements f (Definition 2.6), we define the Spot Check Equivalence SCE of M, with respect to a spot-checking mechanism SC as*

$$SCE(IEC \leftarrow M) = \sup_X \{f(IEC \leftarrow M) \geq f(IEC \leftarrow SC(X, S, C))\}$$

In our following discussion, to make the spot-checking mechanism $SC(X, S, C)$ non-trivial[1], we use a quality function to score the agents for checked tasks, i.e. $S = Q$. And since we will apply a payment scheme after the performance measurement, the value of the constant score C for unchecked tasks does not matter, thus, we set $C = 0$. We then use $SC(X)$ to denote $SC(X, S = Q, C = 0)$.

In the next section, we will theoretically prove the unification of Sensitivity and Measurement Integrity in certain settings, and consequently, we can show that the Spot Check Equivalence based on the Sensitivity or Measurement Integrity can be used as an interpretable metric for the motivation proficiency of an information elicitation performance measurement. We will show more empirical evidence in our agent-based model simulations (Section 2.5) when relaxing the theoretical assumptions.

[1]An example of a trivial spot-checking mechanism is $S(r_{ij}, g_j) \equiv 0$ and $C \equiv 0$.

2.3 Unification of Sensitivity and Measurement Integrity

In this section, we formally prove that the Spot Check Equivalence based on Sensitivity or Measurement Integrity can be used as a proxy for Spot Check Equivalence based on motivational proficiency in certain settings. Given that computing these three measures has different requirements, the unification allows us to compute Spot Check Equivalence and consequently, measure the motivational proficiency in more scenarios.

We formally propose and prove the unification of Measurement Integrity and Sensitivity in our main theorem (Theorem 2.15). Recall that Zhang and Schoenebeck [2023b] have shown that, within certain settings, high Sensitivity leads to low expected total payment (Lemma 2.8) when applying a tournament payment scheme. Putting these together, we have that the Spot Check Equivalence based on motivational proficiency, Sensitivity, and Measurement Integrity are equal.

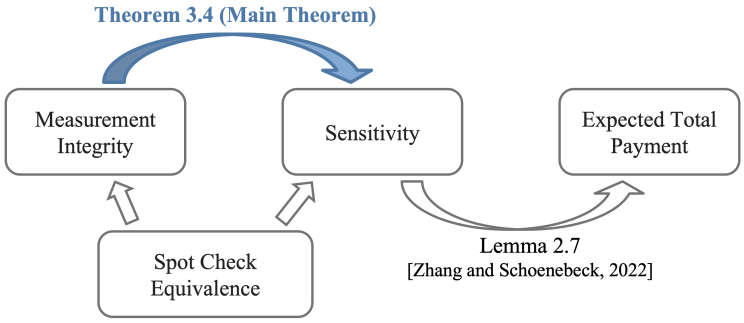


Figure 2.3: Theoretical Analysis Overview: Spot Check Equivalence based on Measurement Integrity, Sensitivity and motivational proficiency (negative expected total payment) are equivalent.

Since the Sensitivity relies on the distribution of the performance score, we make the following assumptions about the distributions of the quantities in our model, similar to prior work [Zhang and Schoenebeck, 2023b].

Assumption 2.12 (The Gaussian assumption for the quality)**.** *Given effort level e_i, the quality q_i follows a normal distribution $N(e_i, \sigma_q(e_i)^2)$. And we further assume that $\sigma_q'(e_i) << \sigma_q(e_i)$.*

Assumption 2.13 (The Gaussian assumption for the score)**.** *Given the report quality q_i, the score s_i follows a normal distribution $N(\mu_{s|q}(q_i), \sigma_{s|q}(q_i)^2)$.*

Zhang and Schoenebeck [2023b] show empirical evidence that the performance score roughly follows a normal distribution, here, we further make this assumption for the quality, a quantity that did not exist in the prior work [Zhang and Schoenebeck, 2023b]. When the

number of agents and the number of tasks per agent grows large, it is reasonable to assume the quality is normal distributed because it can typically be seen as the sum of many i.i.d. random variables.

Note $\mu_{s|q}$ is different from the μ_s in the definition of Sensitivity, in particular,

$$\mu_s(e) = \int_q \mu_{s|q}(q) \Pr[q|e] \, \mathrm{dq}$$

Assumption 2.14 (Independence assumption for scores)**.** *When all agents have the same effort level $e_i = \xi$ and the number of agents $|I|$ goes to infinity, the agents' performance scores are independent.*

Note that different agents' qualities q_i are independent. When we apply a spot-checking performance measurement, the scores are naturally independent. When we apply a peer-sensitive performance measurement (e.g. peer prediction), an agent's performance score depends on her report quality as well as other agents' report quality. However, as the number of agents grows large, the amount that one agent's reports can impact another agents' scores will go to zero. Thus, we assume that the scores are independent when the number of agents goes to infinity (Assumption 2.14).

Theorem 2.15 (Main Theorem)**.** *For a given performance measurement M within an information elicitation context IEC where every agent exerts effort level ξ, when Assumption 2.12 2.13 and 2.14 are satisfied, there exists a linear bijection between the* $\mathrm{MI}_{Q,\mathrm{corr}}(IEC \leftarrow M)$ *and the Sensitivity $\delta(\xi)$, where* corr *is the sample Pearson correlation coefficient and the number of agents goes to infinity.*

The proof of Theorem 2.15 is provided in Appendix A.1.

Note that our main theorem (Theorem 2.15) relies on Assumption 2.12, 2.13, and 2.14, thus, it is important to examine whether the unification of motivational proficiency, Sensitivity, and Measurement Integrity is still true with real scores calculated by various performance measurements. In Section 2.5, we will demonstrate some positive evidence from our agent-based model experiment, where Assumption 2.12, 2.13, and 2.14 are relaxed.

2.4 Computing Spot Check Equivalence

Given the unification of Sensitivity and Measurement Integrity, if we can compute the Sensitivity or Measurement Integrity of a performance measurement, we can get the Sensitivity-based or Measurement Integrity-based Spot Check Equivalence respectively, and consequently, we can get an interpretable metric for motivational proficiency.

If the measure $f(IEC \leftarrow SC(X))$ is monotonic with respect to X, we can use a binary search algorithm (Algorithm 1) to compute the Spot Check Equivalence.

ALGORITHM 1: Binary Search algorithm for SCE

Input: Information Elicitation Context IEC, Performance Measurement M, step size ϵ

Output: Spot Check Equivalence SCE

Function `BinarySearchSCE(`M, f`)`**:**

```
low = 0, high = ⌊100/ϵ⌋
while low ≤ high do
    mid = ⌊(low+high)/2⌋
    if f(IEC ← SC(X = mid * ϵ)) < f(IEC ← M) then
        ans = mid
        low = mid + 1
    end
    else
        high = mid − 1
    end
end
return ans
```

Furthermore, a linear combination of the two spot-checking ratios which has adjacent measure f may be applied for a better approximation.

$$SCE = \text{ans} + \epsilon \cdot \frac{f(IEC \leftarrow M) - f(IEC \leftarrow SC(\text{ans}))}{f(IEC \leftarrow SC(\text{ans} + \epsilon)) - f(IEC \leftarrow SC(\text{ans}))}$$

2.4.1 Computation with Ground Truth

We first propose a workflow to compute the Spot Check Equivalence with the ground truth of the tasks. The Spot Check Equivalence is like accuracy, recall, and F1 score in machine learning, which can only be calculated on training or testing datasets rather than real applications. Similarly, it is reasonable to create datasets to evaluate the information elicitation performance measurements, get a good sense of their motivational proficiency, and then decide which performance measurement to apply in real applications.

With the ground truth of the tasks, we can calculate the quality of the reports, and then, use the correlation between the agents' scores and qualities to estimate the measurement integrity of the performance measurement M and $SC(X)$.

Intuitively, as more tasks are checked, the score s_i will be more correlated to the quality q_i. Our agent-based model experiment also shows that the Measurement Integrity monotonically

increases with the spot-checking ratio (Section 2.5.2). Therefore, we can apply Algorithm 1 to estimate the Spot Check Equivalence based on Measurement Integrity.

2.4.2 Computation without Ground Truth

Considering the current lack of information elicitation dataset, we propose another method to estimate the Spot Check Equivalence without the ground truth.

Recall the definition of Sensitivity (Definition 2.7), both the derivative of the performance score $\frac{\partial}{\partial e_i}\mu_s(e_i)|_{e_i=\xi}$ and the standard deviation $\sigma_s(\xi)$ do not require access to the ground truth. The standard deviation $\sigma_s(\xi)$ can be estimated by the standard deviation of $\{s_i\}$ given that all the agents are homogeneous. However, in real data, to estimate $\frac{\partial}{\partial e_i}\mu_s(e_i)|_{e_i=\xi}$ is tricky because the score after deviating from ξ is not accessible.

We adopt the idea of bootstrap sampling: we randomly select an agent i, and if we know how the effort impacts the report, we can randomly manipulate her report as an intentional degradation of her effort by some ε. We then compute the mean difference between all selected agents' scores before and after the manipulation, denoted as $\Delta\mu$. Note that the mean difference $\Delta\mu$ can be regarded as an estimation of $\varepsilon \cdot \frac{\partial}{\partial e_i}\mu_s(e_i)|_{e_i=\xi}$, and consequently, $\Delta\mu/\sigma_s$ is proportional to the Sensitivity. For example, if decreasing the effort brings uniform noise, we get the Algorithm 2. In Section 2.5.3, we present evidence demonstrating that Algorithm 2 works in our agent-based model simulation.

2.5 Effectiveness of the Spot Check Equivalence: Agent-Based Model Experiment

In this section, we present the results of agent-based model (ABM) experiments to evaluate the effectiveness of the *Spot Check Equivalence* based on Measurement Integrity and Sensitivity in measuring the motivational proficiency, without the assumptions we made in our theoretical proof. We then further compare different peer prediction performance measurements with spot-checking performance measurements in different information elicitation contexts.

2.5.1 Setup

In this subsection, we introduce our agent-based model setup. According to the formal definition of the information elicitation context in Section 2.2, our agent-based model contains the following components:

ALGORITHM 2: Estimate SCE without Ground Truth

Input: Information Elicitation Context IEC, Performance Measurement M, Iteration Times T

Output: Spot Check Equivalence SCE

Function Estimate $\Delta\mu(IEC \leftarrow M)$**:**

```
Δμ = 0
for t = 1 to T do
    Choose i ∈ I u.a.r.
    Compute score s_i with M
    for (i, j) ∈ G do
        if random(0, 1) > ε then
            Choose r_ij ∈ GT u.a.r.
        end
    end
    Compute score s'_i with M
    Δμ = Δμ + (s'_i − s_i)/T
end
return Δμ
```

SCE = `BinarySearchSCE(`$M, f = \Delta\mu/\sigma_s$`)`

2.5.1.1 Agents

We consider a population of $|I| = 50$ agents.

Effort and Cost. Each agent i has an effort level $e_i \in [0, 1]$ with an associated cost function $c(e_i) = e_i^2$.

2.5.1.2 Data-generating Process (for Application Instance)

Tasks. To show the effectiveness of Spot Check Equivalence, we consider an information elicitation application with $|J| = 500$ tasks. To compare the Spot Check Equivalence of different performance measurements in various contexts, we vary $|J|$ according to Table 2.1.

Assignment graph. The data-generating process D will randomly generate a bipartite graph between I and J: To show the effectiveness of Spot Check Equivalence, each agent is assigned to 50 tasks, while each task is assigned to 5 agents. To compare the Spot Check Equivalence of different performance measurements in various contexts, we vary the parameters according to Table 2.1.

Table 2.1: IEC parameter setups for ABM simulations

# agents	# tasks	# tasks per agent	# agents per task
50	50	5	5
50	100	10	5
50	$50 \times K$	$5 \times K$	5

where $K \in \{3, 4, 5, 6, 7, 8, 9, 10\}$

Ground truth and signals. For generating the ground truth and signals, we will apply the generalized Dawid-Skene model from previous work [Dawid and Skene, 1979, Zhang and Schoenebeck, 2023b], in which:

Each task $j \in J$ has a ground truth $g_j \in \mathcal{GT}$ where $\mathcal{GT}$ is a finite set including all possible ground truths. Since $\mathcal{GT}$ is finite, we can use a vector to represent the prior distribution of the ground truth ω, whose ith item represents the probability of ith possible ground truth. For convenience, we also denote that vector as ω.

In our experiment, we use the ω learned from a dataset of a crowdsourcing task on Amazon Mechanical Turk [Zhang and Schoenebeck, 2023b, VENANZI et al., 2015].

$$\omega \approx [0.196, 0.241, 0.247, 0.316]$$

The agent i will receive a signal $o_{ij} \in \Sigma$ on task j given her effort level and the task's ground truth. In our experiment, we assume that $\Sigma = \mathcal{GT}$. Then, we can define two $|\mathcal{GT}| \times |\Sigma|$ confusion matrices, Γ_{work} and Γ_{shirk}. The (row, col) entry of Γ_{work} and Γ_{shirk} represents the probability of getting the col-th signal conditioned on the row-th ground truth when the agent exerts effort level 1 and 0 respectively.

When the agent i exert e_i effort, the confusion matrix is

$$\Gamma_i = e_i * \Gamma_{work} + (1 - e_i) * \Gamma_{shirk}$$

where the confusion matrix Γ_{work} is also learned from the above dataset [Zhang and Schoenebeck, 2023b, VENANZI et al., 2015], and we set Γ_{shirk} as a matrix where each row represents a uniform distribution.

$$\Gamma_{work} \approx \begin{bmatrix} 0.771 & 0.122 & 0.084 & 0.024 \\ 0.091 & 0.735 & 0.130 & 0.044 \\ 0.033 & 0.062 & 0.866 & 0.039 \\ 0.068 & 0.164 & 0.099 & 0.669 \end{bmatrix}$$

2.5.1.3 Performance Measurement

We consider several performance measurements M to evaluate the effectiveness of the Spot Check Equivalence. We implement several peer prediction mechanisms, which yield different Spot Check Equivalences within the above information elicitation context.

Output Agreement (OA) [Von Ahn and Dabbish, 2004, 2008, Waggoner and Chen, 2014] In the OA mechanism, we pair the agents who work on the same task, and compare their reports. If the reports match, both agents receive a score of 1, and if they don't, the score is 0. One agent's total score is the summation of the scores on all the tasks assigned to him.

Peer Truth Serum (PTS) [Faltings et al., 2017] Similar to OA, we pair the agents who work on the same task, and compare their reports. Instead of scoring 1 for agreement, we score the inverse of the frequency of the agent's report, which means, intuitively, we reward more for the agreement of the uncommon reports.

Correlated Agreement (CA) [Shnayder et al., 2016] The CA mechanism is an extension of Dasgupta and Ghosh [2013]'s mechanism for multi-signal information elicitation. Without loss of generality, we denote the signal (report) space as $\Sigma = [1, 2, ..., n]$. Given two agents 1 and 2, let r_1 and r_2 denote the random variables of agent 1 and 2's report on all their tasks respectively. Then, we define an $n \times n$ delta matrix Δ as

$$\Delta_{row,col} = \Pr[r_1 = row, r_2 = col] - \Pr[r_1 = row] \Pr[r_2 = col]$$

We score the agent according to the agreement on positively correlated reports, i.e. we score agent 1 and agent 2 the following value.

$$\sum_{row,col} \Delta_{row,col} \cdot \mathbb{1}[\Delta_{row,col} > 0]$$

f-Mutual Information (f-MI) [Kong and Schoenebeck, 2018b, 2019] The f-MI mechanism scores the agent with the mutual information between her and her peer's reports, the intuition is that any manipulation on the two random variables will decrease the mutual information between them. We define the f-Mutual Information between agent 1 and 2's report as:

$$f\text{-MI} = \sum_{row,col} \Pr[r_1 = row, r_2 = col] f\left(\frac{\Pr[r_1 = row] \Pr[r_2 = col]}{\Pr[r_1 = row, r_2 = col]}\right)$$

Table 2.2: f-Mutual Information

$f(x)$	f-divergence	Notation
$-log(x)$	KL-divergence	KL
$\lvert x-1\rvert$	Total Variation Distance	TVD
$(\sqrt{x}-1)^2$	Squared Hellinger	H^2

where f is a convex function and $f(1) = 0$. For example, when we take $f(x) = -\log(x)$, then f-MI is the Shannon Mutual Information. Note that the f-Mutual Information could be regarded as the f-divergence ($D_f = \sum_x p(x) f\left(\frac{q(x)}{p(x)}\right)$) [Ali and Silvey, 1966] of the joint distribution and the product of the marginal distributions of the agents' reports.

We list all the f-Mutual Informations we will use in our simulations in Table 2.2. Note that when applying a Total Variation Distance divergence, the f-MI mechanism is almost equivalent to the CA mechanism, thus, we only show the result of the CA mechanism in our simulations.

Determinant-based Mutual Information (DMI) [Kong, 2020] Kong [2020] generalize the Shannon Mutual Information to the Determinant-based Mutual Information (DMI). Let an $n \times n$ matrix $\mathbf{U}_{X,Y}$ denote the joint distribution of two random variables, the DMI of these two random variables are defined as $DMI(X;Y) = |\det(\mathbf{U}_{X,Y})|$.

Then, given two agents 1 and 2, the score of the DMI mechanism is defined as the product of the DMI of the two agents' reports on two disjoint partitions of all the tasks. Specifically, we divide all the common tasks of 1 and 2 into two parts, calculate their DMI in these two parts respectively, and score each agent with the product of the two DMI.

However, even though DMI demonstrates impressive theoretical properties, it does not perform well in our simulation due to the considerable noise of its performance score (Figure 2.7).

Spot-checking In addition, to calculate the Spot Check Equivalence, we also implement a spot-checking performance measurement (Definition 2.10) as a benchmark, in which we use accuracy as the quality function, i.e. $Q(r, g) = \mathbb{1}[r = g]$. We denote this spot-checking performance measurement as $SC(X)$ where X is the checking percentage.

2.5.1.4 Payment Scheme

We now introduce the tournament payment scheme we use in our simulation.

Borda-count payment scheme. A very intuitive way to pay an agent according to her ranking is to pay her how many agents she beats. When there is a draw, we split the payoff evenly. Formally, we have

$$p_i = C \cdot \#\text{beaten} = C \sum_{i' \in |I|, i' \neq i} \left(\mathbb{1}[s_i > s_{i'}] + \frac{1}{2} \mathbb{1}[s_i = s_{i'}] \right)$$

where C is a constant parameter and the total payment is $C \times \binom{|I|}{2}$.

To calculate the total payment[2] in the Borda-count scheme for a specific performance measurement M for the symmetric local equilibrium where every agent exerts $e_i = \xi$ effort, we set the derivative of agent i's expected payoff equal to 0, which implies the parameter C should be

$$C = \frac{\frac{\partial}{\partial e_i} \mathbb{E}[\#\text{beaten} | e_i, e_{-i} = \xi]|_{e_i = \xi}}{\frac{\partial}{\partial e_i} c(e_i)|_{e_i = \xi}}$$

Note that to guarantee Individual Rationality (The agents' expected utility is non-negative), when the calculated optimal payment is less than the total cost of all the agents, we set the total payment as the total cost. We assume that if a payment scheme can incentivize effort level ξ using the optimal payment, it can also incentivize the same effort when the total payment is greater than the optimal payment.

2.5.2 Measurement-Integrity-based Spot Check Equivalence.

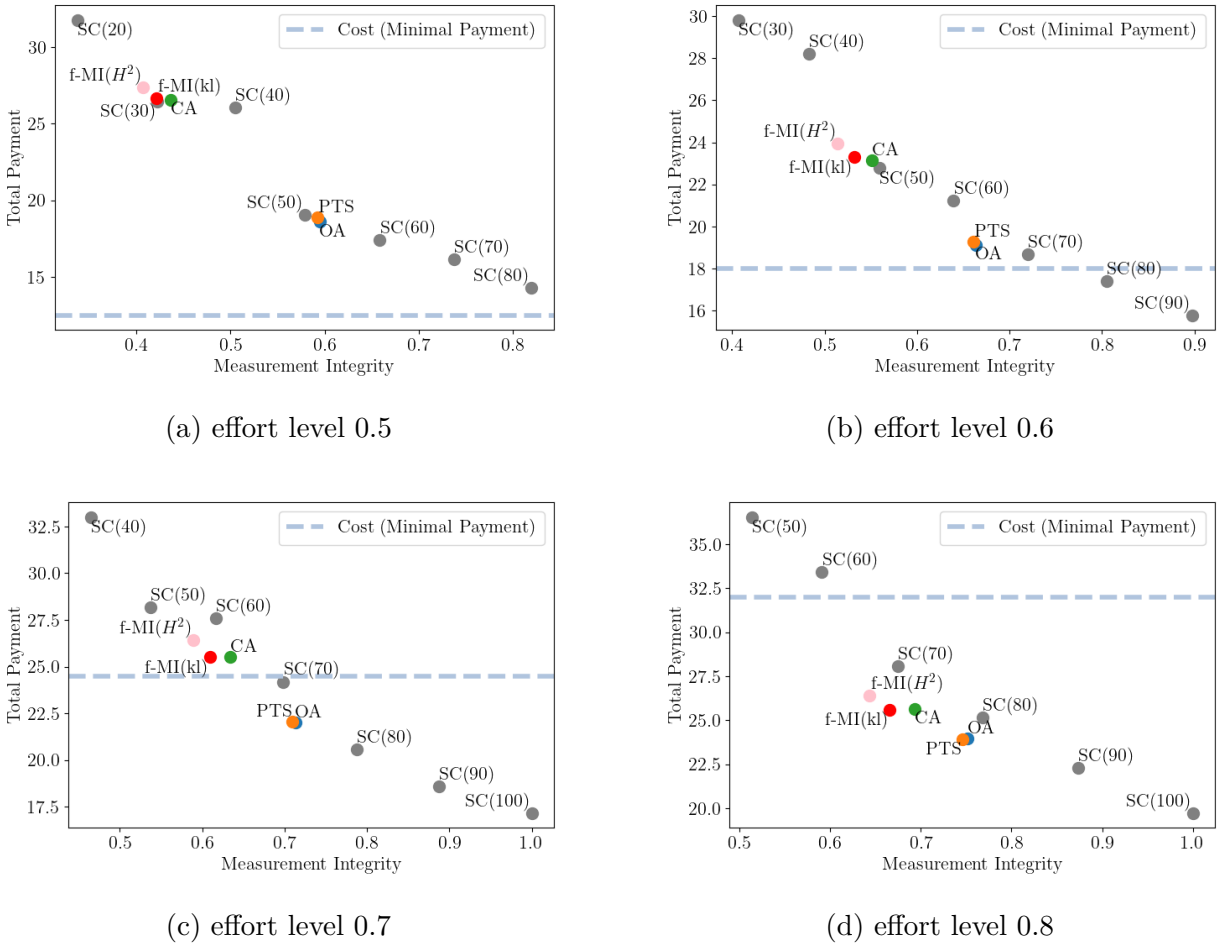
We examine whether the Spot Check Equivalence based on Measurement Integrity can indicate a performance measurement's motivational proficiency. Recall that the motivational proficiency of a performance measurement could be quantified by the amount of the expected total payment we need to elicit a fixed effort level in an information elicitation context. Thus, in the experiment examining the effectiveness, we mainly investigate the relationship between the Measurement Integrity and the expected total payment.

For several fixed effort levels, we apply all the performance measurements and estimate their Measurement Integrity and the total payment needed to elicit that equilibrium with the Borda-count payment scheme by iterating the data-generating process 5000 times. Detailed implementation is shown in Appendix A.2.

Recall that to satisfy Individual Rationality, the total payment needs to be at least the agents' cost to exert the effort. Since the minimal payment is the same for all performance measurements when the effort level is the same, we visualize that as a horizontal line in our

[2]Note that the total payment of Borda-count is deterministic, thus, we use "total payment" in the rest of this section instead of "expected total payment".

result.



(a) effort level 0.5

(b) effort level 0.6

(c) effort level 0.7

(d) effort level 0.8

Figure 2.4: Measurement Integrity v.s Total Payment of Borda-count payment scheme: the x-axis is the Measurement Integrity and the y-axis is the total payment needed to elicit that equilibrium within the tournament payment scheme. The horizontal line shows the agents' cost to exert the effort level, which implies the minimal payment to satisfy Individual Rationality.

With the results in Figure 2.4, we can observe that:

1. The Measurement Integrity monotonically increases with the spot-checking ratio.

2. The Measurement Integrity and the total payment are significantly negatively correlated.

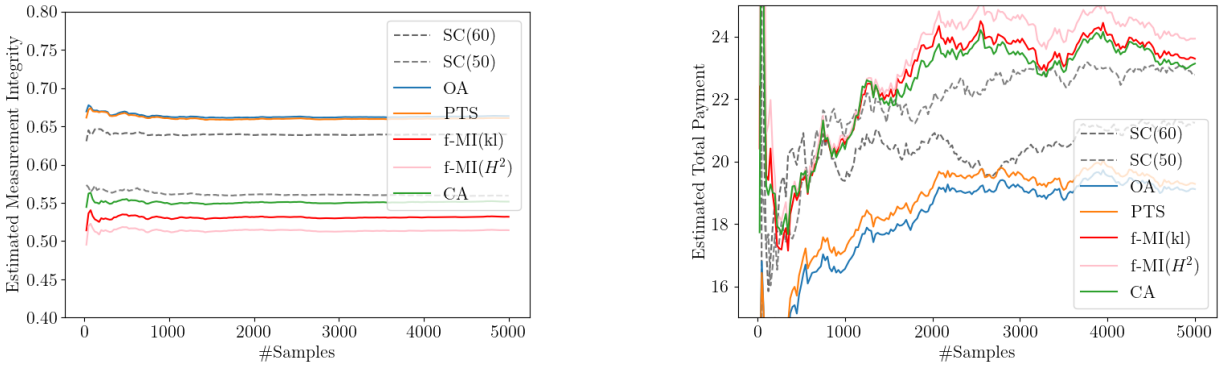
This implies that, at a symmetric equilibrium, if a performance measurement M has Measurement Integrity equal to spot-checking $SCE\%$, then it has a similar motivational proficiency, i.e. similar total payment, to that spot-checking performance measurement within

a tournament payment scheme. Therefore, a higher Spot Check Equivalence indicates higher motivational proficiency.

Note that, even if considering IR, the motivational proficiency is still monotonically increasing with the Spot Check Equivalence, however, when IR is binding, more spot-checking does not further decrease the total payment.

2.5.2.1 Measurement Integrity is a computationally efficient proxy.

In addition, we find that the Measurement Integrity converges significantly faster than the total payment. Figure 2.5 illustrates the variation in both the Measurement Integrity and total payment as the number of iterations goes up.



(a) Measurement Integrity

(b) Total payment of Borda-count

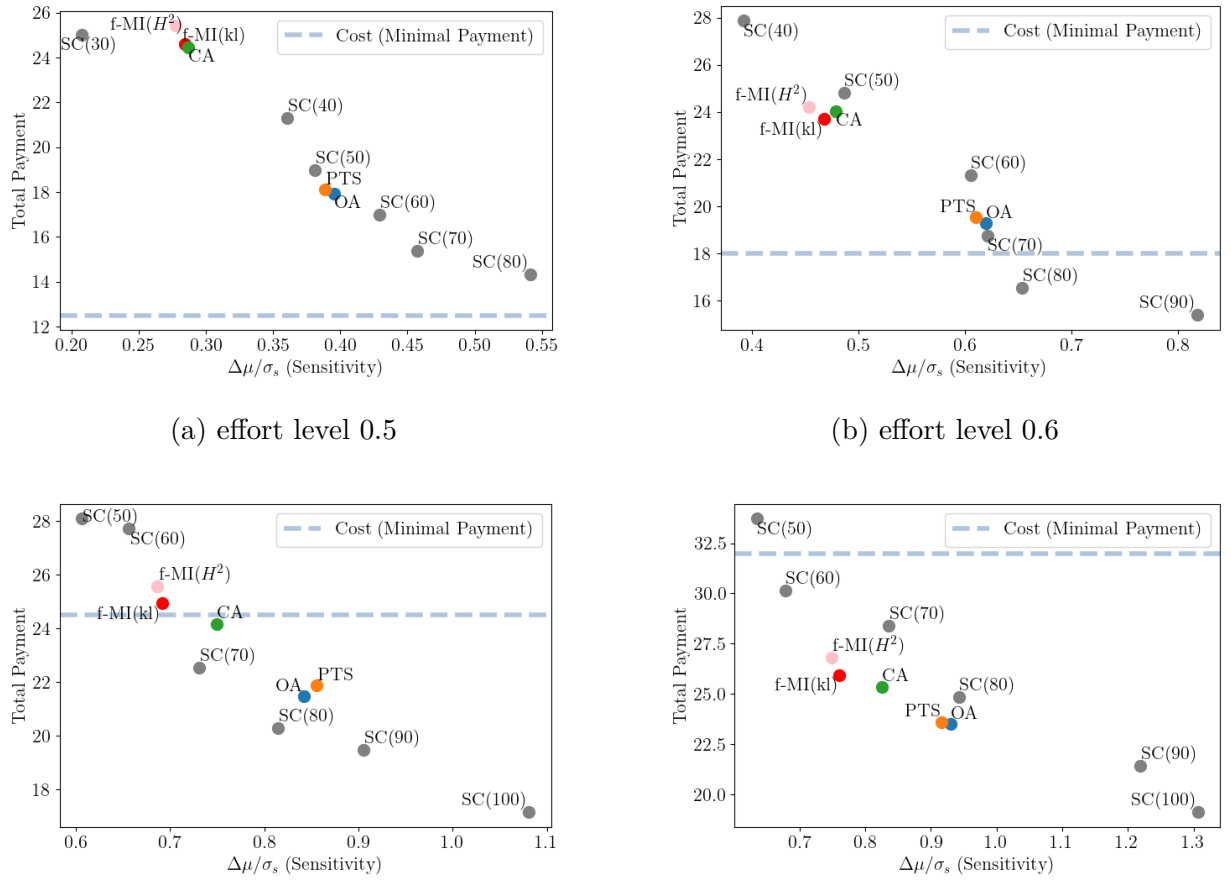
Figure 2.5: Convergence speed of the Measurement Integrity and the total payment: the x-axis is the number of the samples, and the y-axis is the estimated Measurement Integrity and the estimated total payment of the Borda-count payment scheme at effort level $\xi = 0.6$ respectively.

In the previous results of Figure 2.4 (b), we find that when eliciting effort level of $\xi = 0.6$, the f-MI(kl), f-MI(H^2) and CA performance measurements have a little less motivational proficiency comparable to 50% spot-checking. Meanwhile, the OA and PTS performance measurements are better than 60% spot-checking.

Figure 2.5 demonstrates that achieving the same outcome requires significantly fewer iterations for the calculation of Measurement Integrity compared to the total payment. This suggests that even when it's possible to compute the expected total payment (e.g. with agent-based model simulation), utilizing Measurement Integrity as a proxy offers better computational efficiency.

2.5.3 Sensitivity-based Spot Check Equivalence

We now examine the effectiveness of the Spot Check Equivalence based on Sensitivity as a metric of motivational proficiency. Here, when estimating $\Delta\mu/\sigma$ (which is proportional to the Sensitivity), we consider a more realistic scenario where there is only one sample from the data generating process and no ground truth. We estimate $\Delta\mu/\sigma$ for each performance measurement according to Algorithm 2 with $T = 5000$ iterations. We then compare the $\Delta\mu/\sigma$ with the total payment estimated in the same way as in the previous subsection. The results are shown in Figure 2.6.



(a) effort level 0.5

(b) effort level 0.6

(c) effort level 0.7

(d) effort level 0.8

Figure 2.6: Sensitivity v.s Total Payment of Borda-count payment scheme: the x-axis is the $\Delta\mu/\sigma$ and the y-axis is the total payment needed to elicit that equilibrium within the tournament payment scheme. The horizontal line shows the agents' cost to exert the effort level, which implies the minimal payment to satisfy the Individual Rationality.

Similarly, with the results in Figure 2.6, we can observe that: (1) The Sensitivity monotonically increases with the spot-checking ratio. (2) The Sensitivity and the total payment are

significantly negatively correlated. This observation implies that the Spot Check Equivalence based on Sensitivity can be used as a metric of motivational proficiency

2.5.4 Spot Check Equivalence of Peer Prediction

2.5.4.1 Peer Prediction works better to elicit high effort.

We apply the workflow in our agent-based model experiment to further compare the Spot Check Equivalence of different performance measurements in various contexts. Previous works [Zhang and Schoenebeck, 2023b, Burrell and Schoenebeck, 2021, Gao et al., 2016] conduct comparisons of certain metrics in specific contexts. To further study the motivational proficiency of the performance measurements, we calculate the Spot Check Equivalence across various information elicitation contexts.

We then enumerate the effort levels and calculate the Spot Check Equivalence via the Measurement Integrity of each performance measurement (Figure 2.7).

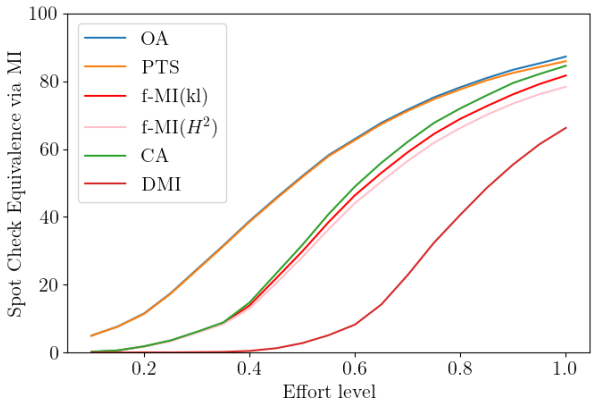


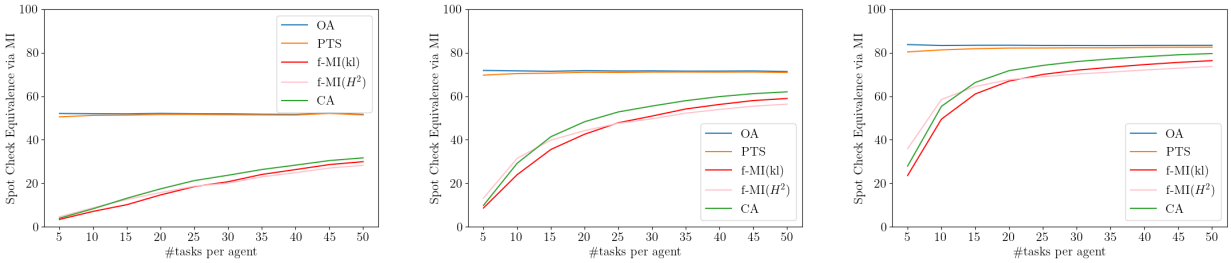
Figure 2.7: Spot Check Equivalence on different effort levels

We find that when eliciting a low-effort equilibrium, the Spot Check Equivalence of peer prediction performance metrics is low, however, the relative motivational proficiency of peer prediction increases fast. That is because a peer prediction performance measurement scores an agent according to the correlation between her report and her peers', when every agent exerts a low effort, her peers' reports are noisy so the score is quite noisy.

2.5.4.2 Mutual-information-based mechanisms work better when #tasks per agent increases.

As the number of tasks per agent increases, the SCE of OA and PTS remains the same, while for the mutual-information-based mechanisms (CA, f-MI), the Spot Check Equivalence

significantly increases. As the #tasks per agent increases, the estimation of the joint distribution of two agents' reports will be more accurate, which leads to a more accurate score.



(a) effort level 0.5 (b) effort level 0.7 (c) effort level 0.9

Figure 2.8: #tasks per agent vs. the Spot Check Equivalence based on Measurement Integrity: the x-axis is the number of tasks assigned to each agent and the y-axis is the Spot Check Equivalence calculated by Measurement Integrity.

2.6 Does Peer Prediction Make Things Worse?

In this section, we provide a detailed discussion about why Gao et al. [2016, 2019] have the result that "peer prediction makes things worse" which contradicts the other literature Zhang and Schoenebeck [2023b] and our main results that show peer-prediction mechanisms can have non-zero Spot Check Equivalence.

We first introduce the Information Elicitation Context in their paper.

Agent $Agent = (I, c, \mathbf{e})$. The agent can choose a binary effort $e \in \{0, 1\}$, where exerting high effort has a cost $c(1) = c^E$ and exerting no effort has a cost $c(0) = 0$.

Application Abstraction In $App = (J, \mathcal{GT}, \omega, \Sigma, D)$, they also model signal generation as a random function:

$$D_{signal} : \{0, 1\} \times \mathcal{GT} \to \Delta\Sigma$$

however, importantly, the signals of agents are not i.i.d. across agents. When the agent exerts high effort, she will get a high-quality signal, which is drawn from a distribution conditional on the task's ground truth. When the agent exerts no effort, she will get a low-quality signal that is uncorrelated with the task's ground truth, but, crucially, the no-effort signals are perfectly correlated across no-effort agents—they all receive the same signal.

Performance Measurement Firstly, they assume that the scoring function in the spot-checking performance measurement (Definition 2.10) can effectively incentivize high effort, i.e.

$$\mathbb{E}\left[S\left(D_{signal}(1, g_j), g_j\right)\right] - c^E > \mathbb{E}\left[S\left(D_{signal}(0, g_j), g_j\right)\right]$$

In addition to the spot-checking performance measurement (Definition 2.10), they propose a spot-checking peer-prediction performance measurement, where for the unchecked tasks, they apply a peer-prediction performance measurement to score the agents.

Payment Scheme They fix a function $f : \mathcal{GT} \times \Sigma \to \mathbb{R}^*$. The payment scheme is additive across tasks and pays agents according to f for answering spot-checked tasks, and according to a peer-prediction mechanism for tasks that are not spot-checked.

Agent reports They allow the agents to strategically report their signals.

Equilibrium They focus on two possible equilibria. In the no-effort equilibrium, each agent exerts no effort and uses the same strategy to report her signal. In the truthful equilibrium, each agent exerts high effort and truthfully reports her signal.

2.6.1 Comparing Spot-checking Mechanisms and Spot-checking Peer-prediction Mechanisms

Notice that, by assumption, with enough spot-checking, you will get the truthful equilibrium. They use the minimum spot-checking ratio that ensures the truthful strategy profile is an equilibrium as a measure of the mechanisms' performance. Formally, they use p_{Pareto} to denote the minimum spot-checking ratio where the truthful equilibrium Pareto dominates the no-effort equilibrium when applying the hybrid peer-prediction/spot-checking mechanism. They use p_{ds} to denote the minimum spot-checking ratio where the truthful strategy profile is a dominant strategy in spot-checking mechanism.

Then they propose a theorem for comparing the spot-checking mechanism and the spot-checking peer-prediction mechanism:

Theorem 2.16 (Section 5 Theorem 3 in Gao et al. [2016])**.** *For any spot-checking peer-prediction mechanism, if the no-effort equilibrium exists and Pareto dominates the truthful equilibrium when the cost of effort is 0 ($c^E = 0$) and no task is checked ($p = 0$), then $p_{Pareto} \geq p_{ds}$ for any $c^E \geq 0$.*

There are three differences between the models in Gao et al. [2016] and Zhang and Schoenebeck [2023b] that account for this stark difference.

The most obvious difference is that the payments in Gao et al. [2016] are restricted to be additive across tasks. This is rather similar to assuming a linear payment rule when the score is additive across tasks. However, Zhang and Schoenebeck [2023b] use tournaments, which are not linear and typically have much better motivational proficiency.

Second, Gao et al. [2016]'s assumption of a fixed payment function f is extremely restrictive. As we discussed in Section 2.1, when the spot-checking ratio decreases, it is possible to maintain the same incentive properties by simply scaling up the payment function. Thus, if we scale up f, we can make p_{ds} arbitrarily small, and, conversely, by scaling down f, we can make p_{ds} arbitrarily close to 1. On the other hand, in Zhang and Schoenebeck [2023b] the mechanism is defined as a performance measurement and a payment scheme, and the payment scheme is optimized to work with the scoring function, rather than being artificially fixed.

Finally, Gao et al. [2016] make a key assumption on the no-effort signals being perfectly correlated. For example, in peer grading, the writing and formatting quality is a signal that can be accessed with very little effort while assessing the correctness both requires more effort and will likely lead to less agreement.

Notice that the premise of Theorem 2.16 is that when the cost of effort is 0 ($c^E = 0$) and no task is checked ($p = 0$), the equilibrium where agents exert no effort Pareto dominates the truthful equilibrium.

Let's zoom in on this. First, consider the case where the cost of the high effort signal $c^E = 0$. Notice that any spot-checking mechanism that checks any positive ratio of tasks will have the high-effort profile as an equilibrium because agents will receive some positive payoff, but have 0 cost.

Next, consider a peer-prediction mechanism (e.g. a hybrid peer-prediction/spot-checking mechanism with spot-checking ratio 0). Again, the theorem is vacuously true, unless the profile in which no agent exerts effort Pareto dominates the high-effort equilibrium because, in the former, all agents agree and receive a maximal payoff[3], but, in the latter, they do not all agree.

Together, this shows that any peer prediction mechanisms will have a Spot Check Equivalence of 0, since in this case, any spot-checking ratio that is greater than 0 will inevitably lead to the truthful equilibrium given that the effort cost is nonexistent.

Because having a Spot Check Equivalence of 0 is an assumption of the theorem, it is no wonder that such peer-prediction mechanisms do not help!

However, this still potentially makes for a very strong critique of peer-prediction mech-

[3]They are assuming here that complete agreement brings a maximum payoff.

anisms because in many real-world settings, there exist cheap signals. For example, as mentioned in the introduction, when humans are labeling LLM responses, it is much easier to judge them on how authoritative-sounding the responses are than on how truthful the responses are. However, such labels may encourage hallucinations.

Indeed, Gao et al. [2016] led to several papers trying to create peer-prediction mechanisms that are robust against "cheap" signals (i.e. the no-effort/low-effort signals that can bring higher agreement than the high-effort signals).

Kong and Schoenebeck [2018a] propose a peer prediction mechanism called Hierarchical Mutual Information Paradigm (HMIP), assuming a hierarchical information structure where high-effort (or higher expertise) agents have access to the information of low-effort (or lower expertise) agents. HMIP encourages agents to invest effort and incentivizes truthful reporting by paying the high-effort agents for correctly predicting the "cheap" signals from the low-effort agents. Additionally, a human subject experiment [Kong et al., 2022] shows evidence of a hierarchical information structure among the participants.

2.7 Related Work

Besides the theoretical literature discussing peer prediction mechanisms, as highlighted in Section 2.5, there are empirical studies that validate these mechanisms. For instance, Radanovic et al. [2016] experimentally tested their peer prediction mechanism in both peer grading and crowdsourcing scenarios to validate its theoretical properties. Similarly, Shnayder et al. [2016] employed peer grading data from the edX MOOC platform to assess the performance of their proposed mechanisms. Spot checking in peer grading scenarios has already been empirically examined in works such as Zarkoob et al. [2019], Wright et al. [2015]. Additionally, Goel and Faltings [2019] study combining peer prediction and spot-checking, and introduce the Deep Bayesian Trust Mechanism that utilizes peer reports to reduce the need for spot-checking.

Furthermore, in forecasting contexts where agents are rewarded afterward based on the agreement between their forecasts and the outcomes, i.e. the ground truth is accessible for little or no cost, Hartline et al. [2022], Li et al. [2022], Neyman et al. [2021] study how to optimize proper scoring rules to incentivize effort, and consequently, elicit high-quality information. These works suggest the possibility of optimizing the spot-checking mechanisms by scoring the checked tasks according to the optimal proper score rules, which indicates possible direction for future research.

2.7.1 Discussion of the truthful report assumption.

In our Section 2.2, we assume that the agents will truthfully report their signal. This assumption is reasonable when applying a linear payment scheme given the performance measurement is truthful under certain settings. In particular, Dasgupta and Ghosh [2013] propose the first multi-task peer prediction mechanism. Later on, the CA mechanism Shnayder et al. [2016] and f-MI mechanism Kong and Schoenebeck [2019] are proposed to generalize Dasgupta and Ghosh [2013]'s mechanism from binary signal space to finite signal space, the MA mechanism Zhang and Schoenebeck [2023a] generalizes CA to address task-dependent strategies, the DMI mechanism Kong aims to reduce the required number of tasks, Schoenebeck and Yu [2020] handle continuous signals with a learning-based mechanism, Agarwal et al. [2017] deal with the heterogeneous agents, and Schoenebeck et al. [2021] address adversarial attacks in peer prediction.

Furthermore, Burrell and Schoenebeck [2021] examine the robustness of the truth-telling equilibrium with agent-based model experiments. However, when applying a non-linear payment scheme, e.g. winner-take-all, the agents may have the incentive to strategically report their signal, e.g. increasing the variance of their score to get a higher probability of being the winner. Zhang and Schoenebeck [2023b] propose a truthful winner-take-all payment scheme by adding noise to the agents' score which may hurt the incentive for effort. However, further study needs to be conducted to study the robustness of different performance measurements against strategic reports with other non-linear payment schemes. This gap indicates another potential future direction of our research.

2.8 Conclusion and Discussion

In summary, our research provides a methodology for understanding the performance, especially motivational proficiency, of information elicitation mechanisms in various contexts, the Spot Check Equivalence, and consequently offers valuable insights for the design of effective and efficient incentive mechanisms that promote the acquisition of high-quality information.

Future research might be conducted to investigate motivational proficiency in a non-monetary setting, e.g. in peer grading, we care about how to elicit agents' effort with the bounded individual payoff, since the students' grades could only be A, B, C, F, etc. Another future direction might be to study motivational proficiency in a more sophisticated model where the agents have heterogeneous cost functions.

CHAPTER 3

Eliciting Informative Text Evaluations with Large Language Models

3.1 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.

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

[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).

... 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. ...

Figure 3.1: 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.

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.

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, Shnayder et al., 2016, Kong and Schoenebeck, 2019, Liu et al., 2023, Zhang and Schoenebeck, 2023a]. In certain settings such as multiple choice questions, even a small number of tasks may suffice Schoenebeck and Yu [2020], Burrell and Schoenebeck [2021], Kong [2024a]. The second approach called the signal-prediction framework, pioneered by Prelec [2004], 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' " [Radanovic and Faltings, 2014, Witkowski and Parkes, 2012, Schoenebeck and Yu, 2023b, Chen et al., 2021].

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.

3.1.1 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 heuristic implementations for getting the LLM-prediction—Token and Judgment. 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.

3.2 Related Work

Peer prediction mechanisms The idea of peer prediction, introduced by Miller et al. [2005], proposes a mechanism that induces truth-telling as an equilibrium, assuming the mechanism knows the joint distribution of agents' signals and this knowledge is common knowledge among agents. This fundamental assumption, however, might be challenged in many settings. There have been two primary research directions that can mitigate it.

One primary direction is to generalize peer prediction to a multi-task setting, making it possible to estimate (implicitly or explicitly) the joint distribution between the reports with multiple tasks Dasgupta and Ghosh [2013], Shnayder et al. [2016], Kong and Schoenebeck [2019], Schoenebeck and Yu [2020], Zhang and Schoenebeck [2023a], Kong [2020], Agarwal et al. [2017], Schoenebeck et al. [2021], Zhang et al. [2026]. The other direction follows another mechanism, the Bayesian Truth Serum (BTS) Prelec [2004], which also seeks to incentivize truthful reporting as an equilibrium. It directly solicits predictions from agents, i.e., each agent predicts the reports of their peers, bypassing the need for prior knowledge of distributions. Subsequent research has extended the BTS, enhancing its efficacy in diverse scenarios Witkowski and Parkes [2012], Radanovic and Faltings [2013], Zhang and Chen [2014], Schoenebeck and Yu [2023a].

Our contribution markedly diverges from these established approaches in several ways. First, we focus on incentivizing text-based human responses, which entails a highly complex response space, where both estimating the joint distribution and directly eliciting predictions from agents are no longer practical. Second, we introduce the first peer prediction mechanism that employs a Large Language Model (LLM) as an oracle, which is presumed to effectively approximate the joint distribution of responses. As LLM capabilities enhance, our mechanism naturally improves, benefiting from more accurate oracle estimations of joint response distributions. Overall, our approach enables the deployment of peer prediction mechanisms in single-task settings without the necessity of directly soliciting predictions or possessing prior distribution knowledge.

Shortcuts in information elicitation Geirhos et al. [2020] use "shortcut learning" to

describe a machine learning algorithm's reliance on unintended, simpler features rather than learning the intended, complex representations. For example, a deep neural network might only identify cows accurately within grassy scenes, mistaking "grass" for "cows" Beery et al. [2018]. Niven and Kao [2019] show that a language model, BERT Devlin et al. [2018], might seem to understand reasoning but actually relies on spurious statistical cues in the dataset, particularly the presence of specific words, failing when these cues are absent.

This phenomenon not only exists in artificial intelligent agents but also in human agents. As accessing and processing the information may cost agents a considerable amount of effort, human agents also rely on shortcuts or heuristics. A case in point is peer review processes, where reviewers might focus primarily on assessing the story-telling in the introduction as a shortcut, instead of providing a deeper, more insightful evaluation of the work's merits.

To address this issue and elicit high-quality information rather than superficial shortcuts, researchers have proposed several approaches. Li et al. [2022] propose optimal scoring rules that maximize the marginal expected payoff increase of an agent who exerts a higher level of effort. Zhang and Schoenebeck [2023b] suggest that paying the agents according to a tournament based on their scores could also help the efficiency to elicit high effort. When applying peer prediction mechanisms in a peer grading setting, Gao et al. [2016] finds a Pareto-optimal Nash equilibrium where all agents exploit a low-effort shortcut. To address this problem, Kong and Schoenebeck [2018a] propose a peer prediction mechanism robust against such shortcuts, by explicitly eliciting both the intended response and the shortcut, and then conditioning the shortcut out. Our GSPPM is inspired by Kong and Schoenebeck [2018a] but focuses on text-based responses and does not require eliciting additional information from the agents.

Additionally, how to robustly aggregate information when there exists (implicitly or explicitly) a shortcut is also widely studied Prelec et al. [2017], Pan et al. [2023], Chen et al. [2021], Guo et al. [2024], Arieli et al. [2018], Palley and Soll [2019], Lu and Kong [2024], Kong et al. [2022]. A well-known task is "Is Philadelphia the capital of Pennsylvania?" where the majority will take the shortcut that Philadelphia is a famous big city and big cities are usually the capital, thus making mistakes in this question. Prelec et al. [2017] provide a solution to this by eliciting the agents' prediction and aggregating the "surprisingly popular" choice.

LLM applications in peer review The core of our experiments relies on using LLMs to analyze peer review data on ICLR and customer review data on Yelp. Recent works claim that LLMs have great potential to support publication Gao et al. [2022], Salvagno et al. [2023] and review practices Liang et al. [2023] if properly used. In a survey, it is shown

that GPT-generated reviews are appreciated by over 50% of participants, and over 80% of participants found them more beneficial than feedback from at least some human reviewers Liang et al. [2023]. More evidence exists showing that LLMs can be used to handle easy evaluation tasks in peer review, such as summarizing methodologies and assessing writing styles Donker [2023], detecting gender bias Verharen [2023], and verifying author checklists Liu and Shah [2023]. However, other work expresses concerns about the ability of LLMs to handle more complicated reviewing tasks. For example, hallucination occurs when LLMs are asked to provide suggestions for specific improvements Donker [2023], and LLMs tend to perform poorly on distinguishing clearly "better" papers Liu and Shah [2023]. Hosseini and Horbach [2023] provide some recommendations for the proper use of LLMs to support review or editorial processes.

Our work is largely orthogonal to these concerns. Rather than using LLMs to generate reviews directly, our paper focuses on using LLMs as a tool to design automated mechanisms that incentivize high-quality reviews. If the quality of LLM reviews dominated the quality of high-effort human reviews, our work would largely be superfluous, but LLMs cannot yet do this. While LLM-generated reviews may provide additional feedback, for the moment, subjective high-quality human reviews are still essential.

3.3 Preliminaries

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

3.3.1 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 Miller

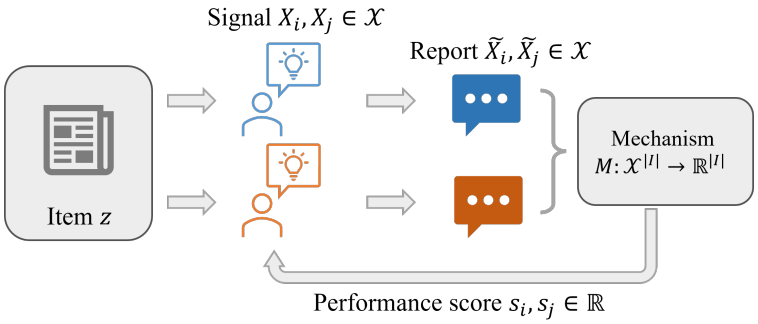


Figure 3.2: An Overview of Our Information Elicitation Model

et al. [2005], Dasgupta and Ghosh [2013], 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, when agents evaluate the same item, their signals are expected to be related 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 3.1 (Stochastic Relevance)**.** *For any $x_i, x_i' \in \mathcal{X}$ such that $x_i \neq x_i'$, there exists $x_j \in \mathcal{X}$ such that*

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

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.

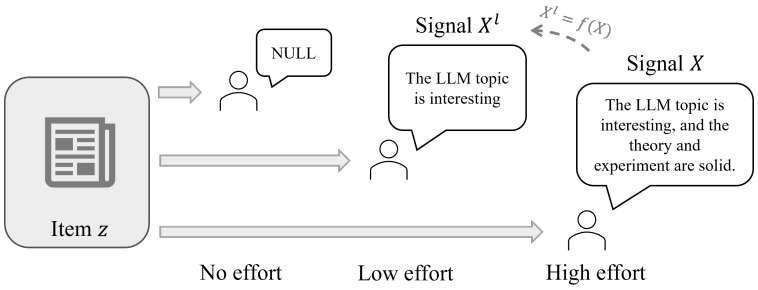


Figure 3.3: An Example of the Three-Level Effort Model

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 [2018a], 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.

Note that our model can also capture a binary effort setting which is assumed in some previous work Miller et al. [2005], Gao et al. [2016], 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 3.2 (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 3.4).*

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 3.3 (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* $x \in \mathcal{X}$.

[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., $H(X \mid Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x, y) \log \left(\frac{P(x,y)}{P(y)} \right)$.

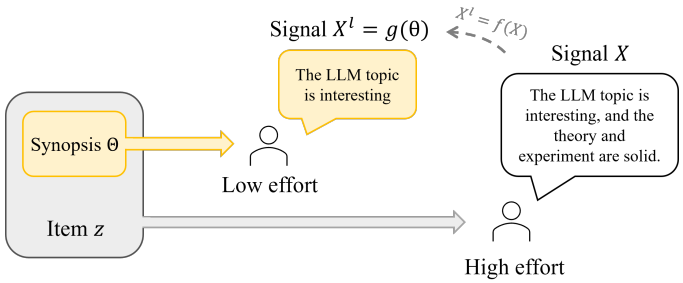


Figure 3.4: An Example of the Synopsis-determined Low-effort Signals

Note that Synopsis-determined and Synopsis-covering are not mutually exclusive. We will use these definitions in Corollary 3.16, but Propositions 3.12, 3.13, 3.14 and 3.15 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. We assume that the utility of agent i is the difference between her payment and her cost of effort, i.e., $u_i = \alpha \cdot s_i + \beta - c_i$.

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

3.3.2 Mechanism Design Goal

We first define the solution concept.

Definition 3.4. *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 3.5 (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.*

3.3.3 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 3.6 (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* Selten [1998], Gneiting and Raftery [2007], 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 (such as the Log Scoring Rule).

Definition 3.7 (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 3.8 (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.

3.4 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.

3.4.1 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 $\Pr_{\text{LLM}(\psi)}$ and refer to it as the *LLM-prediction*. Thus, $\Pr_{\text{LLM}(\psi)}[\tilde{x}]$ 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 , (which it often is as in Figure 3.5), to be consistent with the classic information elicitation model, we use $\Pr_{\text{LLM}(\psi)}[X_j = \cdot \mid Y = y]$ to denote the LLM-prediction $\Pr_{\text{LLM}(\psi(y))}[\cdot]$. We are particularly interested in two predictions. First, $\Pr_{\text{LLM}(\psi)}[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i]$ is the LLM-prediction of agent j's report while integrating agent i's report into the prompt. Second, $\Pr_{\text{LLM}(\psi)}[X_j = \tilde{x}_j \mid X_i = \tilde{x}_i, \Theta = \theta]$ 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 3.9 (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 $\theta \in \Sigma$:*

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

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

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

Assumption 3.9 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 3.5, 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.

3.4.2 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 3.10 (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 3.9 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 3.4.4 while providing the high-level intuitions of why GPPM is potent here.

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

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

details of the particular paper such as an elegant proof and a potential broader impact of the method.

3.4.3 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. 3.1, 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 3.11 (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*

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

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 [2018a], which also used conditioning to motivate high-effort signals above low-effort signals.

3.4.4 Theoretical Results: GPPM and GSPPM are ε-Potent

Here, we provide formal theoretical guarantees of our mechanisms under Assumption 3.9. 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.

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

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

Proposition 3.12. *For the Generative Peer Prediction Mechanism (GPPM), when Assumption 3.9 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)

$$\mathbb{E}[GPPM(\tau(X_i), X_j)] - \mathbb{E}[GPPM(\sigma(\text{Null}), X_j)] \geq I(X_i; X_j) - \epsilon.$$
(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 3.13. *For the Generative Synopsis Peer Prediction Mechanism (GSPPM), when Assumption 3.9 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 3.9), 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 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. We defer the proof to Section C.1.

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.

Formally, for the approximately potent properties, we have the following propositions. Again, we defer the proof to Section C.1.

Proposition 3.14. *For the Generative Peer Prediction Mechanism (GPPM), when Assumption 3.9 holds with parameter $\epsilon \geq 0$. When $I(X_i; X_j \mid X_i^l) > 0$, GPPM is $\alpha\epsilon$-potent, where*

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

When $\epsilon = 0$, GPPM is potent.

Proposition 3.15. *For the Generative Synopsis Peer Prediction Mechanism (GSPPM), when Assumption 3.9 holds with parameter $\epsilon' \geq 0$. When $I(X_i; X_j \mid \Theta, X_i^l) > 0$, GSPPM is $\alpha\epsilon'$-potent, where*

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

When $\epsilon' = 0$, GSPPM is potent.

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 $\text{Gap}(h, l)$ denote the difference in the expected score of agent i between exerting high effort and exerting low effort while both agents report truthfully. Let $\text{Gap}(h, \text{Null})$ be the analogous notation for the gap between exerting high effort and exerting no effort. Combining proposition 3.12 and proposition 3.13 together, we have the following corollary.

Corollary 3.16. *If the low-effort signals are synopsis-determined (definition 3.2) and synopsis-covering (3.3), 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.*

Corollary 3.16 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 3.7.4.

3.5 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 Section B.3. The full implementations and results are available online.

Table 3.1: A high-level comparison of three implementations.

Implementation	Require NN access	LLM usage
Token-Raw	Yes	Generate log-probability
Token-Preprocess	Yes	Preprocess, generate log-probability
Judgment	No	Distill report into judgments, predict judgments

3.5.1 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.

Prompt $\psi(\tilde{x}_i)$: You are the second reviewer for a scientific paper. You are given a peer review from the other reviewer: [**Review** $\tilde{x}_i$] Your task is to provide your own judgments of the paper based on the given materials.

Response: [Predicted Review $\tilde{x}_j$]

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

Figure 3.5: A simplified example of a prompt for TOKEN in academic peer review scenario. In Appendix B.4, we present all exact prompts used in our experiment.

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 within X. Again, we use $\tilde{x}^{(k)}$ to denote the value of the k-th token in a report.

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 $\Pr_{\text{LLM}(\psi)}[X^{(k)} \mid X^{(l)} = \tilde{x}^{(l)}\ \forall l \in \{1, 2, \ldots, k-1\}]$. With Bayes' rule, we can write the probability of the occurrence of a given report $\tilde{x}$ as:

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

3.5.1.1 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

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

3.5.1.2 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 involves employing 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. The details of our preprocessing prompt are presented in Section B.4.1 for reference.

Formally, the performance score of agent i using the implementation 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 Section B.2.1. 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.

3.5.2 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. For example, the popular OpenAI GPT-4 chat completion API can only output sampled textual responses rather than the probability of tokens. We propose an alternative method that uses an LLM as a black box. Our idea is to first summarize each report as a set of judgments and then predict the peer's report by predicting the probability of each of its judgments 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 provide exact prompts used in our experiment in Appendix B.4. 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 3.10. 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.

3.6 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 3.3.1 as we focus on testing the efficacy in a realistic scenario without relying on the theoretical assumptions. We finally introduce the statistical metrics used to measure the significance.

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 score will decrease in expectation. To test this, we apply a t-test to evaluate whether the decrease in score from s_+ to s_- is statistically significant, thereby confirming the sensitivity of our methods to manipulations.

Formally, we describe the implementation of the above workflow in Algorithm 3.

ALGORITHM 3: Evaluation Workflow

Input: A dataset with items and associated text reports. A mechanism M computing the peer prediction scores. A low-quality report generation process D.

Output: Lists $\mathbf{s}_+$ and $\mathbf{s}_-$ and the statistics metrics.

for $k = 1$ **to** K **do**

- Draw an item z u.a.r. from all the items;
- Draw two text reports $\tilde{x_i}$ and $\tilde{x_j}$ u.a.r. from all reports related to z ;
- Compute $s_+^{(k)} := M(\tilde{x_i}, \tilde{x_j})$;
- Replace the text report $\tilde{x_i}$ with $\tilde{x_i}^l$ according to process D;
- Compute $s_-^{(k)} := M(\tilde{x_i}^l, \tilde{x_j})$;

end

Add the computed values to lists $\mathbf{s}_+ = \{s_+^{(k)}\}$, $\mathbf{s}_- = \{s_-^{(k)}\}$;

Compute the statistics metrics based on lists $\mathbf{s}_+$ and $\mathbf{s}_-$;

3.6.1 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 the 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 3.7.2) can also be used to test whether they can incentivize truth-telling.

3.6.2 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. The prompt for this task is presented in Section B.4.4.

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 3.6.1, 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

[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.

the original signal. Thus, our focus remains on assessing the effectiveness of differentiating various quality levels of reports across different mechanisms.

3.6.3 Statistics Metrics

The principal criterion for our method's effectiveness is a statistically significant decrease in scores following the degradations. To empirically validate this criterion, we conduct statistical tests on the samples of scores computed with the original reports (s_+) and the degraded/LLM-generated reports (s_-). Specifically, we employ the t-test, which is able to identify whether there is a significant decrease in scores from s_+ to s_-. Our empirical distributions (Section 3.7.5) show that the measured differences ($s_+ - s_-$) are approximately normally distributed, confirming our data is suitable for the t-test. Below, we introduce the detailed implementation of the t-test.

Paired Difference t-test For each pair of scores (s_+, s_-), we use the paired difference t-test to verify if the mean of their difference is greater than 0. Let $d_k := s_+^{(k)} - s_-^{(k)}$, and $\bar{d}$ denote the mean of the differences d_k. Formally, the hypotheses for the Paired Difference t-test are:

- Null Hypothesis (H_0): The mean difference $\bar{d}$ is equal to 0.
- Alternative Hypothesis (H_1): The mean difference $\bar{d}$ is greater than 0.

The test statistic for the Paired Difference t-test is calculated as follows:

$$t = \frac{\bar{d}}{\sigma_d / \sqrt{K}}$$

where σ_d is the standard deviation of the differences $\{d_t\}$, and K is the number of pairs. Furthermore, we denote $SE(\bar{d}) = \frac{\sigma_d}{\sqrt{K}}$, which is the standard error of the mean difference.

The p-value is obtained by $p = \Pr[X > t]$, where X is a random variable following the t-distribution with the calculated degrees of freedom $K - 1$, and t is the calculated t-statistic above. Typically, a p-value threshold of 0.05 is used, where a p-value lower than 0.05 leads to the rejection of the null hypothesis, indicating that there's a statistically significant difference.

3.7 Experiments

This section presents the setup of our experiments and the empirical results.

3.7.1 Experiment Setup

3.7.1.1 Datasets

We use two datasets for our experiment.

Yelp Online Review Data (Yelp) is publicly available online review data from Yelp. The restaurants featured on Yelp are interpreted as the "items" in our model, and customer reviews for each restaurant are interpreted as the "text reports" for our model. 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] In this dataset, the "items" are the submitted papers, and the "text reports" are the peer reviews they received. Given the typically longer and more informative nature of ICLR reviews compared to Yelp reviews, we choose a smaller sample size for the ICLR dataset to manage computational demands efficiently. Thus, we construct our dataset by randomly sampling 300 items (papers) from the entire dataset. 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.

3.7.1.2 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.

[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 3.6, 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.

Llama-2 For the TOKEN implementations, including both TOKEN-PREPROCESS and TOKEN-RAW (As discussed in Section 3.5.1, we defer the results for TOKEN-RAW to Section B.2.1), 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.

Table 3.2: Table illustrating the use of different LLM models in various tasks

Task	LLM Usage	Notes
Preprocessing	ICLR: gpt-4-1106-preview Yelp: gpt-3.5-1106-preview	Discussed in footnote[6]
TOKEN implementations	llama-2-70b-chat	Open source LLM
JUDGMENT implementation	gpt-4-1106-preview	/
Generating LLM reviews	gpt-3.5-1106-preview gpt-4-1106-preview	Discussed in Section 3.6.2

We provide all prompts used, along with example inputs and outputs, in Section B.4.

3.7.1.3 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 3.6.1) and LLM-generated-review evaluation (Section 3.6.2).

GSPPM. We test the GSPPM on the ICLR dataset. This is because the Yelp dataset we use lacks detailed features of the items (restaurants) and thus we do not have access to a suitable synopsis of each item. For the ICLR dataset, we consider the abstract of a paper as the commonly-known synopsis. Conditional on this, we apply the LSR to calculate the score in the GSPPM.

We only apply the TOKEN-PREPROCESS implementation for two reasons: First, when we have access to a neural network of an LLM, TOKEN-PREPROCESS implementation is more time- and cost-efficient than JUDGMENT, given there is no need for API calls. Second, the results in GPPM indicate that TOKEN-PREPROCESS generally works better than JUDGMENT on the ICLR dataset.

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 3.6.2) to test the GSPPM.

Baseline. Additionally, we present a baseline mechanism (Definition 3.7) that uses only the numerical ratings from reports. This approach rebuilds the joint distribution of two ratings[7] based on historical data and assigns a log scoring rule to agent i based on her rating compared to agent j's rating. 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.

3.7.2 Result 1: GPPM Effectively Penalizes Report Degradations.

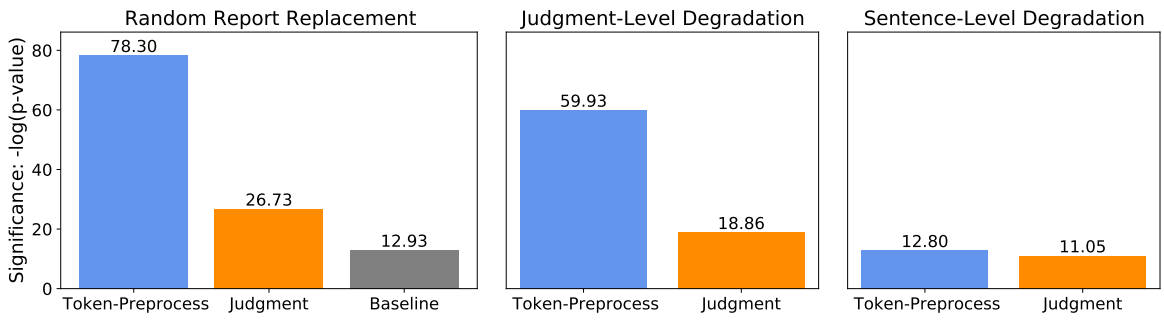
We now delve into our main results. We first describe the results of evaluating the GPPM with three report degradations defined in Section 3.6.1. We apply sample sizes of $K = 500$ and $K = 1000$ for the experiments on the ICLR and the Yelp datasets respectively. We use both Token-Preprocess and Judgment to compute the estimated conditional probability. We visualize the p-values in Figure 3.6 and defer the comprehensive statistics metrics to Section B.2 (Table B.2, B.3 and B.4).

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.

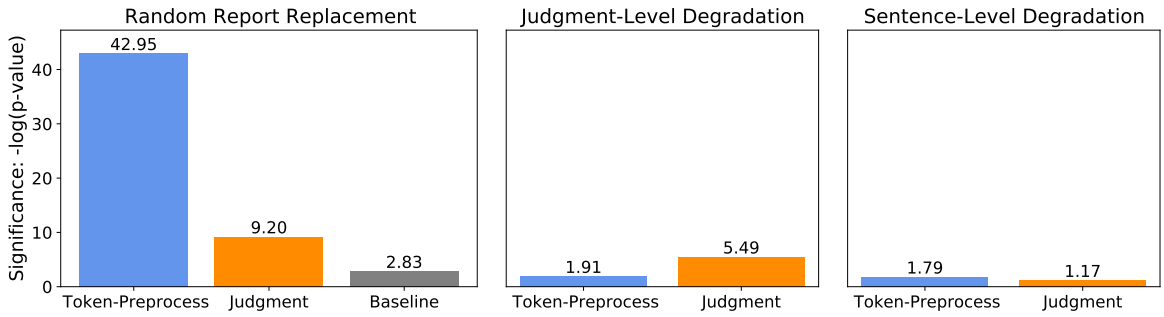
Token outperforms Judgment on the ICLR dataset. Compared with the Judgment implementation, we observe a higher significance score (equivalent to lower p-values) of Token-Preprocess implementations on the ICLR dataset. However, Token-Preprocess does not perform well on the "judgment-level degradation" 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 Section B.2. 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

Figure 3.6: 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 $-\log_{10}$(p-value) > 1.3 (equivalent to p-value < 0.05) is regarded as significant difference.

3.7.3 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 3.6.2. 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 3.7 and provide the statistics metrics in Table 3.3.

In Table 3.3, 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 3.3, the scores are on the same scale as they are computed by the same mechanism. Thus, we observe that while the standard

Table 3.3: 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.

Mechanism	Review-Gen. 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

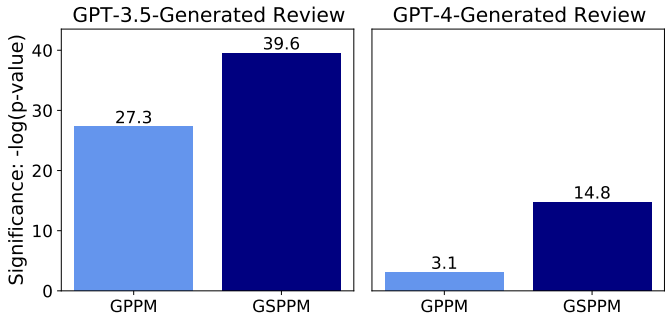


Figure 3.7: LLM-generated Review Evaluation Result for GPPM and GSPPM with TOKEN-PREPROCESS Implementation: Significance ($-\log_{10}$(p-value)) of the expected score difference $\bar{d} > 0$, higher is better. Typically, a significance score $-\log_{10}$(p-value) > 1.30 (equivalent to p-value < 0.05) is regarded as significant difference.

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 3.6.2, 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.

3.7.4 Result 3: GSPPM Penalizes LLM-Generated Peer Review More than GPPM.

We compare the efficacy of GSPPM and GPPM in differentiating between human-written and LLM-generated reviews. As shown in Corollary 3.16, we expect that the GSPPM can do a better job than the GPPM.

Note that the performance scores are no longer on the same scale. Therefore, we focus on the significance scores in Figure 3.7. 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.

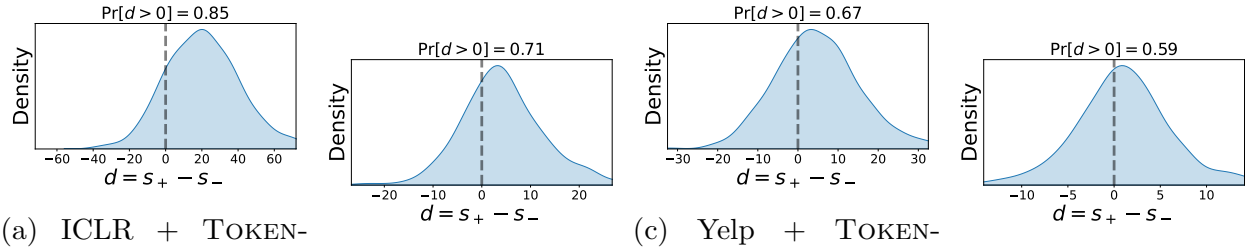
3.7.5 Empirical Distribution of the Performance Score Change

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.

Here, we use kernel density estimation (KDE)[8] 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.

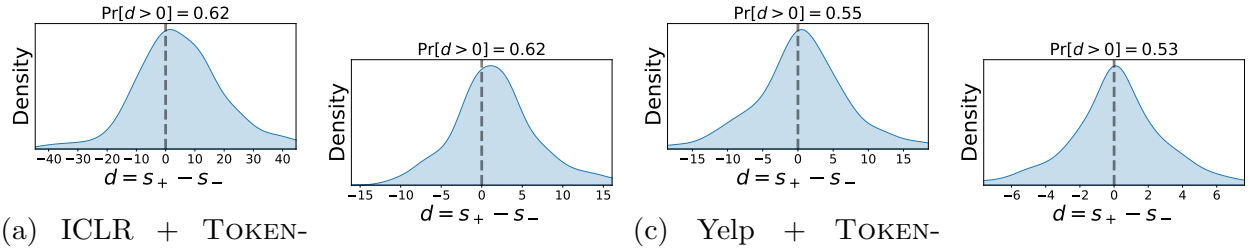
[8]We use seaborn.kdeplot (seaborn.pydata.org/generated/seaborn.kdeplot.html) with default parameters to plot the KDEs.

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

Figure 3.8: Empirical Distribution of the Performance Score Change of Random Signal Replacement Evaluation.



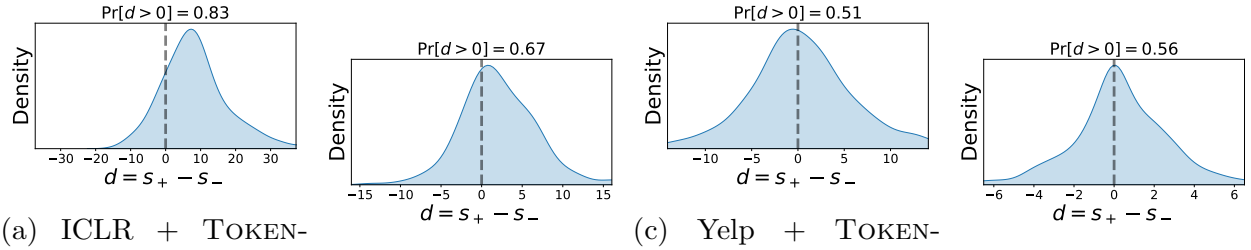
(a) ICLR + Token-Preprocess (b) ICLR + Judgment (c) Yelp + Token-Preprocess (d) Yelp + Judgment

Figure 3.9: Empirical Distribution of the Performance Score Change of Sentence-Level Degradation Evaluation.

3.8 Limitation, Future Work, and Conclusion

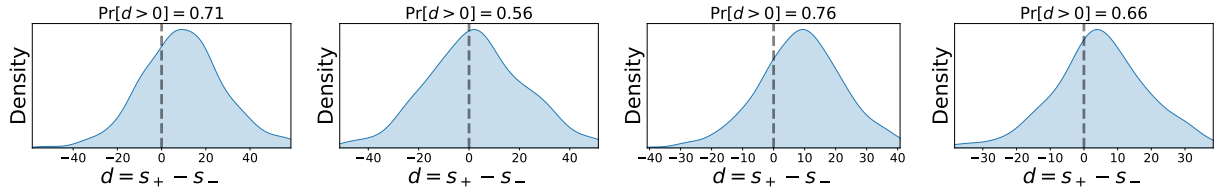
Limitations As the first paper, to our best knowledge, exploring eliciting subjective textual data with LLMs, our work has several limitations, and each of them may lead to a future direction.

First, the theoretical effectiveness of the GPPM and GSPPM highly depends on how well the LLM prediction estimates the real underlying distribution (Assumption 3.9). The quality of this prediction can be influenced by many factors, including prompt engineering,



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

Figure 3.10: 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

Figure 3.11: Empirical Distribution of the Performance Score Change of LLM-Generated Review Evaluation. This result corresponds to Table 3.3.

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 primarily focused on integrating LLMs with Miller et al. [2005]'s mechanism. Future directions may involve applying our method to a broader set of classic peer prediction mechanisms, especially in the multi-task setting Dasgupta and Ghosh [2013], Shnayder et al. [2016], Kong and Schoenebeck [2019], Schoenebeck and Yu [2020], Zhang and Schoenebeck [2023a], Kong [2020], Agarwal et al. [2017], Schoenebeck et al. [2021]. 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.

Furthermore, in addition to the potent property, prior work has investigated other desiderata, including fairness Burrell and Schoenebeck [2021] and efficiency Zhang and Schoenebeck [2023b], Xu et al. [2024]. 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.

Evaluating LLMs on various benchmarks has become crucial in natural language processing (NLP) research. However, several studies have highlighted potential issues of data contamination and data leakage, where the testing data may have been inadvertently included in the training data for LLMs Sainz et al. [2023], Deng et al. [2023]. These issues question the validity of current benchmarks and necessitate the exploration of alternative approaches.

One potential benchmark that mitigates data contamination is to evaluate LLMs on creating peer reviews for the latest research papers. With thousands of new research papers published annually across various fields, these papers are unlikely to appear in the pre-training data, providing a large number of novel test cases. While previous studies have evaluated LLMs' ability to generate new content, such as writing stories Chakrabarty et al. [2024], generating crowdsourcing responses Boussioux et al. [2023], and summarizing news articles Zhang et al. [2024], a common limitation has been the reliance on human evaluators to assess the LLM-generated content. This reliance on manual evaluation can be time-consuming, expensive, and potentially subjective, hindering the scalability and generalizability of these benchmarks.

In the context of academic peer review, existing approaches have focused on analyzing the fraction of common opinions among human and LLM-generated reviews Liang et al. [2023] or studying the correlation between LLM review scores and human scores Thelwall [2024]. However, a more sophisticated benchmark is needed to comprehensively evaluate LLM performance in this domain.

To address this need, our proposed mechanism provides an automated scoring method for the quality of LLM-generated academic peer reviews. By demonstrating the ability to differentiate between reviews from humans, GPT-4, and GPT-3.5, this mechanism shows

the potential to serve as an automated benchmark for peer review tasks. This approach offers several advantages, including efficient and scalable evaluation, mitigation of subjectivity associated with human evaluation, and a more standardized and reproducible framework for comparing LLM performance.

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.

CHAPTER 4

Benchmarking LLMs' Judgments with No Gold Standard

4.1 Introduction

High-quality and reliable Large Language Model (LLM) benchmarks can effectively guide research, encourage innovation, monitor their advancement, and inform users of which model to use for their purpose. The importance of the final goal is underscored by the over 900k models currently available on Hugging Face, an online platform for open-source LLMs[1]. Various benchmarks are proposed for evaluating LLMs' ability in different aspects, including ARC [Chollet, 2019], HellaSwag [Zellers et al., 2019], Massive Multitask Language Understanding (MMLU) [Hendrycks et al., 2020], MT Bench [Zheng et al., 2023a], GSM8K [Cobbe et al., 2021], TruthfulQA [Lin et al., 2022], Natural Questions [Kwiatkowski et al., 2019], etc.

Most benchmarks are based on multiple-choice questions or other questions with objective gold standard answers, since it is easy to verify the LLMs' outputs. While they provide valuable evaluation for LLMs, open-ended tasks, e.g. providing judgment about a research paper, encompass a broader array of skills and require both objective and subjective reasoning. In addition, concern has been raised about data contamination (also called data leakage), where the training data contains information about the tasks in the benchmarks, as the LLMs are pre-trained on massive online data, which is also the source of some of the benchmark tasks. LLMs can show unreliably good performance due to data contamination [Sainz et al., 2023, Golchin and Surdeanu, 2023, Oren et al., 2023]. In contrast, with open-ended questions, LLMs can be asked to provide judgments about newly created content, e.g. the latest academic papers, for which LLMs have yet to index human evaluations or responses.

However, it is not clear how to automate the evaluation of subjective response quality. An added challenge is that there is no gold standard quality response with which to compare.

[1] https://huggingface.co/docs/hub/en/index

We would like an evaluator to have two properties. First, it should be **accurate** and be sensitive to the semantic content of the response. As current LLMs have already shown strong language ability, we want to focus on evaluating the semantic informativeness of candidate responses instead of their syntax or style. Second, the evaluator should be **manipulation-resistant**-we should not be able to manipulate a response in a trivial fashion to consistently increase evaluation scores. This is important because otherwise one cannot determine whether a high evaluation score indicates that the LLM output a high-quality response or merely results from manipulation designed to achieve an artificially inflated score.

Given these gaps in previous research, our research question is: **Can we develop accurate, manipulation-resistant, and automated evaluation metrics for textual responses with no gold standard reference to compare with?**

A straightforward approach may involve utilizing an alternative LLM as an oracle examiner to directly provide the evaluation, which has demonstrated efficacy in evaluating open-response QA [Bai et al., 2024], chatbot [Zheng et al., 2023b], etc. However, LLM examiners are susceptible to certain manipulations. In our experiments, elongating all responses by adding the same fixed sentences can significantly increase the GPT-4o LM examiner's score. Interestingly, previous research has shown that even human evaluations can be manipulated by adding meaningless text [Goldberg et al., 2023]. Meanwhile, other automated natural language generation (NLG) evaluation metrics, including BLEU [Papineni et al., 2002], ROUGE [Lin, 2004], BERTScore [Zhang et al., 2019], BARTScore [Yuan et al., 2021] and GPTScore [Fu et al., 2023], rely on comparison with a gold standard reference.

4.1.1 Our Contributions

We propose the *Generative Estimator for Mutual information (GEM)* which uses the estimated Shannon mutual information (MI) between a set of candidate responses and a set of peer reference responses, which need not be of gold standard quality. The mutual information measures the amount of information revealed about the reference responses by obtaining the candidate responses. By artificially removing stylistic and syntactical information, our approach measures how much semantic information the candidate responses can reveal about the reference responses, which is related to the concept of *semantic coverage* [Yuan et al., 2021, Nenkova and Passonneau, 2004]. We additionally propose a variant of our method, *Generative Estimator for Mutual Information with Synopsis (GEM-S)*, which estimates the mutual information conditional on a synopsis of the task, e.g. the abstract of the paper. This prevents a candidate response from receiving a high score based solely on superficial information. Consequently, the score emphasizes the additional semantic information gained

from the candidate responses.

For implementation, we utilize a generative language model to estimate the conditional distribution between two text responses. Thus, the GEM can be categorized as an LLM-probability-based metric, with techniques similar to BARTScore [Yuan et al., 2021] and GPTScore [Fu et al., 2023], from which our metric inherits the effectiveness of evaluating objective tasks with a gold standard.

Our work establishes a bridge between the information theoretical framework in the literature of information elicitation without verification [Kong and Schoenebeck, 2019, Lu et al., 2024] and the NLG evaluation problem. Though GEM and GEM-S resemble the GPPM and GSPPM mechanisms in Lu et al. [2024], with manipulation resistance aligned to their incentive compatibility, we make a necessary change to make the score more suitable for the NLG evaluation problem. Specifically, for incentive compatibility, the score can rely solely on accuracy of predicting the peer reference, while we use mutual information to capture the gain in that accuracy for evaluation purposes.

Results Overview The results of experiments validate GEM's accuracy and resistance to manipulation, and compare it with many different NLG evaluation metrics.

1. **Positive Correlation with Human Annotation** (Section 4.4.1) On a human-annotated dataset, the GEM metrics, especially GEM-S, achieve a significant positive correlation with human-labeled quality scores and demonstrate competitive performance with GPT4-o Examiner while outperforming all other baseline metrics.

2. **Better Sensitivity to Degradations** (Section 4.4.2) Compared to various baseline metrics, GEM and GEM-S are the only metrics that demonstrate significant sensitivity to all semantic degradations in our experiment, by effectively penalizing degraded responses.

3. **Better Robustness against Manipulations** (Section 4.4.3) GEM and GEM-S are the only metrics that exhibit no significant score increases after meaningless elongation (Figure 4.4) and GPT-4o/Llama-3.1 rephrase, whereas LMExaminer show vulnerabilities by significantly increasing scores after all manipulations.

Building on GEM and GEM-S, we introduce the **GRE-bench** (Generating Review Evaluation Benchmark) to evaluate LLMs' peer review capabilities with data from open-access research papers and peer reviews. With the continuous influx of new data each year, GRE-bench can mitigate data contamination issues. In addition, since GRE-bench is based on GEM or GEM-S, it inherits their nice properties demonstrated in the experiments above.

We run GRE-bench for various state-of-the-art LLMs on the ICLR 2023 dataset, and present the results in Section 4.5. We find a strong correlation between parameter size and GRE-bench score within LLM families, which further validates the effectiveness of GEM and GEM-S.

4.1.2 Related Work

LLMs in academic peer review. Given the success of LLMs, they have the potential to assist with peer reviews when used appropriately [Robertson, 2023, Kuznetsov et al., 2024]. Liang et al. [2023] use a survey to evaluate GPT-4-generated reviews, and find that the GPT-generated reviews are thought helpful by more than 50% of participants. However, there have been concerns regarding issues such as hallucinations [Donker, 2023] and inconsistent performance Liu and Shah [2023]. While our study uses the peer review scenario to evaluate LLM capabilities, it also provides a quantitative comparison for the effectiveness of various LLMs in generating informative reviews.

Information Elicitation. Both information elicitation and machine learning employ a principal-agent model [Ali and Silvey, 1966]. Specifically, the principal aims to elicit information from the agent, e.g. the probability of rain tomorrow. To incentivize the agent to provide truthful and informative reports, payment mechanisms are employed that reward truthful and informative reports more than untruthful or uninformative ones.

When the information is verifiable, e.g., we will eventually know whether it rains tomorrow, *proper scoring rules* [Good, 1952, Brier, 1950, Hendrickson and Buehler, 1971] can be applied, which are similar to the loss functions in supervised learning. When the information is subjective and unverifiable, Miller et al. [2005] propose the *peer prediction* mechanism, suggesting rewarding data by how well it predicts a peer's subjective report which is judged by a proper scoring rule. Prelec [2004] propose the Bayesian Truth Serum mechanism. Kong and Schoenebeck [2019] propose an information-theoretic framework, suggesting paying the agents according to the f-mutual information between their reports. The framework provides a unified theoretical view of previous mechanisms in Dasgupta and Ghosh [2013], and Prelec [2004]. Lu et al. [2024] first generalize the peer prediction to elicit informative textual responses with LLMs, providing the foundation for our GEM metrics.

Most information elicitation mechanisms theoretically ensure that truthful and informative reports yield higher expected scores. However, high ex-post correlation between scores and report qualities is critical for practical application, as highlighted by Burrell and Schoenebeck [2021], Xu et al. [2024]. In our study, to validate GEM's effectiveness in evaluating textual

reports, we measure both the ex-post correlation with human-annotated quality scores and expected score change after several degradation and manipulation strategies.

Estimation of Mutual Information. Kong and Schoenebeck [2018b] bridges information elicitation without verification and co-training, by formulating an optimization problem that maximizes the mutual information between two Bayesian predictors with two data sources. Cao et al. [2019], Xu et al. [2019] apply this information theoretical approach in co-training and de-noising respectively. Concurrently and independently, Belghazi et al. [2018] and Hjelm et al. [2018] propose neural network-based methods for estimating mutual information, which can be used as an optimization goal in unsupervised and self-supervised learning settings.

4.2 Preliminaries

In this section, we first introduce our model for benchmarking LLMs' judgments with no gold standard and then present Shannon mutual information (MI) and pointwise mutual information (PMI) as an unbiased estimator of MI.

Model We consider a scenario with a **candidate** under evaluation and a peer **reference** working on the same tasks. We denote $\mathcal{W}$ as the space of all tasks, with an underlying distribution over it, denoted as $\Delta\mathcal{W}$. Given a sample task $W \sim \Delta\mathcal{W}$, both the candidate and the reference generate a response, represented by the random variable X and Y respectively. The response space for both X and Y is denoted by Σ. The tuple (W, X, Y) follows an underlying distribution $\mathcal{D} := \Delta(\mathcal{W} \times \Sigma^2)$. *Note that the reference's responses do not need to be gold standard.* We assume that X and Y are independent conditioning on the task W.

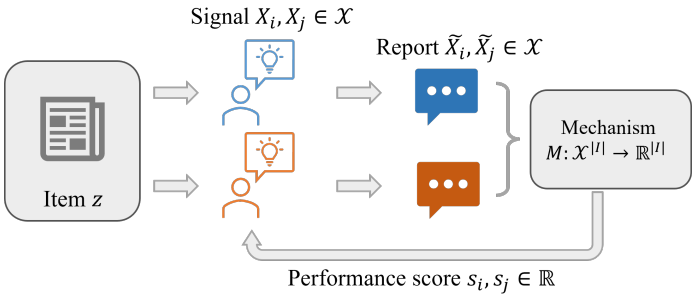


Figure 4.1: Our model

Empirically, in a dataset D, we have a list of sampled tasks $\mathbf{w} = \{w_1, \cdots, w_n\}$, and a list of corresponding reference responses $\mathbf{y} = \{y_1, \cdots, y_n\}$. Let the candidate generate a list of

responses $\mathbf{x} = \{x_1, \cdots, x_n\}$ to the corresponding tasks. We adopt a common assumption that all the samples (w_i, x_i, y_i) are generated i.i.d. following the underlying distribution $\mathcal{D}$.

Benchmarking the candidate The *evaluation metric* $f : \mathcal{W} \times \Sigma^2 \to \mathbb{R}$ maps a tuple (w, x, y) to a *score* S. Note that some evaluation metrics only utilize one or two of the items in the tuple. If the evaluation metric utilizes a language model to compute the score, we call the model used as the *evaluation-LM*. Different evaluation-LMs may lead to different evaluation results. A nice property of our GRE-bench is the consistency over evaluation-LMs, i.e., using a smaller evaluation-LM will not significantly change the evaluation results, we will discuss it in Section 4.5. Given a dataset D and an evaluation metric f, we can estimate $\mathbb{E}_{(W,X,Y)\sim\mathcal{D}}[f(W, X, Y)]$, by using the average of the evaluation metrics over all samples, which is named the *benchmark*.

Shannon Mutual Information Ideally, we aim to score the candidate according to the informativeness of her responses, thus, Shannon Mutual Information (MI) [Shannon, 1948] is a suitable metric. The MI between two random variables, X and Y, provides a quantitative measure of the information shared between them. Specifically, it assesses how much knowing one variable can inform us about the other. This is formally defined as: $\mathrm{I}(X;Y) = \sum_{x,y} \Pr[X = x, Y = y] \log \frac{\Pr[X=x,Y=y]}{\Pr[X=x]\Pr[Y=y]}$. The conditional mutual information $I(X; Y \mid Z)$ measures the mutual information between X and Y, conditioned on a random variable Z:

$$\mathrm{I}(X;Y|Z) = \sum_{z} \Pr[Z = z] \sum_{x,y} \Pr[X = x, Y = y|Z = z] \log \frac{\Pr[X = x, Y = y|Z = z]}{\Pr[X = x|Z = z]\Pr[Y = y|Z = z]}.$$

In an academic peer review scenario, Z can be some superficial information of a paper (e.g. the abstract), the conditional MI measures how much information is revealed beyond the abstract and emphasizes the MI gained from the semantics judgment.

Point-wise Mutual Information as an Estimator To estimate the MI with samples, we employ Point-wise Mutual Information (PMI) [Fano and Hawkins, 1961, Church and Hanks, 1990], an unbiased estimator of the MI. Formally, the PMI between realization x and y is defined as

$$\mathrm{PMI}(x;y) := \log \frac{\Pr[X = x, Y = y]}{\Pr[X = x]\Pr[Y = y]} = \log \Pr[Y = y \mid X = x] - \log \Pr[Y = y].$$

Similarly, the PMI between x and y conditional on z is defined as

$$\mathrm{PMI}(x; y \mid z) = \log \Pr[Y = y \mid X = x, Z = z] - \log \Pr[Y = y \mid Z = z].$$

4.3 Generative Estimator for Mutual Information (GEM)

In this section, we propose the Generative Estimator for Mutual Information (GEM), prove its theoretical effectiveness even if the reference's responses are not gold standard under certain assumptions, and discuss its empirical practice.

As discussed in Section 4.2, we use the mean of sample PMIs as an unbiased estimator of the MI. Given a sampled tuple with a task w_i, a candidate response x_i and a reference response y_i, we aim to compute $\mathrm{PMI}(x_i; y_i) = \log \Pr[Y = y \mid X = x] - \log \Pr[Y = y]$.

Following Yuan et al. [2021], Fu et al. [2023], Lu et al. [2024], we use a generative language model to estimate the probability distribution of the reference response conditional on the candidate's output, denoted by $\Pr_{\mathrm{LLM}}[Y = y \mid X = x]$, and the marginal distribution, denoted by $\Pr_{\mathrm{LLM}}[Y = y]$.

Specifically, we use the LLMs' inherent token prediction function. We tokenize y as $y = y^{(1)}y^{(2)}\cdots y^{(|y|)}$, and then integrate x into the prompt. Given any length-k prefix $y^{(1)}y^{(2)}\cdots y^{(k)}$, we use the LLM to predict $\Pr_{\mathrm{LLM}}[Y^{(k+1)} = y^{(k+1)} \mid Y^{(1)}\cdots Y^{(k)} = y^{(1)}\cdots y^{(k)}, X = x]$. By the chain rule of probability, multiplying over k, we have

$$\Pr_{\mathrm{LLM}}[Y = y \mid X = x] = \prod_{k=0}^{|y|} \Pr_{\mathrm{LLM}}[Y^{(k+1)} = y^{(k+1)} \mid Y^{(1)}\cdots Y^{(k)} = y^{(1)}\cdots y^{(k)}, X = x].$$

Similarly, we can estimate the marginal distribution $\log \Pr_{\mathrm{LLM}}[Y = y]$. Thus, we have the estimated PMI, denoted as $\widehat{\mathrm{PMI}}(x_i; y_i) = \log \Pr_{\mathrm{LLM}}[Y = y \mid X = x] - \log \Pr_{\mathrm{LLM}}[Y = y]$, and consequently, the estimated MI over dataset D is $\widehat{I}(X; Y) = \frac{1}{n}\sum_{i=1}^{n} \widehat{\mathrm{PMI}}(x_i; y_i)$.

4.3.1 Theoretical Guarantees: Benchmarking LLMs with No Gold Standard

We now show that even when the reference is not gold standard, the estimated MI can still be a benchmark where better candidates achieve higher scores theoretically. We adopt a widely-used model in decision-making theory [Blackwell, 1951, 1953]. The candidate's response follows an *information structure*, a mapping $\sigma : \mathcal{W} \to \Delta\Sigma$, where Σ is the response space. It represents the distribution of responses conditional on the task.

Within this model, Blackwell's order [Blackwell, 1951, 1953] provides a partial order of

the informativeness of information structures. Consider two candidates, H and L, whose responses X_H and X_L follow information structures σ_H and σ_L respectively. Information structure σ_H is more informative than σ_L in the sense of Blackwell order if there exists a stochastic mapping Γ, such that $\sigma_L = \Gamma\sigma_H$. Intuitively, the information structure with a lower Blackwell order is more noisy, and thus, can be regarded as lower quality.

We now show that higher Blackwell order information leads to (approximately) higher GEM scores when the LLM can provide a "good" estimation. Formally, we have

Proposition 4.1. *When the KL-divergence*[2] *between the LLM estimated distribution and the underlying distribution satisfies*

$$D_{KL}\left[\Pr_{\text{LLM}}[Y = \cdot \mid X = x_H] \middle\| \Pr[Y = \cdot \mid X_H = x_H]\right] < \epsilon,\ \forall x_H.$$

For the two candidates H and L discussed above, the information structure of H Blackwell dominates L's, when the size of dataset n goes to infinity, we have $\widehat{I}(X_H; Y) \geq \widehat{I}(X_L; Y) - \epsilon$.

The proof is provided in Appendix C.1, following previous information elicitation literature [Lu et al., 2024]. It mainly relies on the data processing inequality [Shannon, 1948, Cover and Thomas, 1999]. Additionally, we provide a similar proposition for conditional MI in Appendix C.1.

We highlight that while this theoretical result requires a strong assumption, it gives intuitions of why and how our GEM scores could work. In the next several sections, we present our experiment validating the effectiveness of GEM with real data.

4.3.2 Filtering Out Shortcuts

Textual data is inherently high-dimensional, encompassing various aspects. For example, peer reviews may have a hierarchical information structure, including language style, the topic of the paper, and critical judgments, as shown in Figure 4.2. While each of these dimensions influences LLM predictions and consequently contributes to the mutual information, they are not equally important when assessing the quality of LLM-generated responses, particularly in the context of peer reviews.

Empirically, the correlation between superficial information can act as "shortcuts" [Geirhos et al., 2020] confounding LLM predictions, cause unintended bias in the estimated mutual information, and consequently distort the evaluation metric towards superficial correlations rather than thorough judgments. Since we focus on the semantic quality of these responses,

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

we aim to filter out dimensions like language styles that could skew the evaluation metric by offering "shortcuts" in LLM predictions.

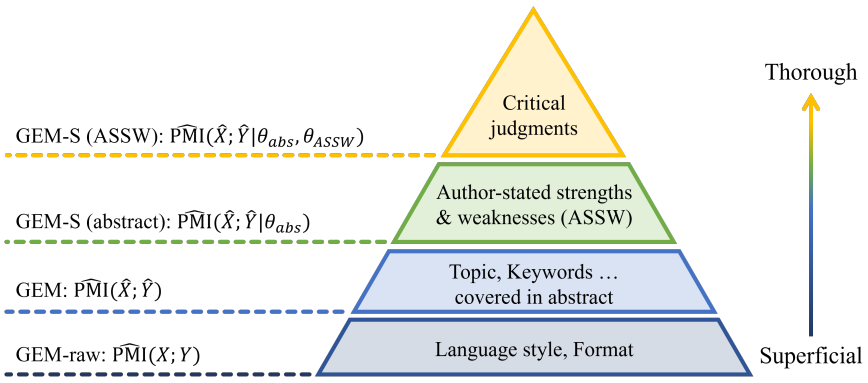


Figure 4.2: Example of Hierarchical Information Structure in Peer Reviews

Preprocessing A straightforward yet effective method proposed by Lu et al. [2024], is preprocessing responses using a specific LLM to rephrase and summarize the content. This preprocessing step standardizes language style and eliminates superficial information, thereby reducing the influence of irrelevant dimensions. We denote the preprocessed versions of responses x_i and y_i as $\hat{x}_i$ and $\hat{y}_i$ respectively.

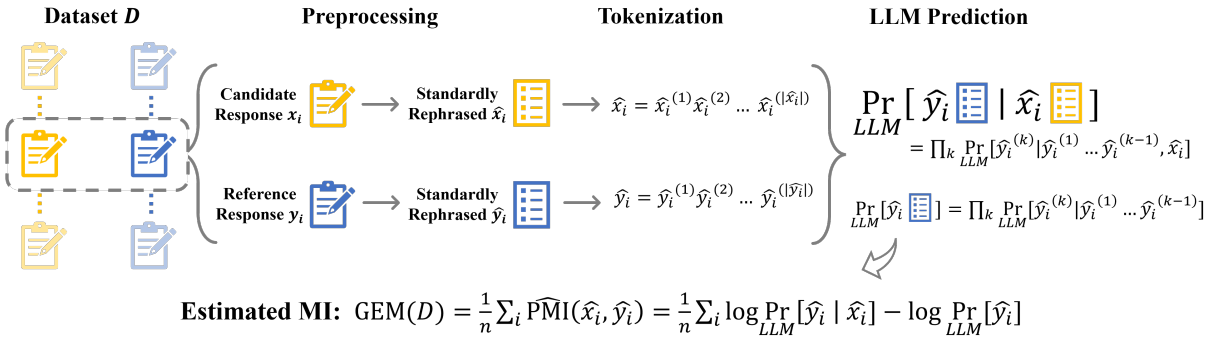


Figure 4.3: An Overview of Our Generative Estimator for Mutual Information

We have now outlined all steps of our Generative Estimator for Mutual Information (GEM) framework, as illustrated in Figure 4.3. The **GEM** metric between a candidate response x_i and a reference response y_i is defined as follows:

$$\text{GEM}(x_i, y_i) := \widehat{\text{PMI}}(\hat{x}_i, \hat{y}_i).$$

By taking the arithmetic average of all $\text{GEM}(x_i, y_i)$ over the dataset D, we have an estimated mutual information between random variables X and Y, as shown in Figure 4.3.

For comparison, we also introduce the **GEM-raw** metric calculated without preprocessing as follows: GEM-raw$(x_i, y_i) := \widehat{\text{PMI}}(x_i, y_i)$, which serves as a baseline.

GEM-S: conditioning out the synopsis Kong and Schoenebeck [2018a] suggest surface-level information can inflate mutual information through simple correlations, reducing the mutual information's effectiveness in reflecting semantic informativeness, especially when the responses have a hierarchical information structure as shown in Figure 4.2. As some superficial information can be contained in a synopsis of the task, e.g. the abstract of the paper to be reviewed, conditional mutual information can be used to distill the semantic informativeness [Lu et al., 2024].

The GEM-S metric is designed to mitigate this issue by conditioning out the synopsis of the task, denoted as $\theta(w_i)$. The **GEM-S** metric is formally defined as:

$$\text{GEM-S}(\theta(w_i), x_i, y_i) = \widehat{\text{PMI}}(\hat{x}_i, \hat{y}_i \mid \theta(w_i)).$$

With the conditional formulation above, only the additional, non-trivial mutual information is measured, thereby offering a more accurate assessment of the candidate's performance in providing information beyond superficial content.

Moreover, we can adjust the granularity of the synopsis to make the GEM-S metric focus on evaluating different aspects of the candidate LLMs' ability. In the example of Figure 4.2, the standard GEM measures the general ability to produce relevant reviews. By conditioning on the abstract, GEM-S (abstract) evaluates the LLM's ability to generate judgments, including retrieving author-stated strengths and weaknesses (ASSW) and providing critical feedback. By further conditioning on a summary of these author-stated strengths and weaknesses, GEM-S (ASSW) narrows the focus to the LLM's capacity for critical thinking and delivering constructive feedback.

4.4 Experiment: Validating GEM's Effectiveness

In this section, we provide our experiment setup to validate the effectiveness of the GEM metric in benchmarking LLMs without gold-standard references, and show the results.

GEM setup We present the empirical setup of the GEM metric and its variants. In the validation experiment in this section, Llama-3.1 8B is used as the evaluation-LM to estimate $\log \Pr_{\text{LLM}}[Y = y \mid X = x]$ and $\log \Pr_{\text{LLM}}[Y = y]$. After confirming the effectiveness of GEM metrics using the small model, for even better performance, we scaled up to a larger,

70B-parameter version of Llama-3.1 for computing the GRE-bench on the ICLR 2023 dataset in Section 4.5, where we also show the correlation between results of the smaller (8B) and larger (70B) models, to ensure consistency of our approach.

For text preprocessing, we employ GPT-4o. We also present the results with Llama-3.2 90B preprocessing in Appendix C.4.3, and demonstrate the robustness of GEM metrics with various preprocessing models. For GEM-S, without further indication, we use the abstract of the paper as the synopsis in this section. The prompts for probability estimation and preprocessing can be found in Appendix C.2.

Baselines We compare the performance of GEM and its variants with the following established evaluation metrics commonly used in NLG evaluation, including two metrics from the pre-LLM era, *BLEU* [Papineni et al., 2002] and *ROUGE-L* [Lin, 2004], and an embedding-based metric, *BERTScore* [Zhang et al., 2019], a probability-based metric *BARTScore* [Yuan et al., 2021], and a GPT-based metric *LMExaminer (Language Model as Examiner)* [Bai et al., 2024]. Here, we mainly discuss the LLM-based metrics, the detailed implementation of all baselines is provided in Appendix C.3.1.

BERTScore [Zhang et al., 2019] evaluates generated text by measuring the cosine similarity between token embeddings from a pre-trained language model. Instead of BERT, we use a recent model stella_en_400M_v5 [Zhang, 2024] as the evaluation-LM for better token embedding performance.

BARTScore [Yuan et al., 2021] is a probability-based metric. It has three variants, precision, recall, and F1-score. Instead of BART, we use the state-of-the-art Llama3.1-8b as the evaluation-LM for estimating the probability, which is the same as our GEM metrics.

LMExaminer [Bai et al., 2024] is one of the recently popular methods where a language model (LM) is used to evaluate the quality of generated text based on specific evaluation criteria, serving as an examiner. With empirical evidence suggesting that GPT-4 outperforms open-source models and even finetuned models of pointwise grading on AlignBench [Ke et al., 2023], we use GPT-4o as a baseline in our experiment. For peer review tasks, our prompt adopts criteria based on Review Quality Indicators (RQIs) [Goldberg et al., 2023, Van Rooyen et al., 1999, Superchi et al., 2019] including four aspects, understanding, coverage, substantiation, constructiveness.

4.4.1 Result 1: Positive Correlation with Human Annotation

In this section, we show the experiment results utilizing a human-annotated dataset to validate how well the GEM metrics correlate with human-labeled quality scores compared to

various baselines. A higher correlation indicates that the evaluation metric better aligns with human preference.

Note that, a perfect correlation can be impossible to achieve for several reasons: Human annotations are inherently noisy [Goldberg et al., 2023], evaluation of individual responses can also be noisy due to subjectivity, and there is no gold-standard reference. Nonetheless, our goal is to demonstrate systematic-level positive correlation.

Score. For task w_i with corresponding responses $\mathbf{x}_i = \{x_{ij}\}$, where x_{ij} is the j-th response, the score of response x_{ij} is computed by $\frac{1}{|\mathbf{x}_i|-1}\sum_{k\neq j} f(w_i, x_{ij}, x_{ik})$ where f is the evaluation metric.

Human-Annotated Peer Grading Dataset. This dataset contains 30 student project proposals in a graduate-level class on machine learning and about 180 peer evaluations of the proposals. We use GPT-4o to generate a short abstract for each proposal. Each proposal has at least 4 reviews, and each review contains an overall score and textual feedback, including "Strengths of the project", "Weaknesses of the project/likely roadblocks", and "Ideas for improvement or specific directions". According to the quality of students' textual feedback, an instructor manually grades the reviews with grade A, B, or C.

Evaluation Metric	ρ	p-value
BLEU	0.023	0.772
ROUGE-L	-0.244	0.002
BERTScore	-0.061	0.439
BARTScore-F1	-0.237	0.002
BARTScore-prec.	-0.511	2.3e-12
BARTScore-recall	**0.164**	0.036
LMExaminer	**0.537**	1.1e-13
GEM-raw	**0.300**	9.2e-05
GEM	**0.431**	7.5e-09
GEM-S	**0.479**	7.4e-11

Table 4.1: Spearman's correlation coefficient (ρ) and p-values between various evaluation metrics and instructor-annotated grades. Significant positive correlations ($p < 0.05$) are bolded.

Statistics Metric and Results. Table 4.1 shows Spearman's correlation coefficient ρ between the scores of various evaluation metrics and the instructors' grades. All variants of our GEM metrics show a significant positive correlation. The GEM-S has the highest correlation over all three GEM variants, implying the effectiveness of filtering out shortcuts. It also shows competitive performance compared to GPT-4o LMExaminer which has the highest correlation. All other baselines perform worse than GEM, and only BARTScore-Recall has a significant positive correlation. In later sections, we provide further comparisons between GEM and LMExaminer.

4.4.2 Result 2: Better Sensitivity to Degradation

We now compare how sensitive the evaluation metrics are to degradations that obviously reduce the semantic informativeness of the responses. A statistically significant score decrease indicates the sensitivity of the metric. We first introduce our validation workflow in Algorithm 4, the ICLR peer review dataset, and then introduce several degradation strategies. The workflow and dataset will also be used in the robustness check against manipulation strategies in Section 4.4.3.

ALGORITHM 4: Validation Workflow

Input: A dataset D with n tuples of tasks and associated text responses. An evaluation metric f computing the scores. A degradation/manipulation strategy M.

Output: Statistics metrics.

for $i = 1$ **to** n **do**

  Get the i-th tuple from the dataset: task w_i, candidate response x_i, and reference response y_i ;

  Compute $s_i := f(w_i, x_i, y_i)$;

  Replace the response x_i with x'_i according to degradation/manipulation strategy M;

  Compute $s'_i := f(w_i, x'_i, y_i)$;

end

Compute the means μ,μ' and standard deviations σ,σ' of $\{s_i\}_{i\in[n]}$ and $\{s'_i\}_{i\in[n]}$ respectively.

Note that the evaluation metrics that we compare are in different scales, thus, instead of comparing the mean difference, we standardize the scores with pooled standard deviation $\sigma_{\text{pooled}} := \sqrt{(\sigma^2 + \sigma'^2)/2}$, and compare the standardized mean difference (SMD) $d := \frac{\mu' - \mu}{\sigma_{\text{pooled}}}$ [Andrade, 2020]. This form of SMD is also known as Cohen's d [Cohen, 1988].

ICLR Dataset. We use the ICLR 2023[3] peer review data publicly available on OpenReview. We randomly select 300 papers as our dataset, and for each paper, we randomly select 3 original human reviews: one review to serve as a human candidate, and the other two as reference responses.

Degradation Strategies. We introduce three degradation strategies as follows. Note that the degraded responses after *Sentence Deletion* or *Deletion & Completion* can be regarded as Blackwell-dominated by the original one. The detailed setup is provided in Appendix C.3.3.

1. *Sentence Deletion.* We delete every other sentence of x_i to get a degraded response x'_i.

[3]At the time of our experiment, ICLR 2024 review data was not yet available through the OpenReview API.

2. *Deletion & Completion.* After deletion, we use GPT-4o to complete the deleted sentence with consistent language style and semantics to the original response, which provides an informatively degraded response x_i' but with a similar length to the original one.

3. *Abstract-only Review.* We use Claude-3-sonnet[4] to create a fictitious review x_i' with only the abstract of the paper, and use x_i' to replace x_i.

Statistics and Results. We compare the standardized mean differences (SMD) of evaluation metrics in Table 4.2: GEM and GEM-S are only two metrics that consistently have significant negative SMD on all three degradations. The LMExaminer shows the highest effectiveness on Sentence Deletion and Deletion & Completion, but failed to penalize reviews only based on the abstract.

Table 4.2: Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after degradations and manipulations of various evaluation metrics. Significant score decreases ($p < 0.05$) after degradations are highlighted in bold green, implying the evaluation metric can effectively penalize the degradation. Significant score increases ($p < 0.05$) after manipulations are highlighted in bold red, implying the evaluation metric is not robust against the manipulation. We provide 95% confidence interval of SMD values in Appendix C.4.1.

Evaluation Metric	Sentence Deletion	Deletion & Completion	Abstract-only Review	GPT-4o Rephrase	Llama3.1 Rephrase	Meaningless Elongation
BLEU	**-0.282**	0.016	**-0.692**	-0.975	-0.920	-0.165
ROUGE-L	**-0.073**	0.091	0.022	**0.028**	**0.120**	-0.196
BERTScore	**-0.100**	**-0.188**	0.840	**0.134**	**0.063**	**0.064**
BARTS.-F1	0.401	0.201	0.394	**0.130**	**0.332**	-0.304
BARTS.-prec.	0.627	0.317	0.617	**0.218**	**0.541**	-0.439
BARTS.-recall	**-0.017**	**-0.016**	0.004	-0.027	-0.028	**0.004**
LMExaminer	**-1.290**	**-0.417**	0.715	**0.187**	**0.104**	**0.105**
GEM-raw	**-0.123**	**-0.126**	0.020	-0.123	-0.126	**0.020**
GEM	**-0.401**	**-0.308**	**-0.191**	-0.058	-0.107	-0.063
GEM-S	**-0.409**	**-0.206**	**-0.566**	-0.046	-0.114	-0.070

4.4.3 Result 3: Better Robustness against Manipulation

Evidence has been found that the LLM evaluators, and even humans, are not robust against specific manipulation strategies, they may rely on language style or length as shortcuts instead of providing an evaluation based on semantic quality [Panickssery et al., 2024, Bai et al., 2024, Goldberg et al., 2023], thus, it is important to further check our metric's robustness against several manipulations.

[4]We intentionally use a different LLM from Deletion & Completion to make sure the effect of score decrease is not caused only by the LM.

Specifically, we employ two manipulation strategies, which do not significantly change the semantics, but may change the language style or add meaningless starting sentences. After manipulation, if the score significantly increases, the evaluation metric fails to pass the robustness check. We use the same workflow and dataset introduced in Section 4.4.2.

Manipulation Strategies. We now introduce the manipulation strategies. The detailed setup, including the prompts and examples, is provided in Appendix C.3.4.

1. *GPT-4o/Llama-3.1 Rephrase.* Following Bai et al. [2024], we use GPT-4o and Llama-3.1 to rephrase the response x_i while trying to keep the semantics unchanged. We test whether the evaluation metrics have preference towards GPT-4o or Llama-3.1 generated language.

2. *Meaningless Elongation.* We adopt a meaningless elongation method similar to Goldberg et al. [2023]'s human subject experiment where a set of fixed meaningless sentences is added to each section of all the reviews as shown in Figure 4.4.

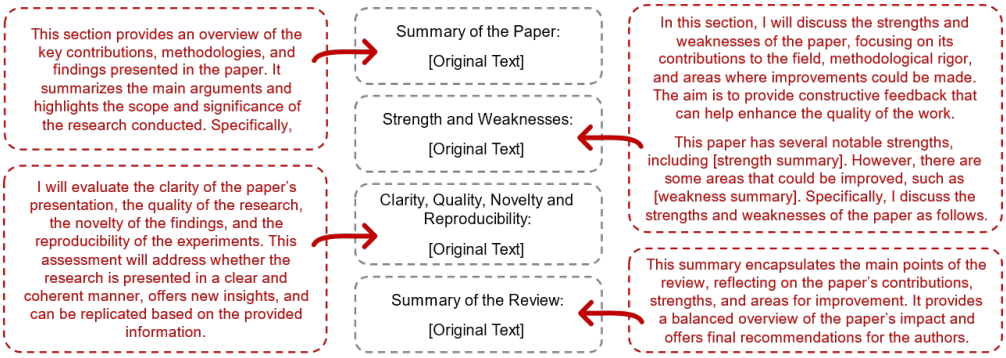


Figure 4.4: Meaningless Elongation

Statistics and Results. The SMDs shown in Table 4.2 demonstrate that LMexaminer has significant score increases after all three manipulations, indicating its lack of manipulation-resistance. In contrast, GEM and GEM-S exhibit no significant score increase across these manipulations. Furthermore, comparing SMDs after degradations and manipulations in Table 4.2, we observe that the scores of GEM and GEM-S decrease less after manipulations compared to degradations, indicating that manipulations result in less information loss than degradations.

4.5 GRE-bench: Benchmarking LLMs with GEM

After validating our evaluation metrics GEM and GEM-S in Section 4.4.1, 4.4.2, and 4.4.3, we now show the results of GRE-bench (Generating Review Evaluation Benchmark) based

on GEM and GEM-S. We use the same ICLR 2023 peer review dataset introduced in Section 4.4.2.

We prompt LLMs to generate detailed reviews for each paper in the dataset, with the prompt inspired by Liang et al. [2023], which covers various aspects, including paper summary, strengths, weaknesses, questions, etc. Details of the prompts are provided in Appendix C.2.3.

We then use GEM and GEM-S to score the LLM-generated reviews. Specifically, as discussed in Section 4.3.2, we use GEM-S (abstract) and GEM-S (ASSW), where the synopsis consists of an abstract for GEM-S (abstract), and an abstract supplemented with a summary of author-stated strengths and weaknesses (ASSW) for GEM-S (ASSW). We prompt GPT-4o to get the ASSW summary of each paper, the prompt is shown in Appendix C.2.4. We visualize the results of the GRE-bench based on GEM, GEM-S (abstract), and GEM-S (ASSW) respectively in Figure 4.5. More detailed numerical results are provided in Table C.4 of Appendix C.4.2.

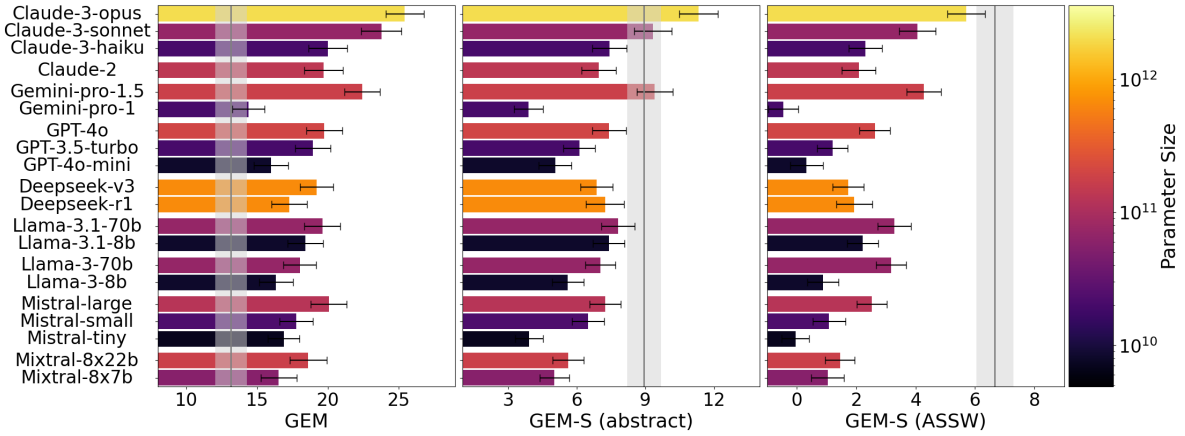


Figure 4.5: Results of GRE-bench based on three evaluation metrics with 90% confidence intervals vs. model parameter sizes[5]indicated by color. The grey line represents the average human baseline, with the 90% confidence interval shaded in grey.

Figure 4.5 shows that, within each LLM family, such as the Llama-3 or Claude series, models with larger parameter sizes tend to perform better. Moreover, newer versions tend to outperform their predecessors with similar model sizes, as seen in comparisons between Llama-3.1 and Llama-3. These observations further validate the effectiveness of our GEM and GEM-S metrics in accurately evaluating LLMs.

Notably, several models with large parameter sizes surpass the human baseline (gray line) when using GEM and GEM-S (abstract) metrics. We hypothesize this is because strong LLMs like Claude-3 opus can effectively retrieve contributions and limitations stated by the authors, and will include that in their reviews. In contrast, human reviewers tend to omit

judgments that are less important or already stated by the authors due to writing costs, resulting in a loss of mutual information. This hypothesis is supported by the observation that, when we condition out the author-stated strengths and weaknesses, the human reviewer has the highest GRE-bench score based on GEM-S (ASSW).

This observation further validates the hierarchical information structure discussed in Section 4.3.2. By conditioning out different levels of information, the benchmark captures subtly different abilities. GEM-S (ASSW) emphasizes the quality of critical judgment, while GEM and GEM-S (abstract) focus more on evaluating LLMs' foundational abilities, including retrieving useful information from the paper, which is especially valuable for differentiating LLMs with smaller parameter size.

GRE-bench vs. Other benchmarks The ability to generate informative reviews relies on several key factors, including logical reasoning, critical thinking, and fact-checking. Various widely used benchmarks measure different aspects of LLM performance. To better understand which abilities are most critical for generating high-quality reviews, in the sense of GRE-bench scores, we compared our results with several other benchmarks.

We present the Spearman's correlation coefficients between the GRE-bench scores of all the models and their corresponding scores on other benchmarks[6] in Table 4.3 in Appendix C.4.2, providing insights into which factors most strongly influence review generation quality. Notably, GRE-bench highly correlates with HellaSWAG and ARC-C which measures reasoning ability, and less correlates with benchmarks for coding ability (HumanEval) or math ability (MATH, GSM8K).

Table 4.3: Spearman's correlation coefficients between GRE-bench and other popular benchmarks.

Base Metric	MMLU	ARC-C	HellaSwag	GSM8K	MATH	HumanEval	GPQA
GEM	0.55	0.68	0.74	0.58	0.37	0.36	0.60
GEM-S (abstract)	0.66	0.70	0.82	0.73	0.43	0.43	0.67
GEM-S (ASSW)	0.68	0.78	0.84	0.73	0.43	0.48	0.71

Consistent results over various Evaluation LMs. The Spearman's correlation coefficients between the GRE-bench scores based on GEM metric computed using Llama-3.1 8B

[5]In Figure 4.5, the parameter sizes of closed-source models are estimated from unofficial sources. This information is only to provide context, and none of our conclusions or key findings depend on these estimates.

[6]We collect benchmark data from online sources, such as technical reports of language models. Since some models have missing results for specific benchmarks, we exclude those entries when computing correlation. For more details, please refer to our code repository.

and that same score computed with Llama-3.1 70B is 0.93. This indicates that the score is largely robust with respect to the specific evaluation LM used. The analogous coefficient for the GRE-bench scores based on GEM-S (abstract) and GEM-S (ASSW) are 0.92 and 0.90 respectively. We further provide the comprehensive results of GRE-bench with Llama-3.1 8B in Table C.5 in Appendix C.4.2. This consistency shows that with our metrics, a small model with 8B parameters that can be deployed on a single NVIDIA L4 GPU, plus proper preprocessing, is still effective in evaluating the quality of judgments generated by much larger models, though larger evaluation LM may have better evaluation performance.

4.6 Conclusion and Discussion

In conclusion, we introduce GEM and GEM-S, two metrics for evaluating LLMs on tasks without gold-standard references. They have proved to be accurate and manipulation-resistant, outperforming other metrics in the experiment. Based on GEM and GEM-S, we present GRE-bench for assessing LLMs' peer review capabilities while minimizing the risk of data contamination.

Our approach has certain limitations. First, although our experiments demonstrate its effectiveness, the theoretical results depend on the accurate estimation of the underlying distribution by the LLM, which may not always be reliable. With ongoing advancements in LLM technology, we hope that these estimations will become more accurate over time. Second, while our method does not require gold-standard references, it requires reference that includes original human-generated information. If all reviewers submit solely LLM-generated reviews—such as those only from GPT-4o—the mutual information-based approach may unfairly assign the highest score to GPT-4o. This echoes recent research demonstrating that the absence of original human-generated data in AI evaluation can result in model collapse [Shumailov et al., 2024].

More generally, when reference reviews contain LLM-generated or LLM-assisted text, LLM candidate scores may be inflated due to correlation between the candidate review and the LLM-generated component of the reference reviews. For example, candidate LLMs may share similar prior knowledge, phrasing patterns, or review opinions with the LLMs used by human reviewers. GEM-S can mitigate this issue by conditioning on task-level information, such as the abstract or author-stated strengths and weaknesses. As discussed in the Section 4.5, this helps distinguish superficial information from critical judgment. If the LLM-generated portion of a reference mainly restates superficial or author-provided information, GEM-S can reduce the resulting inflation. However, GRE-bench relies on references preserving sufficient (conditionally) independent judgment. Its effectiveness therefore depends on the choice of

synopsis, which remains an open question for future work.

For future work, we can use new open-access peer review data to update GRE-bench every year and observe how the evaluation evolves over time, to monitor any improvements in LLMs' peer review capabilities. This will also ensure the benchmark stays current and minimizes data contamination. Additionally, exploring GEM's potential to evaluate subjective content in other scenarios, such as book reviews and movie reviews, could broaden its applicability and further demonstrate its versatility. We provide initial results in validating GEM on a Yelp restaurant review dataset in Appendix C.4.4.

CHAPTER 5

Beyond Correlation: Systematically Designing and Testing Aligned Text Evaluation Metrics with f-Mutual Information

5.1 Introduction

Reference-based evaluation metrics are essential for scalable assessment of natural language generation systems, especially in the era of large language models. Given a candidate response and one or more reference responses for the same task, such metrics assign a score intended to reflect the quality, informativeness, or correctness of the candidate. Human evaluation remains the gold standard, but it is too slow and expensive at large scale. In practice, reference-based metrics are used not only to compare systems after training, but increasingly to guide development and optimization itself. As their role expands from measurement to optimization, we need to understand not only whether their scores correlate with human judgments, but also what incentives these scores create.

Most standard reference-based metrics are designed to approximate human judgments. Following the dissertation-wide terminology introduced earlier, this corresponds to **statistical alignment**: a useful metric should track human assessments of text quality. However, once a metric is used as an optimization objective, correlation with human judgments is not enough. The metric also shapes the behavior of systems and people optimizing against it.

The complementary requirement is **strategic alignment**: a metric should be robust to score-seeking perturbations of the text. Recent work proposes evaluation metrics with improved robustness to such strategic behavior [Lu et al., 2024, Xu et al., 2025, Robertson and Koyejo, 2025, Wu and Hartline, 2024]. These works show that evaluation metrics must be judged not only by their statistical correlation with human ratings, but also by whether they remain reliable under strategic perturbations.

Contribution 1: Test Principles Existing metrics are introduced for different goals and tested on different datasets with different criteria, making head-to-head comparison difficult.

As a result, it remains unclear how evaluation metrics should be systematically validated when the goal is not only statistical alignment with human ratings, but also strategic alignment under score-seeking behavior.

In this chapter, we address this validation gap by introducing unified test principles for reference-based evaluation metrics. Generalized from previous studies [Wu and Hartline, 2024, Xu et al., 2025, Lu et al., 2024, Robertson and Koyejo, 2025], we propose three complementary principles for comparing metrics. The first principle is correlation with human preference: a metric should agree with human judgments under ordinary, non-adversarial conditions. This principle corresponds to statistical alignment and captures the standard desideratum in NLG evaluation.

The second and third principles evaluate strategic alignment from two complementary perspectives. The second principle is degradation sensitivity: when a text is deliberately degraded by removing or corrupting task-relevant information, the metric score should strictly decrease. This tests whether the metric discourages low-effort reports, which save effort by providing strictly less information while attempting to receive similar credit.

The third principle is manipulation robustness: when a text is strategically modified to inflate the metric score without adding genuine content, the metric should not increase. This tests whether the metric discourages untruthful reports, which exploit superficial features of the evaluation procedure. Unlike degradation, manipulation need not strictly reduce the information content of the text. It may only weakly decrease it or preserve much of the original content while changing style, format, or wording to obtain a higher score. Together, these three principles provide a common standard for evaluating both prior and new metrics on equal footing.

Contribution 2: Metric Design Framework Having introduced a unified test framework, we now turn to the complementary construction question: how should one design metrics that pass these tests? The two principles under strategic alignment point to *mutual information* (MI) as a natural basis for manipulation-resistant evaluation metric design. Both degradation and manipulation reduce the informativeness of the text: degradation strictly reduces information, whereas manipulation weakly reduces information to exploit the metric. Thus, a well-behaved metric should obey the information-monotonicity principle: transformations that weakly decrease information should not increase the expected score. MI-based metrics are designed precisely around this principle. They score a candidate by estimating the mutual information it shares with an independent reference for the same task. The strategic alignment guarantee thus follows from the information monotonicity of the mutual information. Recent methods instantiate this idea in different forms: GEM [Xu et al., 2025] and GPPM [Lu et al.,

2024] use KL-based pointwise mutual information estimated from token-level autoregressive probabilities or LLM support judgments, while TVD-MI [Robertson and Koyejo, 2025] uses an LLM critic to estimate a total-variation MI objective through same-task versus cross-task classification. These works demonstrate the promise of information-theoretic scoring for semantic informativeness and manipulation resistance.

However, there is no unified framework for constructing strategically aligned metrics. Previous papers are developed as separate methods rather than as instances of a common design principle. They differ in the choice of the MI measure, estimator, representation granularity, and prediction model, and these choices are usually evaluated in isolation.

We propose a unified design framework for constructing MI-based evaluation metrics. Our framework decomposes every MI-based metric into four elements: an information measure (MI), an MI estimation method, a representation of a text, and a prediction mechanism. This decomposition recovers existing methods such as GEM, GPPM, and TVD-MI as special cases, while also exposing a broader design space of new metrics. The framework therefore gives a principled recipe for constructing new metrics and for analyzing which design components drive desirable behavior.

Our empirical results support a central takeaway: statistical alignment alone is not enough to certify an evaluation metric as a reliable scoring rule. Across peer review, summarization, and question answering, we find that LLM-as-a-Judge achieves the strongest correlations with human ratings, yet are vulnerable to manipulations. In contrast, MI-based metrics provide stronger strategic alignment. In particular, the statement-level TV f-variational metric, a new design exposed by our framework, is the most reliable overall.

Together, our two contributions provide a unified perspective on aligned text evaluation beyond simple human correlation. The test principles explain how evaluation metrics should be validated, while the design framework explains how MI-based metrics can be constructed and varied. Combining the two allows us to compare existing methods fairly, explore new points in the design space, and identify which combinations of information measure, estimator, and distribution model lead to the most reliable evaluation behavior across domains.

5.2 Problem Setup

We evaluate a scoring metric S. Let W denote a task. We consider a candidate report X and a reference report Y, both belonging to the text space $\mathcal{V}^{\leq L}$ over a vocabulary $\mathcal{V}$ with maximum length L. The candidate X is the random variable of the response under evaluation, while Y is the random variable of a reference independently produced for the same task. We write (x, y) as the realized value of a candidate-reference pair.

A standard *reference-based metric* is a function $S(x, y)$ that maps a candidate-reference pair to a scalar score. The most general form of the metric is given additional optional *negative reference* for contrastive scoring, $y^- \in \mathcal{V}^{\leq L} \cup \{\emptyset\}$, drawn from the marginal distribution P_Y. For example, a reference sampled from another task.

Given these inputs, we write the general form of a metric as a contrastive scoring function

$$S(x, y, y^-)$$

Intuitively, given optional input y^-, this contrastive formulation allows the metric to score agreement with the matched reference y while penalizing patterns that are common under the marginal distribution P_Y.

5.3 Test Principles

We compare evaluation metrics along statistical alignment and strategic alignment, under our test principles, generalized from previous work [Wu and Hartline, 2024, Lu et al., 2024, Xu et al., 2025, Robertson and Koyejo, 2025]. Specifically, for statistical alignment, we test the statistical correlation with human quality ratings; for strategic alignment, we use degradation strategies to simulate low-effort reporting and manipulation strategies to simulate untruthful reporting. We test whether the metrics can penalize degradations and resist manipulations.

Statistic alignment test: Correlation with human ratings. For datasets providing absolute human scores, let h_i denote the human rating on candidate response x_i, and s_i denote the corresponding score assigned by an evaluation metric. We measure statistical alignment using Spearman's rank correlation,

$$\rho = \text{Spearman}\Big(\{s_i\}_{i=1}^n, \{h_i\}_{n=1}^n\Big).$$

When multiple annotators are available, we average their scores at the item level before computing ρ.

Strategic alignment test 1: Sensitivity to degradation A degradation reduces the semantic informativeness of a candidate by deleting evidence, restricting context, or forcing low-effort responses; a well-behaved metric should assign strictly lower scores to degraded variants. Degradation tests are perturbations that reduce the task-relevant information, coverage, or specificity of a candidate.

For each candidate report x_i, we apply a degradation strategy M to produce a perturbed report x_i^M, and compute scores s_i and s_i^M for the original and perturbed report respectively. Table D.1 lists 6 strategies used in our experiments in Section 5.5.

For each strategy M, we compute the paired score change

$$\Delta_i^{(M)} = s_i^M - s_i, \qquad \bar{\Delta}^{(M)} = \frac{1}{n}\sum_{i=1}^{n} \Delta_i^{(M)}.$$

To compare metrics' effect size on different numerical scales, we report Cohen's d (standardized mean difference):

$$d_M = \frac{\mu_M - \mu}{\sqrt{(\sigma_M^2 + \sigma^2)/2}},$$

where (μ, σ) and (μ_M, σ_M) are the empirical mean and standard deviation of the clean and perturbed scores respectively. This scale-normalized reporting is adapted from Xu et al. [2025]. For degradations, desirable behavior is $\bar{\Delta}^{(M)} < 0$ and $d_M < 0$.

Strategic alignment test 2: Manipulation Resistance A manipulation is designed to inflate the score by exploiting metric shortcuts, such as verbosity, stylistic regularity, or prior-hugging generic content, without introducing new task-specific evidence; a robust metric should be insensitive to such perturbations. Manipulation tests are perturbations that attempt to increase a score by exploiting information already available to the evaluator, without introducing new task-specific evidence.

As with the degradation tests, for each candidate report x_i, we apply a manipulation strategy M to produce a perturbed report x_i^M. Table D.2 lists 6 strategies used in our experiments in Section 5.5.

We compute the paired score change $\Delta_i^{(M)}$ and Cohen's d_M using the same formulas established for degradations. For manipulations, the desirable behavior is $\bar{\Delta}^{(M)} \leq 0$ together with a small absolute effect size.

5.4 f-MI Metric Design Framework

We introduce an f-MI metric design framework that separates the design space into two conceptually distinct parts. The first part consists of theoretical design choices: which information measure is being estimated, and which estimator is used. The second part consists of empirical design choices: how reports are represented, and how predictive quantities are obtained from those representations.

Definition 5.1 (Design tuple)**.** *An f-MI evaluation metric is specified by a tuple*

$$S = (R, \Pi, F, E),$$

where:

(i) ***Information measure*** *F: the choice of f-divergence defining the objective $I_f(X;Y)$.*
(ii) ***Estimation method*** *E: the method used to estimate the chosen objective $I_f(X;Y)$.*
(iii) ***Representation*** *R: a map from the text space to a representation space, $R : \mathcal{V}^{\leq L} \to \mathcal{U}$.*
(iv) ***Prediction*** *Π: a mechanism that maps represented candidate-reference pairs to predictive quantities used in the estimation method.*

Intuitively, the representation R and prediction procedure Π determine what quantities can be estimated from text, while the information measure F and estimator E determine how those quantities are assembled into an f-MI score.

Given a dataset of n tasks $\{w_i\}_{i=1}^n$, each task w_i has an associated candidate report x_i, a positive reference report y_i, an optional negative reference report y_i^- sampled from the marginal distribution P_Y, and an optional superficial context z_i. For a fixed design tuple (F, E, R, Π), the metric assigns each item a sample-level score

$$S_{F,E,R,\Pi}(x_i, y_i, y_i^-), \quad \text{or} \quad S_{F,E,R,\Pi}(x_i, y_i, y_i^-; z_i).$$

The construction of a reference-based metric applies the framework above. To score a single item, we take the chosen MI estimation method and replace each expectation with an empirical average over the sampled positive and negative references. Averaging these per-item scores across the dataset then yields an estimator of the corresponding f-MI quantity determined by F, E, R, and Π:

$$\mathbb{E}\left[S_{F,E,R,\Pi}(x_i, y_i, y_i^-)\right] = \hat{I}_F^{(E,R,\Pi)}(X;Y), \quad \text{or} \quad \mathbb{E}\left[S_{F,E,R,\Pi}(x_i, y_i, y_i^-; z_i)\right] = \hat{I}_F^{(E,R,\Pi)}(X;Y \mid Z).$$

The rest of this section describes the definition of f-MI, the methods for MI estimation, representation, and prediction. Section 5.4.5 presents prior work under our decomposition. Section D.2 shows the approximate manipulation robustness of the metrics as a result of the information monotonicity, but under estimation errors.

5.4.1 f-Mutual Information

We first present the definition of the f-mutual information (f-MI).

Definition 5.2 (f-divergence and f-mutual information)**.** *Let P and Q be probability distributions on a finite space $\mathcal{X}$, and let $f : \mathbb{R}_+ \to \mathbb{R}$ be convex with $f(1) = 0$. The f-divergence is*

$$D_f(P\|Q) \coloneqq \sum_{x\in\mathcal{X}} q(x)\, f\left(\frac{p(x)}{q(x)}\right).$$

For jointly distributed random variables $(X, Y) \sim P_{XY}$, the f-mutual information is defined as the f-divergence between the joint distribution and the product distribution of X and Y:

$$I_f(X;Y) \coloneqq D_f\Big(P_{XY} \,\|\, P_X \otimes P_Y\Big).$$

Thus, $I_f(X;Y)$ measures how far the true joint distribution is from independence. When $f(t) = t\log t$, the f-MI recovers Shannon mutual information [Cover and Thomas, 1999].

The information monotonicity is instantiated by the information-processing inequality of an MI. This provides the information-theoretic foundation of our framework: degradations and manipulations can be viewed as post-processing a report, so a score built upon f-mutual information ideally should not increase under information-losing perturbations.

Proposition 5.3 (Information-Processing Inequality)**.** *For any mapping $\sigma : \mathcal{X} \to \mathcal{X}'$,*

$$I_f(\sigma(X);Y) \le I_f(X;Y).$$

As mentioned above, we will also use the conditional mutual information when the scoring metric uses a context z. Conditioning on Z removes dependence already explained by the shared context and isolates the residual information between the candidate and the reference. The conditional f-mutual information is defined as

$$I_f(X;Y \mid Z) \coloneqq \mathbb{E}_Z\left[D_f\Big(P_{XY|Z} \,\|\, P_{X|Z} \otimes P_{Y|Z}\Big)\right].$$

The central object underlying all f-MI computation is the density ratio between the joint and product-of-marginals distributions $r(x, y)$.

$$r(x,y) \coloneqq \frac{P_{XY}(x,y)}{P_X(x)\,P_Y(y)} = \frac{P_{Y|X}(y \mid x)}{P_Y(y)} = \frac{P_{X|Y}(x \mid y)}{P_X(x)}.$$

The population quantity $I_f(X;Y)$ can be computed exactly from r:

$$I_f(X;Y) = \mathbb{E}_{P_X\otimes P_Y}\Big[f(r(X,Y))\Big]. \tag{5.1}$$

5.4.2 MI Estimation

Since the vast combinatorial space of text prevents the explicit expression of joint distributions, I_f cannot be computed in closed form. To circumvent this, we explore two families of estimators: direct density ratio estimation and f-variational estimation. The main difference lies in whether the method requires estimating the density ratio $r(x, y)$ directly. For MI defined by the commonly used KL and TV divergences, direct density ratio estimation requires estimating $r(x, y)$, while the f-variational requirement is weaker.

5.4.2.1 Direct Density Ratio Estimation

By the density ratio formulation of MI in Equation (5.1), if the density ratio r is known, MI can be calculated via a Monte Carlo estimator by averaging $f(r(x, y))$ over samples $(x, y) \sim P_X \otimes P_Y$.

For KL, $f(t) = t \log t$, so

$$I(X;Y) = \mathbb{E}_{P_X \otimes P_Y}\Big[r(X,Y)\log r(X,Y)\Big] = \mathbb{E}_{P_{XY}}\Big[\log r(X,Y)\Big].$$

The quantity $\mathrm{PMI}(x;y) \coloneqq \log r(x,y) = \log p(y \mid x) - \log p(y)$ is the pointwise mutual information; its expectation under the joint distribution recovers the Shannon mutual information.

Direct estimation is also available for TVD-MI, where $f(t) = \frac{1}{2}|t-1|$, so we have

$$I_{\mathrm{TV}}(X;Y) = \frac{1}{2}\mathbb{E}_{P_X \otimes P_Y}\Big[|r(X,Y)-1|\Big] = \frac{1}{2}\sum_{x,y \in \mathcal{V}^{\leq L}} |P(x,y) - P(x)P(y)|.$$

This can also be written as $I_{\mathrm{TV}}(X;Y) = \frac{1}{2}\mathbb{E}_{P_{XY}}\Big[|1 - \frac{1}{r(X,Y)}|\Big]$, providing two approaches of Monte Carlo estimation.

5.4.2.2 f-Variational Estimation

Another commonly used estimator in the literature is the f-variational estimator, constructed via the Fenchel conjugate f^*:

$$I_f(X;Y) = \mathbb{E}_{P_{XY}}[T^*(X,Y)] - \mathbb{E}_{P_X \otimes P_Y}\Big[f^*(T^*(X,Y))\Big], \tag{5.2}$$

where $T^*(x) \in \partial f\left(\frac{p(x)}{q(x)}\right)$ is defined by the subgradient of f. Every f-divergence admits this variational lower bound.

Definition 5.4 (Variational representation)**.** *Let f^* denote the Fenchel conjugate of f,*

$$f^*(v) \coloneqq \sup_{t>0}\{vt - f(t)\}.$$

Then

$$D_f(P\|Q) = \sup_T \left\{\mathbb{E}_P[T(X)] - \mathbb{E}_Q\left[f^*(T(X))\right]\right\},$$

where the supremum is over functions T such that $f^*(T(x)) < \infty$ for all x with $Q(x) > 0$.

Here T is a critic: a scalar scoring function that tries to assign higher values to samples from P than to samples from Q. We write T^* for the population-optimal critic. When f is differentiable,

$$T^*(x) = f'\left(\frac{p(x)}{q(x)}\right),$$

and in general

$$T^*(x) \in \partial f\left(\frac{p(x)}{q(x)}\right),$$

where ∂f denotes the subdifferential of f.

The f-mutual information also admits the variational form

$$I_f(X;Y) = \sup_T \left\{\mathbb{E}_{P_{XY}}[T(X,Y)] - \mathbb{E}_{P_X\otimes P_Y}[f^*(T(X,Y))]\right\}.$$

At the population optimum, equality is attained when $T(x,y) = T^*(x,y)$.

In our setting, we can write $T(x,y)$ for a critic on candidate–reference pairs and $T^*(x,y)$ for the population-optimal critic. The optimal critic is determined by the chosen generator f through

$$T^*(x,y) \in \partial f\Big(r(x,y)\Big).$$

As Table 5.1 shows, different choices of f induce different optimal critics and different conjugate penalty terms. We refer back to this table when instantiating KL-based and TV-based evaluation metrics.

Estimator Here joint samples P_{XY} are candidate-reference pairs from the same task, while marginal samples $P_X \otimes P_Y$ are pairs drawn across different tasks. Recall that the optimum is attained by a critic satisfying $T^*(x,y) \in \partial f\Big(r(x,y)\Big)$. For common choices of f, the corresponding forms of $T^*(x,y)$ and $f^*(T^*(x,y))$ are listed in Table 5.1.

Table 5.1: Common f-divergences, along with a convenient population-optimal critic $T^*(x, y)$ and the corresponding conjugate term $f^*(T^*(x, y))$, written in terms of the density ratio $r(x, y)$.

f-divergence	$f(t)$	$T^*(x, y)$	$f^*(T^*(x, y))$
Total variation	$\frac{1}{2}\lvert t-1\rvert$	$\frac{1}{2}\operatorname{sign}\big(r(x,y)-1\big)$	$\frac{1}{2}\operatorname{sign}\big(r(x,y)-1\big)$
KL divergence	$t\log t$	$1+\log r(x,y)$	$r(x,y)$
Reverse KL	$-\log t$	$-\frac{1}{r(x,y)}$	$-1+\log r(x,y)$
Pearson χ^2	$(t-1)^2$	$2\big(r(x,y)-1\big)$	$r(x,y)^2-1$
Squared Hellinger	$(\sqrt{t}-1)^2$	$1-\frac{1}{\sqrt{r(x,y)}}$	$\sqrt{r(x,y)}-1$

For KL, the variational estimation yields the NWJ objective:

$$\begin{aligned} I(X;Y) &= \sup_T \Big\{ \mathbb{E}_{P_{XY}}[T(X,Y)] - e^{-1}\mathbb{E}_{P_X\otimes P_Y}\Big[e^{T(X,Y)}\Big] \Big\} \\ &= 1 + \mathbb{E}_{P_{XY}}\Big[\log r(X,Y)\Big] - \mathbb{E}_{P_X\otimes P_Y}\Big[r(X,Y)\Big]. \end{aligned}$$

For total variation, the estimator reduces to

$$\begin{aligned} I_{\mathrm{TV}}(X;Y) = &\sup_{\|T\|_\infty \le 1/2} \Big\{ \mathbb{E}_{P_{XY}}[T(X,Y)] - \mathbb{E}_{P_X\otimes P_Y}[T(X,Y)] \Big\} \\ =&\frac{1}{2}\mathbb{E}_{P_{XY}}\Big[\operatorname{sign}(r(X,Y)-1)\Big] - \frac{1}{2}\mathbb{E}_{P_X\otimes P_Y}\Big[\operatorname{sign}(r(X,Y)-1)\Big], \end{aligned}$$

where the optimal critic $\operatorname{sign}(r(X, Y) - 1)$ is equivalent to a binary classifier that distinguishes same-task pairs from cross-task pairs.

The f-variational estimator is computed given a task w_i, candidate x_i, positive reference y_i, and negative reference $y_i^- \sim P_Y$. The estimator is an unbiased estimator $S(x_i, y_i, y_i^-)$, a sample-level score. $S(x_i, y_i, y_i^-)$ estimates the population expectations in Equation (5.2) by replacing the joint expectation over P_{XY} with computation over (x_i, y_i), and the marginal expectation over $P_X \otimes P_Y$ with (x_i, y_i^-).

5.4.2.3 Examples

Table 5.2 summarizes the four cases and makes explicit the sample-level score $S(x_i, y_i, y_i^-)$ that the distribution-model component D needs to estimate. Averaging S over the n tasks then estimates the corresponding population f-MI, as required by the design of Definition 5.1.

Note that KL-direct and TV-direct both require estimating the density ratio $r(x, y)$, while the variational estimators only require a critic $T^*(x, y)$. In particular, TV-variational only needs a bounded classifier for whether $r(x, y)$ is above or below 1.

Table 5.2: Sample-level score and quantity to estimate for each (F, E) pair.

F	E	Sample-level score	Quantity to estimate
KL	Direct	$\log r(x_i, y_i)$	$\log r(x_i, y_i)$
KL	f-variational	$T^*(x_i, y_i) - e^{T^*(x_i, y_i^-)-1}$	$T^*(x_i, y_i) = 1 + \log r(x_i, y_i)$
TV	Direct	$\frac{1}{2}\left\|1 - r(x_i, y_i)^{-1}\right\|$ or $\frac{1}{2}\left\|r(x_i, y_i^-) - 1\right\|$	$r(x_i, y_i)$ or $r(x_i, y_i^-)$
TV	f-variational	$T^*(x_i, y_i) - T^*(x_i, y_i^-)$	$T^*(x_i, y_i) = \frac{1}{2}\,\mathrm{sign}(r(x_i, y_i) - 1)$

Producing these estimators from text requires two further choices: a representation R that fixes the granularity at which the report is processed (Section 5.4.3), and a prediction mechanism Π that maps represented pairs to the required predictive quantities (Section 5.4.4).

5.4.3 Representation

The representation $R : \mathcal{V}^{\leq L} \to \mathcal{U}$ fixes the units at which a report is processed by the downstream predictor. We consider three natural granularities.

- **Token**. At the finest granularity, a report y is represented by its token sequence $(y_1, \ldots, y_T)$. This preserves all information in the text and exposes the autoregressive factorization of language models, $p(y \mid x) = \prod_{t=1}^{T} p(y_t \mid y_{<t}, x)$, so that report-level conditional probabilities can be assembled from per-token factors. Token representation is only meaningful in combination with a predictor that consumes token-level signals, such as an autoregressive language model with logit access.

  In practice, prior work [Xu et al., 2025, Lu et al., 2024] reports that style-normalization pre-processing is important when using token representations, since autoregressive language models can be confounded by superficial features of the text such as writing style.

- **Full-Report**. At the coarsest granularity, a report is treated as an atomic unit, $R(y) = y$. Predictive quantities are obtained holistically from the candidate-reference pair (x, y): a single conditional likelihood $p(y \mid x)$, or a single oracle judgment about the pair (e.g., "do these two reports describe the same item?").

- **Statement**. An intermediate granularity decomposes a report into its atomic claims. Let

$$\Psi(y) = \left\{\psi_1(y), \ldots, \psi_{K(y)}(y)\right\}$$

  denote the statement decomposition of y, where each $\psi_k(y) \in \mathcal{V}^{\leq L}$ is an atomic statement, typically obtained via a separate LLM call. Statement-level representation

naturally loses information by the data-processing inequality (Proposition 5.3). In practice, however, isolating individual claims often improves the predictor's performance, especially for an LLM-oracle, which tends to reason more reliably about single facts than about long passages. With statement-level decomposition, the estimation quality may even improve. We compare these tradeoffs empirically in Section 5.3 and Section 5.5.2.

5.4.4 Prediction Mechanism

The prediction mechanism Π maps a represented candidate-reference pair to the predictive quantities in the representation space, as required by the chosen estimator E. See Table 5.2 for examples of required predictive quantities. We describe two prediction mechanisms below.

5.4.4.1 Autoregression

An autoregressive language model with parameters ϕ assigns a probability to any token sequence via the chain rule. For a candidate report x in the prompt, we obtain

$$p_\phi(y \mid x) = \prod_{t=1}^{T} p_\phi(y_t \mid y_{<t}, x),$$

where each factor is read off the softmax over output logits. Autoregression interacts with the token-level representation. It also requires logit access, which some commercial APIs do not expose, and inherits any miscalibration of the underlying language model.

For the example of KL-based MI estimation, substituting a null report for x yields a marginal estimate $p_\phi(y)$. The following PMI is thus obtained for direct KL estimation:

$$\widehat{\mathrm{PMI}}(x; y) = \log r(x, y) = \log p_\phi(y \mid x) - \log p_\phi(y).$$

Exponentiating gives a density-ratio estimate that feeds any of the four (F, E) pairs in Table 5.2.

5.4.4.2 LLM-Oracle

When token-level probabilities are unavailable, an LLM can be prompted for a prediction about a represented pair. The prediction is then used as inputs to the estimators as required.

LLM-oracle prediction composes with the whole-report and statement representations. At the whole-report level, the prompt simply contains x and y and the oracle returns a single prediction. At the statement level, the candidate report x is paired with each $\psi_k(y)$,

and the resulting per-statement quantities are combined into a report-level prediction by an aggregation rule $A : \bigcup_{K \geq 1} \mathbb{R}^K \to \mathbb{R}$.

The following two regimes of an LLM oracle cover the cases in Table 5.2.

- Critic-style prediction fits naturally into the f-variational framework: a scalar or categorical LLM is prompted to directly output a critic used as $T^*(x, y)$. For example, Robertson and Koyejo [2025] shows that the TVD-MI is especially convenient. The T^* required in Equation (5.2) for TVD-MI can be implemented as a binary classifier distinguishing same-task from cross-task pairs. A simple prompt, e.g., "Do these two reports describe the same item?" implements this T^* for TVD-MI.

- Likelihood-style prediction supports direct estimation. The LLM is asked, for instance, how strongly the candidate x supports the reference y, using a small set of ordered probability or support bins [Lu et al., 2024]. Mapping the selected bin to a numeric value yields a coarse surrogate for $\log \hat{p}(y \mid x) - \log \hat{p}(y)$, and hence for the density ratio $\hat{r}(x, y)$. Such direct LLM-oracle estimators are typically biased. Their outputs are discretized and often poorly calibrated. But they provide a workable approximation when only API access is available.

5.4.5 Prior Work as Instances of the Framework

Combining the preceding components, a complete design tuple (F, E, R, Π) specifies a concrete reference-based metric. Table 5.3 shows how existing methods instantiate this framework.

Table 5.3: Existing methods as instances of the design tuple (F, E, R, Π).

Method	F	E	R	Π
GEM [Xu et al., 2025] / GPPM-token [Lu et al., 2024]	KL	Direct (PMI)	Token	Autoregression
GPPM-judgment [Lu et al., 2024]	KL	Direct (PMI)	Statement	LLM-oracle
TVD-MI [Robertson and Koyejo, 2025]	TV	f-variational	Whole-report	LLM-oracle

A practical benefit of this decomposition is that it makes unexplored combinations easy to identify. Cells of the tuple (F, E, R, Π) that Table 5.3 leaves blank correspond to metrics that are implementable in principle but have not been studied. In Section 5.5, we empirically study several such metrics, as shown in Table 5.5. We also provide a formal definition of the ($F = \text{TV}$, $E = f$-variational, $R = \text{Statement}$, $\Pi = \text{LLM-oracle}$) metric as an example in Appendix D.3.

5.5 Experiment

We compare scoring methods in a controlled way under our test principles (Section 5.3); the design space is laid out in Section 5.5.1.

Research Question. To keep the analysis focused, we frame the experiments around the decision a practitioner actually faces. The predictor Π and representation R are typically dictated by task and deployment constraints, e.g., whether logit access is available, and whether atomic statements can be reliably extracted, while the information measure F and estimator E are under the researcher's direct control. We therefore ask:

Given a fixed predictor and representation pair (Π, R), which information measure and estimator (F, E) yields the best-behaved evaluation metric?

Other reference-based baselines. We also compare against popular reference-based metrics as baselines, including ROUGE-L, BLEU, BERTScore (based on roberta-large), and LLM-as-a-Judge (based on various models). These baselines test whether accuracy and manipulation-resistance gains from the MI framework go beyond lexical overlap, embedding similarity, or direct LLM judging.

Datasets. We evaluate metrics across three domains that stress different aspects of reference-based text evaluation: peer review, summarization, and question answering. In peer review, the task input is a paper or essay and the candidate reports are full reviews; in summarization, the task input is a source document and the reports are summaries; in question answering, the task input is a question and the reports are answers. Across these domains, reports differ substantially in length, structure, subjectivity, and semantic complexity. Our evaluation includes seven datasets: three peer-review datasets, two summarization datasets, and two question-answering datasets. Four datasets provide item-level human quality annotations and are used to measure human-rating correlation; all datasets are used for degradation and manipulation tests. When multiple independent reports are available for the same task, we treat them as positive references. Table 5.4 summarizes the datasets.[1] Note that not all datasets carry human quality annotations; those that do are used for alignment evaluation, while the remainder are used only for degradation, manipulation, and model-differentiation tests.

[1]Peer grading-WH [Wu and Hartline, 2024] and Peer grading-XLSK [Xu et al., 2025] are not publicly accessible while the others are public datasets.

Table 5.4: Overview of evaluation datasets. n is the number of candidate reports; #ref is the typical number of positive references per task. "Human rating" indicates whether item-level human quality annotations are available for candidate reports.

Domain	**Dataset**	n	Ref	**Human rating**	**Reference**
Peer review	ICLR 2026	300*	peer	No	OpenReview
	Peer grading-WH	534	GT	Numerical	Wu and Hartline [2024]
	Peer grading-XLSK	165	peer	Numerical	Xu et al. [2025]
Summarization	SummEval	800*	GT	Numerical	Fabbri et al. [2021]
	SPACE	50	GT	No	Angelidis et al. [2021]
QA	LFQA-E	320*	GT	Pairwise	Fan et al. [2026]
	MedAESQA	400*	GT	Numerical	Gupta et al. [2025]

* Down-sampled from the whole dataset for a more trackable test scale.

5.5.1 Experiment Design

To compare scoring methods in a controlled way, we organize our experiments around the design space induced by Definition 5.1. Each scoring method corresponds to a choice along four axes:

(i) the information measure $F \in \{\mathrm{KL}, \mathrm{TV}\}$;

(ii) the estimator $E \in \{\text{Direct}, f\text{-variational}\}$;

(iii) the representation $R \in \{\text{Token}, \text{Whole-report}, \text{Statement}\}$;

(iv) the predictor $\Pi \in \{\text{Autoregression}, \text{LLM-oracle}\}$.

The predictor and representation do not vary independently: autoregression composes only with token-level representation, while the LLM-oracle composes with whole-report or statement-level representation (Section 5.4.4). The joint (Π, R) axis therefore has three valid settings, yielding a $2 \times 2 \times 3 = 12$-cell factorial design (Table 5.5). Of the 12 cells, three correspond to previously proposed methods (GEM, GPPM-Judgment, and TVD-MI). By holding three axes fixed and varying one, we can isolate the marginal effect of each design choice on alignment, degradation sensitivity, and manipulation resistance.

5.5.2 Results

Table 5.6 reports Spearman correlation with human ratings on the four datasets carrying item-level numerical annotations: Peer Grading-WH [Wu and Hartline, 2024], Peer Grading-

Table 5.5: Coverage of the design space. Each cell is labeled by the name of the corresponding scoring method, where applicable. NEW denotes combinations introduced or evaluated in this work.

		Autoregression	**LLM-oracle**	
F	E	Token-representation	Report-representation	Statement-representation
KL	Direct	GEM	NEW	GPPM-J
	f-variational	NEW	NEW	NEW
TV	Direct	NEW	NEW	NEW
	f-variational	NEW	TVD-MI	NEW

XLSK [Xu et al., 2025], SummEval [Fabbri et al., 2021], and MedAESQA [Gupta et al., 2025]. Rows are grouped by the representation and predictor pair (R, Π); within each block, the four rows sweep $(F, E) \in \{\mathrm{KL}, \mathrm{TV}\} \times \{\text{Direct}, f\text{-var.}\}$. Bold marks the best method within each block; underlines mark the best f-MI metric and the best overall metric per dataset.

For degradation and manipulation tests, a full sweep of all configurations across every perturbation strategy and dataset is computationally prohibitive. We therefore carry forward the best f-MI configurations from each representation-prediction block in Table 5.6: $(\mathrm{KL}, \text{Direct})$ with token-level autoregression, corresponding to GEM; and $(\mathrm{TV}, f\text{-var.})$ with an LLM-oracle at both report and statement levels. This selection deliberately favors the correlation winners and asks whether they remain reliable under strategic perturbations. For large datasets, we down-sample to roughly 200 candidate reports per setting. We report result overviews in Tables 5.7 and 5.8, and detailed Cohen's d statistics in Section D.5.

Key takeaways. The results show a separation between statistical and strategic alignment.

- **LLM-as-a-Judge is strongest on human-rating correlation, but collapses under manipulation.** Among the non-MI baselines, LLM-as-a-Judge achieves the strongest correlations with human ratings. By the standard statistical alignment alone, it appears to be the best metric. However, across the four judge models, manipulations succeed in 18 to 26 of the 30 tests.

- **The best f-MI design depends on the representation and predictor.** Under token-level autoregression, $(\mathrm{KL}, \text{Direct})$ (GEM) leads its block on three of four datasets and is the best f-MI metric overall on both peer-review datasets. Under an LLM-oracle, statement-level representation is substantially more effective: statement-level

Table 5.6: Human-rating correlation results. Rows are scoring methods, grouped by distribution model D and its granularity; within each group, rows sweep the information measure $F \in \{\text{KL}, \text{TV}\}$ and estimator $E \in \{\text{Direct}, f\text{-var.}\}$. Columns are datasets; entries are Spearman rank correlations with human ratings. For SummEval we report the mean over all dimensions. Best result within each group is bolded; best among f-MI metrics per column and best overall per column are underlined.

Evaluation Metric	**Peer review**		**Summarization**	**QA**
(F, E)	PG-WH	PG-XLSK	SummEval	MedAESQA
Autoregression, Token-representation				
(KL, Direct) (GEM)	**<u>0.460</u>**	**<u>0.475</u>**	**0.138**	0.113
(KL, f-var.)	-0.218	-0.355	0.059	0.051
(TV, Direct)	0.312	0.380	0.095	-0.025
(TV, f-var.)	-0.053	-0.036	0.085	**0.129**
LLM-oracle, Report-representation				
(KL, Direct)	0.235	0.367	0.031	**0.163**
(KL, f-var.)	0.252	**0.419**	0.031	0.162
(TV, Direct)	-0.231	-0.350	0.017	-0.238
(TV, f-var.) (TVD-MI)	**0.286**	0.042	**0.140**	0.110
LLM-oracle, Statement-representation				
(KL, Direct) (GPPM-J)	0.254	0.307	0.073	0.041
(KL, f-var.)	**0.281**	0.251	0.075	0.064
(TV, Direct)	-0.292	-0.396	-0.043	-0.129
(TV, f-var.)	0.201	**0.468**	**<u>0.208</u>**	**<u>0.228</u>**
Non-MI baselines				
ROUGE-L	-0.244	0.167	0.111	0.151
BLEU	-0.332	0.256	0.102	0.127
BERTScore	-0.133	-0.063	0.252	0.166
LLM-as-Judge / Claude-Haiku-4.5	0.462	0.569	0.400	**<u>0.315</u>**
LLM-as-Judge / Claude-Sonnet-4.5	0.512	0.622	0.450	0.301
LLM-as-Judge / GPT-5-mini	**<u>0.539</u>**	**<u>0.631</u>**	**<u>0.477</u>**	0.256
LLM-as-Judge / GPT-4o-mini	0.492	0.528	0.452	0.208

Table 5.7: Sensitivity to degradation. For each (metric, dataset) pair, we apply m degradation strategies (header) and count those on which the metric fails to produce a significant score decrease at $p < 0.05$; lower is better. A ✓ marks zero failures; otherwise, letter codes mark the strategies the metric missed (legend below).

Evaluation Metric	**Peer review**			**Summarization**		**QA**		**total**
(F, E, R, Π)	PG-WH $m=5$	PG-XLSK $m=6$	ICLR26 $m=6$	SummEval $m=5$	SPACE $m=5$	LFQA-E $m=5$	MedAESQA $m=5$	fail $m=37$
MI-based Mechanisms								
(KL, Direct, Token, Autoregression)	✓	✓	P	✓	✓	P U	P U	5
(TV, f-var, Report, LLM-oracle)	D S U	O	O P U	P U	✓	P U	D S P U	15
(TV, f-var, Statement, LLM-oracle)	✓	✓	✓	P	✓	P	P	3
Non-MI baselines								
LLM-as-Judge / Claude-Haiku-4.5	✓	✓	P	P	✓	P	D P	5
LLM-as-Judge / Claude-Sonnet-4.5	✓	✓	P	P	✓	P	P	4
LLM-as-Judge / GPT-4o-mini	D	D O R	D O R P	P	D	P	P	12
LLM-as-Judge / GPT-5-mini	✓	✓	P	P	✓	P	D P	5

Codes: D=deletion and completion; O=opinion flip; R=random replacement.
Codes: S=sentence deletion; P=surface report; U=ultra concise compression.
Codes mark failed degradation criteria.

Table 5.8: Manipulation resistance. For each (metric, dataset) pair, we apply m manipulation strategies (header) and count those the metric fails to resist, i.e., that produce a significant score increase at $p < 0.05$; lower is better. A ✓ marks zero failures; otherwise, letter codes flag the strategies the metric failed to resist (legend below).

Evaluation Metric	**Peer review**			**Summarization**		**QA**		**total**
(F, E, R, Π)	PG-WH $m=6$	PG-XLSK $m=6$	ICLR26 $m=6$	SummEval $m=2$	SPACE $m=2$	LFQA-E $m=4$	MedAESQA $m=4$	fail $m=30$
MI-based Mechanisms								
(KL, Direct, Token, Autoregression)	✓	S	N H P S R	✓	✓	✓	✓	6
(TV, f-var, Report, LLM-oracle)	✓	✓	✓	✓	✓	✓	✓	**0**
(TV, f-var, Statement, LLM-oracle)	✓	✓	✓	✓	✓	✓	✓	**0**
Non-MI baselines								
LLM-as-Judge / Claude-Haiku-4.5	M N H S R	H P S R	H S R	R	R	H S R	H S R	20
LLM-as-Judge / Claude-Sonnet-4.5	M N H S R	H P S R	M H P S R	R	✓	H S	H	18
LLM-as-Judge / GPT-4o-mini	M N H P S R	M N H P S R	M N H P S R	R	R	H S R	H S R	26
LLM-as-Judge / GPT-5-mini	N H S R	M N H P S R	H P S R	✓	✓	H S R	H S	19

Codes: M=meaningless elongation; N=negative opinion shift; H=hedged opinion shift.
Codes: P=positive opinion shift; S=strong opinion shift; R=rephrase.
Codes mark failed manipulation criteria.

(TV, f-var.) outperforms report-level TVD-MI on three of four datasets and is the best f-MI metric overall on SummEval and MedAESQA.

- **The unified design framework identifies a dominating new metric.** The statement-level (TV, f-var.) metric is not present in prior work, but is exposed by our design framework. Its strong performance suggests that decomposing references into atomic claims can make the oracle prediction task easier, enough to compensate for the information loss introduced by statement extraction. The (TV, f-var.) metric is the most reliable on degradation sensitivity and manipulation robustness. It fails only 3 of 37 degradation tests, compared with 5 failures for GEM and 15 failures for report-level (TV, f-var.) / TVD-MI. It has 0 failures out of 30 manipulation tests, meaning that no tested manipulation produces a significant score increase.
- **Robustness of MI-based metrics comes from the design, not merely from the model.** Since Claude-Haiku-4.5 is vulnerable as an absolute LLM judge but robust when used inside the TV f-variational metric, the improvement comes from the MI-based formulation.

It is worth to notice that no single axis of (F, E, R, Π) dominates in isolation; rather, the design choices interact. Token-level autoregression favors KL-direct estimation, while LLM-oracle prediction benefits from the TV f-variational formulation, especially with statement-level representations. MI-based metrics also appear stronger on peer-review tasks than on summarization and QA, where the evaluation target can be more complex or subjective.

ROUGE-L, BLEU, and BERTScore are generally weaker baselines, and they fail to obtain positive correlation on all datasets, especially in peer review. This suggests that lexical overlap and embedding similarity are insufficient for evaluating richer, more subjective reports.

Additional discussion. On ICLR2026, GEM fails the rephrase manipulation test, which appears to contrast with the original GEM results [Xu et al., 2025], where rephrasing did not produce a significant score increase on ICLR2023. A likely explanation is that our rephrase manipulation is stronger. Whereas Xu et al. [2025] primarily prompt the model to improve language quality, our prompt also permits the model to reorganize the review. Although it prohibits adding new facts or changing the reviewer's stance, this reorganization can make the candidate review easier for the metric to match with the reference. The result therefore suggests that GEM retains some sensitivity to presentation quality. However, the effect size (Cohen's d) for GEM remains relatively small ($+0.18 \pm 0.14$, 95% CI), especially compared with direct LLM judging, even for the two most robust judge models: Claude-Sonnet-4.5 ($+0.38 \pm 0.14$, 95% CI) and GPT-5-mini ($+0.24 \pm 0.14$, 95% CI), as shown in Appendix D.5.

This observation reinforces the value of the MI-based formulation, and further motivates the new statement-level (TV, f-var.) metric, which remains robust to this stronger rephrasing test.

We also noticed that surface-report degradation is a strong test because the degraded reports are still fluent and task-specific, generated from weak but informative context such as a news lead, question, or paper abstract, by Claude-4.5-Haiku. Thus, they can remain plausible even while sometimes omitting important task-relevant information.

Under this test, MI-based metrics also compare favorably with the most robust LLM judges. The desired Cohen's d is significantly negative. MI-based metrics consistently pass more tests than LLM judges. Moreover, as shown in Appendix D.5, all MI-based metrics either significantly decrease on surface reports or remain close to insensitive, with no significant positive failures. By contrast, even the strongest LLM judges can not penalize but reward surface reports: Claude-Sonnet-4.5 gives $d = +0.26 \pm 0.14$ (95% CI) on SummEval, $+0.60 \pm 0.15$ on LFQA-E, and $+0.83 \pm 0.16$ on MedAESQA, while GPT-5-mini gives $+0.09 \pm 0.14$, $+0.51 \pm 0.15$, and $+0.75 \pm 0.16$ on the same datasets. Thus, surface reports expose a hard case for all metrics, but the failure mode differs: MI-based metrics can sometimes be under-sensitive, whereas LLM judges can directionally fail by rewarding plausible surface-level reports.

5.6 Related Work

NLG Evaluation Evaluation of AI systems spans many dimensions. Holistic frameworks consider accuracy, robustness, fairness, and toxicity [Liang et al., 2022]; alignment research emphasizes helpfulness, honesty, and harmlessness [Askell et al., 2021]; and a growing body of work targets truthfulness [Lin et al., 2022], the multi-faceted nature of fairness [Gallegos et al., 2024], and pluralism over legitimate human perspectives [Sorensen et al., 2024].

Our framework instead targets a complementary and foundational property, semantic informativeness: whether an evaluation score faithfully reflects how much useful, truthful, task-relevant information a report conveys. A response cannot be genuinely helpful if it carries little task-relevant signal, and mechanisms that reward truthful, informative reporting can discourage guessing and some forms of hallucination [Wu and Hartline, 2024].

Mutual Information Estimation Machine learning has produced sample-based MI estimators built on the variational representation of f-divergences via Fenchel duality [Nguyen et al., 2010], and neural network estimators such as MINE [Belghazi et al., 2018], Deep InfoMax [Hjelm et al., 2018], and InfoNCE [van den Oord et al., 2018].

In information elicitation without verification, MI plays a different role: Kong and Schoenebeck [2019] score each agent a measure of MI between her report and a peer's reference, using the data-processing inequality to make truthful reporting an equilibrium. This echoes the same strategic-alignment principle behind our MI-based metrics (Proposition 5.3). Kong and Schoenebeck [2018b] extends the same MI-maximization principle to co-training, where two Bayesian predictors over distinct views of the data play the role of the two agents.

Closest to our setting are pre-trained language model approaches to MI estimation: Padmakumar and He [2021] compute pointwise MI between extractive summaries and source articles, while Xu et al. [2025], Lu et al. [2024], Robertson and Koyejo [2025] develop reference-based MI metrics for NLG evaluation, which can be recovered as specific instances by our design framework.

CHAPTER 6

Conclusion and Future Directions

This dissertation studies how to elicit and evaluate textual information when gold-standard references are unavailable or prohibitively expensive. Its central claim is that strategic alignment and statistical alignment should not be developed in isolation. A useful evaluation must do more than correlate with human ratings on a fixed dataset: it should also reward truthful, effortful, and informative reports over low-effort, superficial, or strategically manipulated ones. Throughout, I focus on semantic informativeness as the target property: whether a report conveys useful, truthful, and task-relevant information.

The dissertation makes four main contributions. First, it introduces Spot Check Equivalence as an interpretable way to compare information-elicitation mechanisms, clarifying how statistical alignment relates to strategic alignment. Second, it extends peer prediction from text through GPPM and GSPPM, showing that LLM-based prediction can be used to elicit informative textual reports even without gold-standard references. Third, it develops GEM and GEM-S as evaluation metrics that retain the strategic alignment of GPPM and GSPPM while improving statistical alignment for evaluation, and uses them to construct GRE-bench for benchmarking LLM-generated peer reviews. Fourth, it proposes a systematic design-and-test framework for aligned text evaluation metrics based on f-mutual information, enabling controlled comparison of existing and new metrics across human rating correlation, degradation sensitivity, and manipulation resistance.

Taken together, these contributions bridge information elicitation, information theory, and language model evaluation. More broadly, they suggest that automated text evaluation should be understood as an alignment problem rather than only a correlation problem. A useful evaluator should agree with human judgments, penalize degraded or low-effort reports, and resist manipulations that inflate scores without adding genuine information. The methods developed here do not solve this problem completely, but they show that meaningful progress is possible even in open-ended settings where trusted references are weak or unavailable.

As large language models play a growing role in everyday life, the need for reliable evaluation

will only become more important. I hope this dissertation helps move toward evaluation metrics and information elicitation mechanisms that are more robust, more interpretable, and better aligned with the information we actually want to reward. The next section outlines several directions in which this research can be extended.

6.1 Future Directions

Finer-grained alignment of informativeness. A natural next question is not only whether a report is informative, but also what kind of information it contributes. Future work could investigate which judgments, statements, or tokens are most closely related to semantic informativeness and to alignment with human quality judgments.

More broadly, the conditioning ideas behind GEM-S suggest that score changes may reflect different layers of content, such as information already contained in a synopsis, information that repeats author-stated claims, and genuinely new judgment. At present, however, these distinctions remain heuristic. A more general direction would be to project reports into a decision-relevant space and study informativeness there. Many reports ultimately support a downstream decision, for example, peer reviews contribute to paper acceptance decisions. This raises a natural question: can evaluation focus on the information a report provides about the decision itself, and can mutual-information-based methods be adapted to measure alignment in that space?

Hybrid evaluation under limited verification. A second direction is to study how automated evaluation should be combined with selective spot-checking. In many applications, trusted evaluation by human experts is too expensive to deploy at scale, while purely automated metrics such as LLM-as-Judge remain vulnerable to gaming. A natural hybrid is to use human experts to spot-check for strategic manipulation, keeping the distribution of reports within the regime where the automated judge is known to perform well. This raises several open questions: under a fixed budget or limited-liability constraint, can a hybrid mechanism be designed that provably preserves both statistical and strategic alignment? if so, how much spot-checking is enough? How should spot-checks be allocated across reports?

Strategic behavior in real text settings. The current experiments show that the proposed mechanisms respond to degradations and several manipulations, but richer forms of strategic behavior remain open. In practice, users may rely on LLM assistance, style transfer, prompt engineering, or other strategies that are harder to capture through synthetic perturbations alone. Understanding such behavior is important for evaluating whether an

aligned metric remains robust once it becomes a target of optimization.

A useful next step would therefore be to study strategic behavior in naturalistic text settings rather than only in simulated ones. For example, one could conduct classroom or peer-assessment experiments in which LLM-as-a-Judge is used as part of the grading or feedback process, and participants are given incentives to improve their scores. Such a setting would make it possible to observe how people actually try to game automated evaluation, and to test which mechanisms remain robust when strategic behavior is observed in practice rather than approximated in advance.

APPENDIX A

Proofs for Spot Check Equivalence

A.1 Proof of Main Theorem

Theorem 2.15 (Main Theorem)**.** *For a given performance measurement M within an information elicitation context IEC where every agent exerts effort level ξ, when Assumptions 2.12, 2.13, and 2.14 are satisfied, there exists a linear bijection between the* $\mathrm{MI}_{Q,\mathrm{corr}}(IEC \leftarrow M)$ *and the Sensitivity $\delta(\xi)$, where* corr *is the sample Pearson correlation coefficient and the number of agents goes to infinity.*

Proof. Recall the definition of Measurement Integrity, we have that

$$\underset{Q,\mathrm{corr}}{\mathrm{MI}}\left(IEC \leftarrow M\right) = \mathbb{E}_{IEC}\left[\mathrm{corr}\left(\mathbf{s}, \mathbf{q}\right)\right]$$

where $\mathbf{s}, \mathbf{q}$ represents the vector of all agents' performance scores and report qualities respectively.

In the above definition, we apply the *sample Pearson correlation coefficient* to corr, i.e.,

$$\mathrm{corr}(\mathbf{s}, \mathbf{q}) = r(\mathbf{s}, \mathbf{q}) = \frac{\sum_i \left(s_i - \bar{s}\right)\left(q_i - \bar{q}\right)}{\sqrt{\sum_i \left(s_i - \bar{s}\right)^2}\sqrt{\sum_i \left(q_i - \bar{q}\right)^2}}$$

where $\bar{s} = \frac{1}{|I|}\sum_i s_i$ (the average performance score); and analogously for $\bar{q}$.

Recall that, by Assumption 2.14, we have both that, when the number of agents is large, the agents' qualities are independent and the performance scores of the agents are independent. Thus, we can regard each agent's quality and performance score pair as a sample from a joint distribution. In addition, the sample correlation coefficient r is a consistent estimator of the *population Pearson correlation coefficient* ρ as the sample size grows large, which is defined as

$$\mathrm{corr}(s, q) = \rho(s, q) = \frac{COV(s, q)}{\sigma_s \cdot \sigma_q}$$

where s, q are random variables representing the score and quality of one agent respectively.

Therefore, we have the following proposition.

Proposition A.1.

$$\lim_{|I|\to+\infty} \mathbb{E}[r(\mathbf{s}, \mathbf{q})] = \rho(s, q)$$

where s, q are random variables representing the score and quality of one agent respectively.

We then show that there exists a bijection between $\rho(s, q)$ and $\delta(\xi)$. Recall the definition of Sensitivity

$$\delta(\xi) = \frac{\frac{\partial}{\partial e_i}\mu_s(e_i)|_{e_i=\xi}}{\sigma_s(\xi)}$$

We can find that there is $\frac{1}{\sigma_s(\xi)}$ in both the Sensitivity $\delta(\xi)$ and correlation coefficient $\rho(s, q)$, so we only need to find a bijection between $\frac{COV(s,q)}{\sigma_q}$ and $\frac{\partial}{\partial e_i}\mu_s(e_i)|_{e_i=\xi}$.

In the following discussion, since bias does not affect the correlation coefficient, we can assume $\xi = 0$ without loss of generality.

A.1.0.1 From the MI side

Here, every probability is conditioned on $e = \xi$.

$$\begin{aligned}
COV(s, q) =& \mathbb{E}[s \cdot q] - \mathbb{E}[s] \cdot \mathbb{E}[q] \\
=& \int_{q,s} s \cdot q \cdot \Pr[s, q] \,\mathrm{ds}\,\mathrm{dq} && (\mathbb{E}[q] = \xi = 0) \\
=& \int_{q,s} s \cdot q \cdot \Pr[s|q] \cdot Pr[q] \,\mathrm{ds}\,\mathrm{dq} \\
=& \int_{q} q \cdot Pr[q] \left(\int_{s} s \cdot \Pr[s|q] \,\mathrm{ds} \right) \mathrm{dq} \\
=& \int_{q} q \cdot Pr[q] \cdot \mu_{s|q}(q) \,\mathrm{dq}
\end{aligned}$$

A.1.0.2 From Sensitivity side

$$\begin{aligned}
\frac{\partial}{\partial e}\mu_s(e) =& \frac{\partial}{\partial e} \int_q \mu_{s|q}(q) \Pr[q|e] \,\mathrm{dq} \\
=& \int_q \mu_{s|q}(q) \frac{\partial}{\partial e} \Pr[q|e] \,\mathrm{dq}
\end{aligned}$$

Here, since q follows a normal distribution with mean e and standard deviation σ_q, we

have that

$$\begin{aligned}\frac{\partial}{\partial e}\Pr[q|e] =& \frac{\partial}{\partial e}\frac{1}{\sqrt{2\pi}\sigma_q}\exp -\frac{(q-e)^2}{2\sigma_q(e)^2} \\ =& \frac{1}{\sqrt{2\pi}\sigma_q(e)^3}\exp -\frac{(q-e)^2}{2\sigma_q(e)^2}\left((q-e)+\frac{(q-e)^2\sigma_q'(e)}{\sigma_q(e)}\right) \\ =& \frac{1}{\sigma_q^2}\cdot q\cdot \Pr[q|e] \qquad \text{(Assumption 2.12)}\end{aligned}$$

Thus, by combining the above two equations, we have

$$\begin{aligned}\frac{\partial}{\partial e}\mu_s(e) =& \int_q \mu_{s|q}(q)\cdot\frac{1}{\sigma_q^2}\cdot q\cdot \Pr[q|e]\,\mathrm{dq} \\ =& \frac{1}{\sigma_q^2}\int_q q\cdot Pr[q|e]\cdot\mu_{s|q}(q)\,\mathrm{dq}\end{aligned}$$

Letting $e=\xi$, we have

$$\begin{aligned}\frac{\partial}{\partial e}\mu_s(e) =& \frac{1}{\sigma_q^2}\int_q q\cdot Pr[q|e=\xi]\cdot\mu_{s|q}(q)\,\mathrm{dq} \\ =& \frac{COV(s,q)}{\sigma_q^2}\end{aligned}$$

Therefore, we get the linear bijection between Sensitivity and correlation is

$$\delta = \mathrm{corr}(s,q)/\sigma_q.$$

□

A.2 Simulation Details

To show the effectiveness of the Measurement-Integrity-based Spot Check Equivalence, we can plug different performance measurements in the information elicitation context described in the previous subsection, then calculate the Measurement Integrity (which implies the Spot Check Equivalence) and the motivational proficiency (i.e. the total payment) respectively, and finally, observe the correlation between the Spot Check Equivalence and the motivational proficiency.

For a given information elicitation context and a target equilibrium effort level ξ, we propose the following workflow for one sample:

1. Set all the agents' effort levels at ξ.

   (a) Generate a sample for the application abstraction according to the data-generating process D.

   (b) Use the performance measurement M to produce all the agents' scores $\mathbf{s}$.

   (c) Use the quality function Q to calculate the quality of the agents' report, and then calculate the Pearson correlation coefficient.

2. Change agent 1's effort level to $e_1 = \xi - \varepsilon$

   (a) Generate another sample for the application abstraction according to the data-generating process D.

   (b) Use the performance measurement M to produce all the agents' scores $\mathbf{s}'$.

Calculating the Measurement Integrity. We sample 5000 times and use the mean of the Pearson correlation coefficient in the samples[1] as the Measurement Integrity of performance measurement M. We apply the same workflow to all the performance measurements in Section 2.5.1.3, as well as spot-checking performance measurements with different checking ratios.

Calculating the total payment. We sample 5000 times and use the agents' scores $\mathbf{s}$ and $\mathbf{s}'$ to estimate the $\frac{\partial}{\partial e_i}\mathbb{E}[\#\text{beaten}|e_i, e_{-i} = \xi]|_{e_i=\xi}$, which can be used to compute the total payment for the Borda-count payment schemes.

[1] It is a consistent estimator of the Measurement Integrity.

APPENDIX B

Appendix for Generative Peer Prediction

B.1 Omitted Proofs

Proposition 3.12. *For the Generative Peer Prediction Mechanism (GPPM), when Assumption 3.9 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.

Proof of Proposition 3.12. 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

$$\begin{aligned}
&\sum_{x_j} \Pr[X_j = x_j \mid X_i = x_i] \log \Pr_{\mathrm{LLM}(\psi)}[X_j = x_j \mid X_i = x_i] && \text{(Assumption 3.9)} \\
\geq &\sum_{x_j} \Pr[X_j = x_j \mid X_i = x_i] \log \Pr[X_j = x_j \mid X_i = x_i] - \epsilon \\
= &- H(X_j \mid X_i = x_i) - \epsilon,
\end{aligned}$$

Taking the expectation of X_i, her expected score is

$$
\begin{aligned}
&\sum_{x_i} \Pr[X_i = x_i] \sum_{x_i} \Pr[X_j = x_j \mid X_i = x_i] \log \Pr_{\mathrm{LLM}(\psi)}[X_j = x_j \mid X_i = x_i] \\
\geq &\sum_{x_i} \Pr[X_i = x_i] \left(\sum_{x_j} \Pr[X_j = x_j \mid X_i = x_i] \log \Pr[X_j = x_j \mid X_i = x_i] - \epsilon \right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Assumption 3.9)} \\
= &\sum_{x_i, x_j} \Pr[X_i = x_i, X_j = x_j] \log \Pr[X_j = x_j \mid X_i = x_i] - \epsilon \\
= &- H(X_j \mid X_i) - \epsilon,
\end{aligned}
$$

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

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

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

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 strictly 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 $I(X_i; X_j) - \epsilon = H(X_j) - H(X_j \mid X_i) - \epsilon \geq H(X_j \mid$

$X_i^l) - H(X_j \mid X_i) - \epsilon = I(X_i; X_j \mid X_i^l) - \epsilon.$

□

Proposition 3.13. *For the* Generative Synopsis Peer Prediction Mechanism *(GSPPM), when Assumption 3.9 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.

Proof of Proposition 3.13. 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

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

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. □

Proposition 3.14. *For the Generative Peer Prediction Mechanism (GPPM), when Assumption 3.9 holds with parameter $\epsilon \geq 0$. When $I(X_i; X_j \mid X_i^l) > 0$, GPPM is $\alpha\epsilon$-potent, where*

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

When $\epsilon = 0$, GPPM is potent.

Proof of Proposition 3.14. Proposition 3.12 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,

$$\begin{aligned}
&U_i((\tau, c_h), (\tau, c_h)) - U_i((\sigma_i, c_h), (\tau, c_h)) \\
=&\alpha \left(\mathbb{E}[\text{GPPM}(X_i, X_j)] - \mathbb{E}[\text{GPPM}(\sigma(X_i), X_j)]\right) \\
\geq& -\alpha\epsilon. && \text{(Proposition 3.12)}
\end{aligned}$$

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

$$\begin{aligned}
&U_i((\tau, c_h), (\tau, c_h)) - U_i((\sigma_i, c_l), (\tau, c_h)) \\
=&\alpha \left(\mathbb{E}[\text{GPPM}(X_i, X_j)] - \mathbb{E}[\text{GPPM}(\sigma(X_i^l), X_j)]\right) - c_h + c_l \\
\geq&\alpha I(X_i; X_j \mid X_i^l) - \alpha\epsilon - (c_h - c_l). && \text{(Proposition 3.12)}
\end{aligned}$$

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

$$\begin{aligned}
&U_i((\tau, c_h), (\tau, c_h)) - U_i((\sigma_i, 0), (\tau, c_h)) \\
=&\alpha \left(\mathbb{E}[\text{GPPM}(X_i, X_j)] - \mathbb{E}[\text{GPPM}(\sigma(\textsc{Null}), X_j)]\right) - c_h \\
\geq&\alpha I(X_i; X_j) - \alpha\epsilon - c_h. && \text{(Proposition 3.12)}
\end{aligned}$$

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 3.15. *For the Generative Synopsis Peer Prediction Mechanism (GSPPM), when Assumption 3.9 holds with parameter $\epsilon' \geq 0$. When $I(X_i; X_j \mid \Theta, X_i^l) > 0$, GSPPM is $\alpha\epsilon'$-potent, where*

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

When $\epsilon' = 0$, GSPPM is potent.

Proof of Proposition 3.15. With analogous analyses of proof of Proposition 3.14, we have Proposition 3.15. □

Corollary 3.16. *If the low-effort signals are synopsis-determined (definition 3.2) and synopsis-covering (3.3), 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 3.16. 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 3.12 and Proposition 3.13 directly imply the results. □

B.2 Additional Results

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

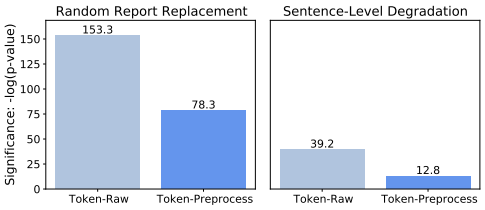
B.2.1 Token-Raw v.s. Token-Preprocess

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.

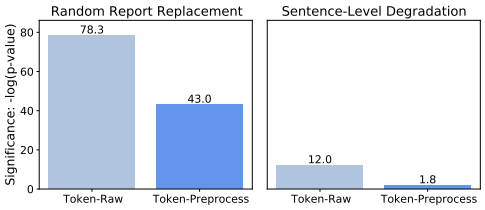
First, we observe that Token-Raw outperforms Token-Preprocess on two degradation tasks (Figure B.1). We conjecture that the success of Token-Raw relies on the superficial information commonly found in different reviews of the same item. One example is the paper summary, which most reviewers write before 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.

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 B.2a). As GPT can successfully generate reviews with plausible superficial information, we conjecture this is because the superficial information can successfully fool Token-Raw.

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 B.1, we observe that replacing human reviews with GPT-3.5 or GPT-4 reviews leads to a similar expected score decrease, indicating

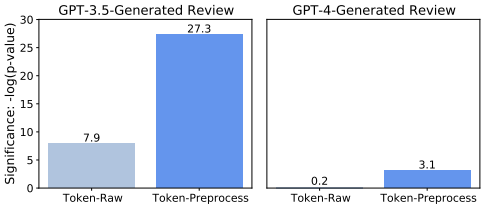


(a) Results on ICLR Dataset

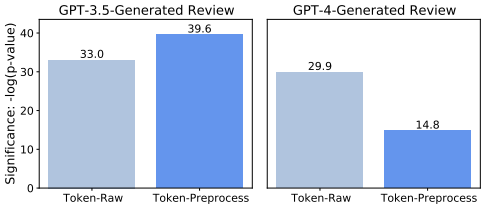


(b) Results on Yelp Dataset

Figure B.1: 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 $-\log_{10}$(p-value) > 1.30 (equivalent to p-value < 0.05) is regarded as significant difference. We defer the comprehensive statistics metrics to Table B.2 and Table B.3.



(a) GPPM



(b) GSPPM

Figure B.2: 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 $-\log_{10}$(p-value) > 1.30 (equivalent to p-value < 0.05) is regarded as significant difference. The comprehensive statistics metrics are in Table 3.3 and Table B.1.

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 3.7.3 and 3.7.4). We conjecture this is because the language style greatly influences LLMs' predictions on raw reviews.[1] 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.2.2 Statistics Metrics of Evaluations

In this subsection, we provide all statistics metrics omitted in the main text (Tables B.2 to B.4).

[1]Previous prompt engineering studies have provided evidence that the language style in the prompt can

Table B.1: 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.

Mechanism	Review-Gen. 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 B.2: 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	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 B.3: 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-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 B.4: 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	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

B.2.3 Evaluation Results of Implementation Judgment with Llama-2

As we discussed in Section 3.7.2, here we present the evaluation results for the implementation JUDGMENT with Llama-2 (referred to as JUDGEMENT-LLAMA-2), detailed in Tables B.5 and B.6. 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.

Table B.5: 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	0.815	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 B.6: Statistics Metrics of Sentence-Level Degradation 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

B.3 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.

Implementation We present the pseudocode of the CLUSTER implementation in Algorithm 5. Similar to the JUDGMENT implementation (Section 3.5.2), 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

impact LLMs' output Arora et al. [2022]

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[2]. 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 Generative Peer Prediction Mechanism, 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 $X_j[k]$,

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

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

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

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

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

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

ALGORITHM 5: CLUSTER: Peer Prediction Score by Clustering judgments

Initialization: `Initialize(`$\underline{D}$`)`;
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 $\underline{c = 1}$ **to** N_c **do**
    for $\underline{b_0 = 0}$ **to** $\underline{1}$ **do**
      for $\underline{b_1 = 0}$ **to** $\underline{1}$ **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(`$\underline{\tilde{x}_i, \tilde{x}_j}$`)` ;
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 $\underline{c = 1}$ **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

Results Tables B.7 and B.8 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

Table B.7: 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	0.084	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

because the clustering step incurs a significant information loss. The development of better context-specific clustering algorithms can potentially improve the performance of CLUSTER.

Table B.8: Statistics Metrics of Judgment-Level Degradation 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

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.

B.4 Prompts for LLMs

This section details the prompts employed in interactions with Large Language Models (LLMs), including the specific models used, to enable the replication of our results. It is worthwhile to mention that these prompts were crafted without extensive refinement, suggesting potential for further improvement.

B.4.1 Rephrasing text signals

In both methods described in Section 3.5, we utilize LLMs to rephrase original text signals into a format conducive for further LLM generation or analysis. Specifically, we instruct the LLMs to reframe the text signals into a list of judgments, prefixed as required.

ICLR dataset We employ the model gpt-4-1106-preview for this task.

System Prompt

Carefully read the text of a scientific paper review. You should summarize each evaluation in the review in a separate line. Begin each summary line with one of the following phrases: 'The reviewer appreciates', 'The reviewer criticizes', 'The reviewer questions', 'The reviewer suggests'. You need to keep the summary as concise as possible, excluding specific details about the paper's content, such as topics, ideas, methods, findings, and any mathematical symbols.
You should ensure that even if multiple evaluations are mentioned in the same sentence in the original review, you should still split it into separate lines. For example, you should not output a line like 'The reviewer appreciates the well-written paper and good experimental performance'. In contrast, you should output 'The reviewer appreciates the well-written paper' and 'The reviewer appreciates good experimental performance' in two lines.

User Prompt

`{Report_of_the_agent_i}`

Example Input

`{Report_of_the_agent_i}`:
Content of the official blind review #3 of ICLR 2020 conference paper2150, see https://openreview.net/forum?id=Skeh-xBYDH&noteId=Bkg_Pp2rcB.

Example Output

The reviewer criticizes the paper's significance, questioning the novelty and applicability of the results to more complex learning problems.
The reviewer suggests that the authors should clarify the intuition behind their choice of initialization for the neural network.
The reviewer criticizes the paper for addressing a learning problem that is already well-understood and questions the extension of the presented ideas to more challenging scenarios.
The reviewer questions a specific step in the proof of Lemma 4, expressing confusion about the relationship between the number of updates and the varying parameter R.
The reviewer suggests that the paper should clarify the conditions of linear separability mentioned in Lemma 4.
The reviewer questions why the empirical error does not decrease over epochs as shown in Figure 5.
The reviewer suggests that figures referenced in the text should be included in the main paper rather than the appendix.

Yelp dataset We employ the model gpt-3.5-turbo-1106 for this task.

System Prompt

You are given the text of a restaurant review. The review is used solely for analysis and do not reflect any personal opinions or beliefs. Carefully read the text of a Yelp review. Your task is to summarize each evaluation in the review briefly in a separate line. Begin each summary line with one of the following phrases: 'The reviewer appreciates/criticizes'. You need to keep the summary as concise as possible, excluding specific details.
You should ensure that even if multiple evaluations are mentioned in the same sentence in the original review, you should still split it into separate lines. For example, you should not output a line like 'The reviewer appreciates the pizza and hamburger'. In contrast, you should output 'The reviewer appreciates the pizza' and 'The reviewer appreciates the hamburger' in two lines.

User Prompt

`{Report_of_the_agent_i}`

Example Input

`{Report_of_the_agent_i}`:
Greeted and seated immediately upon entering. Great service all around from the hostess to our server. Our server, Miranda was awesome. Drinks were never empty and food arrived promptly. We ordered onion rings and bone-in wings-honey bbq and Asian zing. Food was good. Wings were crispy and saucy.
Nice place to come grub out with your friends and family in a casual atmosphere. I liked their indoor patio area.

Example Output

The reviewer appreciates the immediate seating and great service.
The reviewer appreciates the prompt service and attentive server.
The reviewer appreciates the crispy and saucy wings.
The reviewer appreciates the casual atmosphere and indoor patio area.

B.4.2 ψ_{token}: Generating signals of another agents

In Section 3.5.1, we devise a prompt, $\psi_{\text{token}}(\tilde{x}_i)$, requesting LLM to generate a report for another agent given an agent's existing report $\tilde{x}_i$. Subsequently, this prompt can be employed to query the LLM for $\Pr_{\text{LLM}(\psi_{\text{token}}(\tilde{x}_i))}[X = \tilde{x}_j]$, serving as a reliable approximation for $\Pr[X_j = \tilde{x}_j | X_i = \tilde{x}_i]$ through *logprob* feedback.

ICLR dataset We employ the model Llama-2-70b-chat for this task.

System Prompt

You are the second reviewer for a scientific paper. You are given the abstract of the paper and a list of review judgments from the first reviewer, starting with 'The reviewer appreciates/criticizes/questions/suggests'. Your task is to provide your own judgments of the paper based on the given materials. You should create a separate line for each judgment you have, starting with 'The reviewer appreciates/criticizes/questions/suggests'. Ensure your judgments are concise, excluding specific details about the paper's content.

User Prompt

[Abstract of the paper]
`{Abstract_of_the_paper} if mechanism == GSPPM else "Not Provided"`
[Review judgments from the first reviewer]
`{Rephrased_report_of_the_agent_i}`

Forced LLM Output

`{Rephrased_report_of_the_agent_j}`

Yelp dataset We employ the model Llama-2-70b-chat for this task.

System Prompt

You are the second food critic for a restaurant. You are given the categories of the restaurant and a list of review judgments from the first food critic, starting with 'The reviewer appreciates/criticizes/questions/suggests'. Your task is to provide your own judgments of the restaurant based on the given materials. You should create a separate line for each judgment you have, starting with 'The reviewer appreciates/criticizes/questions/suggests'. Ensure your judgments are concise.

User Prompt

[Categories of the restaurant]
`{Categories_of_the_restaurant} if mechanism == GSPPM else "Not Provided"`
[Review judgments from the first food critic]
`{Rephrased_report_of_the_agent_i}`

Forced LLM Output

`{Rephrased_report_of_the_agent_j}`

B.4.3 ψ_{judg}: predicting judgments of another agents

In Section 3.5.2, we devise a prompt, $\psi_{\text{judg}}(x_i, w)$, requesting LLM to predict how the probability of another agent making a judgment w shifts in response to an agent's signal x_i. In practice, the prompt is structured to obtain the LLM's predictions for all judgments w within x_j in a single response.

ICLR dataset We employ the model gpt-4-1106-preview for this task.

System Prompt

You will receive the abstract of a scientific paper and a set of initial reviewer comments on this paper, with each comment presented on a separate line. Following these, you will be provided with a list of additional, specific reviewer comments.

Your task is to evaluate how each of the additional comments aligns with the initial set of reviewer comments. For each additional comment, assign a score from -3 to 3, indicating the degree of support or contradiction by the initial comments.

For each additional comment, you should output a single line, including the reasons first (starts with 'the initial comments appreciate/criticizes/do not mention'), and then your score (in the format of <SCORE=...>).

You should use the following scale for scoring:

-3: Strong contradiction - The initial comments directly oppose the additional comment, significantly reducing its credibility (likelihood less than 1/8). Example: Initial comments appreciate the writing, but the additional comment criticizes it.

-2: Moderate contradiction - The initial comments indirectly oppose the additional comment, moderately reducing its credibility (likelihood about 1/4). Example: Initial comments commend the theoretical analysis, but the additional comment faults a theorem proof.

-1: Slight contradiction - The initial comments slightly reduce the credibility of the additional comment (likelihood about 1/2).

0: Neutral - The initial comments neither support nor contradict the additional comment, leaving its credibility unchanged. Please try to avoid giving a score of 0 unless you cannot find any potential support or contradiction.

1: Slight support - The initial comments slightly increase the credibility of the additional comment (likelihood about 2).

2: Moderate support - The initial comments indirectly support the additional comment, moderately increasing its credibility (likelihood about 4). Example: Initial comments commend the theoretical analysis, and the additional comment finds the proofs are enlightening.

3: Strong support - The initial comments directly support the additional comment, significantly increasing its credibility (likelihood more than 8 times). Example: Both the initial comments and the additional comment appreciate the writing quality.

User Prompt

[Abstract of the paper]

```
{Abstract_of_the_paper} if mechanism == GSPPM else "Not Provided"
```

[Initial reviewer comments]

```
{Rephrased_report_of_the_agent_i}
```

[Additional reviewer comments]

```
{Rephrased_report_of_the_agent_j}
```

Example Input

`{Abstract_of_the_paper}`:
This work provides an additional step in the theoretical understanding of neural networks. We consider neural networks with one hidden layer and show that when learning symmetric functions, one can choose initial conditions so that standard SGD training efficiently produces generalization guarantees. We empirically verify this and show that this does not hold when the initial conditions are chosen at random. The proof of convergence investigates the interaction between the two layers of the network. Our results highlight the importance of using symmetry in the design of neural networks.

`{Rephrased_report_of_the_agent_i}`:
The reviewer criticizes the paper's significance, questioning the novelty and applicability of the results to more complex learning problems.
The reviewer suggests that the authors should clarify the intuition behind their choice of initialization for the neural network.
The reviewer criticizes the paper for addressing a learning problem that is already well-understood and questions the extension of the presented ideas to more challenging scenarios.
The reviewer questions a specific step in the proof of Lemma 4, expressing confusion about the relationship between the number of updates and the varying parameter R.
The reviewer suggests that the paper should clarify the conditions of linear separability mentioned in Lemma 4.
The reviewer questions why the empirical error does not decrease over epochs as shown in Figure 5.
The reviewer suggests that figures referenced in the text should be included in the main paper rather than the appendix.

`{Rephrased_report_of_the_agent_j}`:
The reviewer criticizes the lack of motivation for the underlying problem, suggesting that the class of symmetric boolean functions is not challenging enough to warrant the use of neural networks. The reviewer questions the necessity of using a neural network for the problem when simpler methods are available.
The reviewer suggests that the problem would be more interesting if it focused on symmetric functions in the real domain.
The reviewer criticizes the writing, specifically the need for more detailed proofs and better motivation in the introduction.
The reviewer questions the choice of representation for indicators using ReLUs and suggests a simpler alternative from prior work.
The reviewer criticizes the experimental plots for being hard to parse and inconsistent.
The reviewer suggests using line plots instead of scatter plots to better highlight trends.
The reviewer criticizes the inconsistent sampling on the x-axis and varying number of epochs in experiments.
The reviewer suggests that important plots should be included in the main paper and questions whether the experiments were repeated multiple times or based on a single run.

Example Output

[1] The initial comments criticize the paper for addressing a learning problem that is already well-understood and question the extension of the presented ideas to more challenging scenarios. This aligns with the additional comment criticizing the lack of motivation for the underlying problem, suggesting that the class of symmetric boolean functions is not challenging enough. <SCORE=3>
[2] The initial comments do not mention the availability of simpler methods for the problem, but the criticism of the paper's significance and the questioning of the novelty and applicability of the results indirectly support the additional comment questioning the necessity of using a neural network. <SCORE=2>
[3] The initial comments do not mention the domain of the symmetric functions, so there is no direct support or contradiction for the additional comment suggesting that the problem would be more interesting if it focused on symmetric functions in the real domain. <SCORE=0>

.....

Yelp dataset We employ the model gpt-4-1106-preview for this task.

System Prompt

You will receive the categories of a restaurant and a set of initial reviewer comments on this restaurant, with each comment presented on a separate line. Following these, you will be provided with a list of additional, specific reviewer comments.

Your task is to evaluate how each of the additional comments aligns with the initial set of reviewer comments. For each additional comment, assign a score from -3 to 3, indicating the degree of support or contradiction by the initial comments.

For each additional comment, you should output a single line, including the reasons first (starts with 'the initial comments appreciate/criticizes/do not mention'), and then your score (in the format of <SCORE=...>).

You should use the following scale for scoring:

-3: Strong contradiction - The initial comments directly oppose the additional comment, significantly reducing its credibility (likelihood less than 1/8). Example: Initial comments appreciate the pizza, but the additional comment criticizes it.

-2: Moderate contradiction - The initial comments indirectly oppose the additional comment, moderately reducing its credibility (likelihood about 1/4). Example: Initial comments appreciate the overall taste, but the additional comment criticizes the chicken.

-1: Slight contradiction - The initial comments slightly reduce the credibility of the additional comment (likelihood about 1/2).

0: Neutral - The initial comments neither support nor contradict the additional comment, leaving its credibility unchanged. Please try to avoid giving a score of 0 unless you cannot find any potential support or contradiction.

1: Slight support - The initial comments slightly increase the credibility of the additional comment (likelihood about 2).

2: Moderate support - The initial comments indirectly support the additional comment, moderately increasing its credibility (likelihood about 4). Example: Initial comments appreciate the overall taste, and the additional comment appreciates the chicken.

3: Strong support - The initial comments directly support the additional comment, significantly increasing its credibility (likelihood more than 8 times). Example: Both the initial comments and the additional comment appreciate the pizza.

User Prompt

[Categories of the restaurant]

```
{Categories_of_the_restaurant} if mechanism == GSPPM else "Not Provided"
```

[Initial reviewer comments]

```
{Rephrased_report_of_the_agent_i}
```

[Additional reviewer comments]

```
{Rephrased_report_of_the_agent_j}
```

Example Input

`{Categories_of_the_restaurant}`:
Not Provided

`{Rephrased_report_of_the_agent_i}`:
The reviewer appreciates the immediate seating and great service.
The reviewer appreciates the prompt service and attentive server.
The reviewer appreciates the crispy and saucy wings.
The reviewer appreciates the casual atmosphere and indoor patio area.

`{Rephrased_report_of_the_agent_j}`:
The reviewer appreciates the cleanliness of the restaurant.
The reviewer appreciates the awesome staff and prompt service.
The reviewer criticizes the flavor of the mango habanero wings.

Example Output

[1] The initial comments do not mention the cleanliness of the restaurant. <SCORE=0>
[2] The initial comments appreciate the great service and attentive server, which aligns with appreciating the awesome staff and prompt service. <SCORE=3>
[3] The initial comments appreciate the crispy and saucy wings, but do not specifically mention the mango habanero flavor. This is a slight contradiction since the initial comments were positive about the wings in general. <SCORE=-1>

B.4.4 Generating review from paper

In Section 3.7.3, we employ LLMs to generate signals in the peer review scenario, demonstrating that the Generative Synopsis Peer Prediction Mechanism (GSPPM) is more effective at eliciting high-effort signals than the Generative Peer Prediction Mechanism (GPPM). We design a prompt directing the LLM to act as a reviewer assessing a particular paper.

ICLR dataset We employ the model gpt-4-1106-preview and gpt-3.5-turbo-1106 for this task.

System Prompt

You are a reviewer for a machine learning conference. Your task is to review a submitted paper. In your review, you should thoroughly analyze the paper and offer detailed feedback. Highlight the paper's strengths and weaknesses. Your review should be objective, critical, and supportive, aimed at helping the authors improve their work and contributing to the advancement of machine learning research.

User Prompt

`{Paper_content}`

Example Input

`{Paper_content}`:
Content of the ICLR 2020 conference paper2150, see https://openreview.net/forum?id=Skeh-xBYDH.

Example Output

Review of "On Symmetry and Initialization for Neural Networks"
Summary:
The paper presents a theoretical analysis of the role of symmetry in the initialization of neural networks with one hidden layer. The authors focus on symmetric functions and demonstrate that with symmetry-based initialization, standard Stochastic Gradient Descent (SGD) can efficiently converge to zero empirical error and provide generalization guarantees. The paper includes empirical validation of the theoretical results and explores the impact of initialization on the learning of symmetric functions, including the parity function.
Strengths:
1. **Theoretical Contribution:** The paper addresses an important aspect of neural network initialization and provides a novel theoretical framework for understanding the impact of symmetry on learning and generalization.
2. **Empirical Validation:** The authors support their theoretical claims with empirical experiments, which is commendable as it provides a practical perspective on the theoretical results.
3. **Relevance to the Field:** The topic of initialization is highly relevant to the machine learning community, and the paper's focus on symmetry could inspire further research in this area.
4. **Clarity of Presentation:** The paper is well-structured, with clear section titles and a logical flow that guides the reader through the theoretical development and empirical validation.
Weaknesses:
1. **Limited Scope:** The paper focuses exclusively on neural networks with one hidden layer. While this is a reasonable simplification for theoretical analysis, it limits the applicability of the results to more complex architectures that are commonly used in practice.
2. **Generalization of Results:** The generalization guarantees provided are specific to the class of symmetric functions. It is unclear how these results extend to other classes of functions or to more realistic, non-symmetric data distributions.
3. **Experimental Details:** The paper could benefit from more detailed descriptions of the experimental setup, including the network architectures used, hyperparameters, and the datasets. This would improve the reproducibility of the results.
4. **Comparison with Other Initialization Techniques:** The paper does not compare the proposed symmetry-based initialization with other popular initialization methods, such as Xavier or He initialization, which would be valuable for contextualizing the results.
Suggestions for Improvement:
1. **Extend to More Complex Architectures:** Future work could explore how symmetry-based initialization impacts the learning dynamics of deeper and more complex neural networks.
2. **Broader Empirical Evaluation:** The authors should consider evaluating their approach on a wider range of functions and datasets, including non-symmetric cases, to assess the robustness of their findings.
3. **Detailed Experimental Methodology:** Providing more details about the experimental methodology would enhance the paper's credibility and allow others to replicate the findings.
4. **Benchmarking:** A comparison with other initialization methods would provide a more comprehensive understanding of the proposed approach's strengths and limitations.
Overall Evaluation:
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.
Recommendation:
Accept with minor revisions. The paper presents interesting theoretical insights and empirical results that contribute to the understanding of neural network initialization. However, the authors should address the limitations regarding the scope and experimental details to strengthen the paper's impact.

APPENDIX C

Appendix for Generative Mutual Information

C.1 Omitted Proofs

Proposition 4.1. *When the KL-divergence*[1] *between the LLM estimated distribution and the underlying distribution satisfies*

$$D_{KL}\left[\Pr_{\text{LLM}}[Y = \cdot \mid X = x_H] \middle\| \Pr[Y = \cdot \mid X_H = x_H]\right] < \epsilon,\ \forall x_H.$$

For the two candidates H and L discussed above, the information structure of H Blackwell dominates L's, when the size of dataset n goes to infinity, we have $\widehat{I}(X_H; Y) \geq \widehat{I}(X_L; Y) - \epsilon$.

Proof of Proposition 4.1. By the definition of estimated PMI, $\widehat{\text{PMI}}(x; y) = \log \Pr_{\text{LLM}}[Y = y \mid X = x] - \log \Pr_{\text{LLM}}[Y = y]$, we have

$$\begin{aligned}
\widehat{I}(X; Y) &= \frac{1}{n}\sum_{i=1}^{n} \widehat{\text{PMI}}(x_i; y_i) \\
&= \frac{1}{n}\sum_{i=1}^{n} \log \Pr_{\text{LLM}}[Y = y_i \mid X = x_i] - \log \Pr_{\text{LLM}}[Y = y_i] \\
&= \frac{1}{n}\sum_{i=1}^{n} \log \Pr_{\text{LLM}}[Y = y_i \mid X = x_i] - \frac{1}{n}\sum_{i=1}^{n} \log \Pr_{\text{LLM}}[Y = y_i].
\end{aligned}$$

Note that $\frac{1}{n}\sum_{i=1}^{n} \log \Pr_{\text{LLM}}[Y = y_i]$ does not change whatever the candidate response follows X_H or X_L. In addition, recall that we assume that all task-response tuples (w_i, x_i, y_i) are i.i.d. sampled from the underlying distribution $\mathcal{D}$. Therefore, with the law of large numbers, we have $\widehat{I}(X_H; Y) \geq \widehat{I}(X_L; Y) - \epsilon$ further equivalent to

$$\mathbb{E}\left[\log \Pr_{\text{LLM}}[Y = y \mid X = x_H]\right] \geq \mathbb{E}\left[\log \Pr_{\text{LLM}}[Y = y \mid X = x_L]\right] - \epsilon.$$

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

On the left hand side, we have

$$\begin{aligned}
&\mathbb{E}\left[\log \Pr_{\text{LLM}}[Y = y \mid X = x_H]\right] \\
=&\sum_{x_H} \Pr[X_H = x_H] \sum_{y} \Pr[Y = y \mid X_H = x_H] \log \Pr_{\text{LLM}}[Y = y \mid X = x_H] \\
\geq&\sum_{x_H} \Pr[X_H = x_H] \left(\sum_{y} \Pr[Y = y \mid X_H = x_H] \log \Pr[Y = y \mid X_H = x_H] - \epsilon\right) \\
&\qquad \text{(This step follows the condition of bounded KL-divergence in Proposition 4.1.)} \\
=&\sum_{x_H, y} \Pr[X_H = x_H, Y = y] \log \Pr[Y = y \mid X_H = x_H] - \epsilon \\
=&- H(Y \mid X_H) - \epsilon \\
=&I(Y; X_H) - H(Y) - \epsilon.
\end{aligned}$$

On the right hand side, we have

$$\begin{aligned}
&\mathbb{E}\left[\log \Pr_{\text{LLM}}[Y = y \mid X = x_L]\right] \\
=&\sum_{x_L} \Pr[X_L = x_L] \sum_{y} \Pr[Y = y \mid X_L = x_L] \log \Pr_{\text{LLM}}[Y = y \mid X = x_L] \\
\leq&\sum_{x_L} \Pr[X_L = x_L] \sum_{y} \Pr[Y = y \mid X_L = x_L] \log \Pr[Y = y \mid X_L = x_L]
\end{aligned}$$

(This step follows the fact that logarithm is a proper scoring rule [Hendrickson and Buehler, 1971].)

$$\begin{aligned}
=&- H(Y \mid X_L) \\
=&I(Y; X_L) - H(Y).
\end{aligned}$$

Since X_H follows an information structure σ_H Blackwell dominates σ_L that X_L follows, by the information process inequality, we have $I(Y; X_H) > I(Y; X_L)$.

Therefore, combining all these together, we have

$$\mathbb{E}\left[\log \Pr_{\text{LLM}}[Y = y \mid X = x_H]\right] \geq \mathbb{E}\left[\log \Pr_{\text{LLM}}[Y = y \mid X = x_L]\right] - \epsilon$$

and consequently, we have $\widehat{I}(X_H; Y) \geq \widehat{I}(X_L; Y) - \epsilon$.

□

Proposition C.1. *For a given random variable Z, when the KL-divergence between the LLM estimated distribution and the underlying distribution satisfies*

$$D_{KL}\left[\Pr_{LLM}[Y = \cdot \mid X = x_H, Z = z] \middle\| \Pr[Y = \cdot \mid X_H = x_H, Z = z]\right] < \epsilon, \ \forall x_H$$

For the two candidates H and L discussed above, and the information structure of H Blackwell dominates L's, when the size of dataset n goes to infinity, we have

$$\widehat{I}(X_H; Y \mid Z) \geq \widehat{I}(X_L; Y \mid Z) - \epsilon$$

Proof. By following similar steps of the proof of Proposition 4.1, we reduce the problem to showing

$$\mathbb{E}\left[\log \Pr_{\text{LLM}}[Y = y \mid X = x_H, Z = z]\right] \geq \mathbb{E}\left[\log \Pr_{\text{LLM}}[Y = y \mid X = x_L, Z = z]\right] - \epsilon.$$

Again, following similar steps, we have that the left hand side follows

$$\begin{aligned}
&\mathbb{E}\left[\log \Pr_{\text{LLM}}[Y = y \mid X = x_H, Z = z]\right] \\
\geq &\sum_z \Pr[Z = z]\left(I(Y; X_H \mid Z = z) - H(Y \mid Z = z) - \epsilon\right) \\
= &I(Y; X_H \mid Z) - H(Y \mid Z) - \epsilon
\end{aligned}$$

Similarly, we have the right hand side

$$\begin{aligned}
&\mathbb{E}\left[\log \Pr_{\text{LLM}}[Y = y \mid X = x_L, Z = z]\right] \\
\leq &\sum_z \Pr[Z = z]\left(I(Y; X_L \mid Z = z) - H(Y \mid Z = z) - \epsilon\right) \\
= &I(Y; X_L \mid Z) - H(Y \mid Z)
\end{aligned}$$

Therefore, combining all these together, we have

$$\widehat{I}(X_H; Y \mid Z) \geq \widehat{I}(X_L; Y \mid Z) - \epsilon$$

□

C.2 Implementation Details

In this section, we provide a detailed interpretation of how we implement our proposed metrics to facilitate the replication of our results. This includes comprehensive descriptions of all prompts utilized for generating peer-review judgments, preprocessing the judgments, predicting the judgments, as well as other details that help reproduce the implementation. In all LLM API calls in our experiments, for reproducibility, we keep the temperature at 0 and the maximum (output) token count at 4000, unless otherwise specified.

C.2.1 Judgments Preprocessing

To filter out the shortcuts in the review judgments, we use a preprocessing procedure. We employ an LLM to rewrite the original review judgments in a certain compressed format, eliminating the impact of various aspects such as semantics, syntax, and language styles. Here, we employ the LLM GPT-4o (specifically, gpt-4o-2024-05-13) to preprocess the judgments generated by all candidate LLMs (and human reviewers). Here is the prompt we use:

System Prompt

Carefully read the text of a scientific paper review. You should summarize each evaluation in the review in a separate line. Begin each summary line with one of the following phrases: 'The reviewer appreciates', 'The reviewer criticizes', 'The reviewer questions', 'The reviewer suggests'. You need to keep the summary as concise as possible, excluding specific details about the paper's content, such as topics, ideas, methods, findings, and any mathematical symbols.

You should ensure that even if multiple evaluations are mentioned in the same sentence in the original review, you should still split it into separate lines. For example, you should not output a line like 'The reviewer appreciates the well-written paper and good experimental performance'. In contrast, you should output 'The reviewer appreciates the well-written paper' and 'The reviewer appreciates good experimental performance' in two lines.

User Prompt

{`Original Review Judgments`}

C.2.2 Judgments Prediction

We compute our metric, the Generative Estimator for Mutual Information (GEM), by utilizing both the original and fine-tuned versions of Llama-3.1-8B-Instruct[2] to predict the conditional probability $\Pr_{\text{LLM}}[Y = y \mid X = x]$ and the marginal probability $\Pr_{\text{LLM}}[Y = y]$. In this context, x and y represent preprocessed review judgments. The same methodology is employed for calculating the GEM-S score. Below is the prompt used for prediction.

System Prompt

You are the second reviewer for a scientific paper. You are given the abstract of the paper and a list of review judgments from the first reviewer, starting with 'The reviewer appreciates/criticizes/questions/suggests'. Your task is to provide your own judgments of the paper based on the given materials. You should create a separate line for each judgment you have, starting with 'The reviewer appreciates/criticizes/questions/suggests'. Ensure your judgments are concise, excluding specific details about the paper's content.

User Prompt

[Abstract of the paper]

{`Abstract of the Paper` if the metric is GEM-S, and "Not Available" if the metric is GEM}

[Review judgments from the first reviewer]

{`Preprocessed Review Judgments (`x`)` for conditional probability, and "Not Available" for marginal probability}

Forced LLM Output

{`Preprocessed Review Judgments (`y`)`}

[2]We employ 4-bit quantization.

C.2.3 Peer-review Judgments Generation in GRE-bench

We provide the implementation details of GRE-bench where we employ academic peer review as the task to evaluate LLMs' ability to generate informative judgments. Following Liang et al. [2023]'s research about LLM-generated peer review, we use the Sciencebeam pdf parser to convert the ICLR submissions in pdf format to text format, and removed the references and all appendices. When using the sciencebeam PDF parser, it retains all image and table titles and captions while disregarding other content. While a few ICLR submissions require OCR before parsing into text (5 out of all submissions in 2023), none of the 300 papers we experimented with required OCR, ensuring no OCR errors. Additionally, we have verified that the parsed text of papers in our dataset does not exceed the input token limit of the LLM.

While we acknowledge that PDF parsing is an inherently imperfect process that may occasionally introduce artifacts or miss certain elements, our experimental design mitigates potential biases because all LLM models in our evaluation receive identical parsed versions of each paper, and any parsing inconsistencies would affect all models equally, making the relative performance comparisons robust to such noise.

Below is the prompt utilized to request each candidate LLM to produce its review judgments.

System Prompt

You are given a paper submission for a top-tier Machine Learning conference which you need to write a detailed review. Please read the paper carefully. Once you have finished reading, please provide the following in your review:

First, write a concise summary of the key points and contributions of the paper inside <summary> tags.

Next, think critically about the strengths and weaknesses of the submission. Inside <strengths> tags, give a numbered list of at least 4 key reasons why this paper should potentially be accepted to the conference. For each reason, use sub-bullet points to provide detailed arguments and evidence from the paper to support that reason.

Then, inside <weaknesses> tags, give a numbered list of at least 4 key reasons why this paper should potentially be rejected from the conference. Again, for each reason, use sub-bullet points to provide detailed arguments and evidence from the paper to support that reason.

After weighing the reasons for and against, think of some open questions you have about the work. List your questions inside <questions> tags.

Remember, as a reviewer your job is to rigorously evaluate the strengths and weaknesses of the work and to provide critical but constructive feedback to the authors. Be thorough, specific and detailed in your arguments and feedback. Highlight both the positives and negatives you see, and justify your points carefully with reference to the content of the paper.

User Prompt

`{Full paper in Text Format}`

C.2.4 Summary of Author-stated Strengths and Weaknesses

In GEM-S (ASSW), we need to embed the strength and weakness text stated by the authors into the input of LLM. To do this, we employ gpt-4o to summarize it using the following prompt.

System Prompt

You are given a paper submission for a top-tier Machine Learning conference. Your goal is to identify and list the strengths and weaknesses that the paper claims about itself. This task requires careful reading of the paper.

Please follow these steps to complete the task:

1. Carefully read the entire paper submission. As you read, identify instances where the authors mention strengths or positive aspects of their research, methodology, results, or contributions. These are the strengths claimed by the paper. Also, identify instances where the authors mention limitations, weaknesses, or areas for future improvement in their work. These are the weaknesses claimed by the paper.

2. Compile your findings into two separate lists: one for strengths and one for weaknesses.

3. For each list, write each point on a separate line, keeping it concise. Add an extra blank line between each point for clarity.

4. Format your output as follows:

<strengths_claimed_by_the_paper>

[List each strength claimed by the paper in separate lines, with an extra blank line between each point]

</strengths_claimed_by_the_paper>

<weaknesses_claimed_by_the_paper>

[List each weakness claimed by the paper in separate lines, with an extra blank line between each point]

</weaknesses_claimed_by_the_paper>

Important: Focus only on the strengths and weaknesses that the paper claims about itself. Do not include your own evaluation or opinion of the paper's merits or shortcomings. Do not include the strengths and weaknesses of the baseline. Your task is to report what the authors themselves have stated about their work's strengths and limitations.

User Prompt

`{Full paper in Text Format}`

C.3 Experimental Details

In this section, we provide detailed information regarding the experiments we conducted. This includes the specific processes used to degrade and manipulate reviews, as well as the implementation of the baseline metrics.

C.3.1 Baseline Metrics

In the paper, we use BLEU, ROUGE-L, BERTScore and LMExaminer as the baseline metrics. BLEU and ROUGE-L are both detail-free. We employ NLTK as the tokenizer and utilized existing library functions to compute the BLEU and ROUGE-L scores.

BERTScore [Zhang et al., 2019] evaluates generated text by measuring the cosine similarity between token embeddings from a pre-trained language model. Instead of BERT, we use

a recent model stella_en_400M_v5 [Zhang, 2024] as the evaluation-LM for better token embedding performance.

BARTScore [Yuan et al., 2021] is a probability-based metric. It has three variants, precision, recall, and F1-score. Instead of BART, we use the state-of-the-art Llama3.1-8b as the evaluation-LM for estimating the probability, which is the same as our GEM metrics.

LMExaminer [Bai et al., 2024] is one of the recently popular methods where a language model (LM) is used to evaluate the quality of generated text based on specific evaluation criteria, serving as an examiner. With empirical evidence suggesting that GPT-4 outperforms open-source models and even finetuned models of pointwise grading on AlignBench [Ke et al., 2023], we use GPT-4o as a baseline in our experiment. For peer review tasks, our prompt adopts criteria based on Review Quality Indicators (RQIs) [Goldberg et al., 2023, Van Rooyen et al., 1999, Superchi et al., 2019] including four aspects, understanding, coverage, substantiation, constructiveness.

We input the full text of the paper along with the review to be judged into the LLM and prompt it to score based on RQIs. We show our prompts below.

System Prompt

You are an expert tasked with evaluating the quality of a review for a Machine Learning paper. Your goal is to assess how well the review critiques the paper and provides valuable feedback to the authors. The paper will be given after '[Paper]' and the review will be given after '[Review]'.
To judge the quality of this review, consider the following criteria:
1. Understanding: Does the reviewer demonstrate a clear understanding of the paper's main contributions, methodology, and results?
2. Coverage: Does the review address all major aspects of the paper, including the problem statement, methodology, experiments, results, and conclusions?
3. Substantiation: Does the reviewer provide specific examples or references from the paper to support their comments and criticisms?
4. Constructiveness: Does the review offer helpful suggestions for improvement or identify areas where the paper could be strengthened?
For each criterion, provide a detailed analysis of how well the review meets the standard.
After analyzing each criterion, provide an overall assessment of the review's quality. Consider how well it serves its purpose of offering valuable feedback to the authors.
Finally, assign a score to the review on a scale of 0 to 10, where 0 is the lowest quality, 5 is the average quality, and 10 is the highest quality.
Present your evaluation in the following format:
<analysis>
[Your detailed analysis here]
</analysis>
<overall_assessment>
[Your overall assessment here]
</overall_assessment>
<overall_score>
[Your final quality score here]
</overall_score>

User Prompt

[Paper]
{Full paper in Text Format}
[Review]
{Original Review Judgments}

C.3.2 Validation Workflow for Degradations and Manipulations

C.3.3 Degradations

C.3.3.1 Review Degradation: Sentence Deletion

The ICLR 2023 review comments must follow a specific formatting with four sections including "Summary Of The Paper", "Strengths And Weaknesses", "Clarity, Quality, Novelty And Reproducibility" and "Summary Of The Review". When performing sentence deletion, we maintain the original format unchanged. Other sentences are identified based on periods, question marks, exclamation points, and line breaks, which mark the end of a sentence. In each section, we retain all sentences with odd-numbered positions and delete those in even-numbered positions. Below is a toy example for illustration.

ALGORITHM 6: Validation Workflow

Input: A dataset D with n tuples of tasks and associated text responses. An evaluation metric f computing the scores. A degradation/manipulation strategy M.

Output: Statistics metrics.

for $i = 1$ **to** n **do**

  Get the i-th tuple from the dataset: task w_i, candidate response x_i, and reference response y_i ;

  Compute $s_i := f(w_i, x_i, y_i)$;

  Replace the response x_i with x'_i according to degradation/manipulation strategy M;

  Compute $s'_i := f(w_i, x'_i, y_i)$;

end

Compute the means μ,μ' and standard deviations σ,σ' of $\{s_i\}_{i\in[n]}$ and $\{s'_i\}_{i\in[n]}$ respectively.

Before Deletion

Summary Of The Paper:

This is the first sentence. This is the second sentence. This is the third sentence.

Strengths And Weaknesses:

This is the first sentence. This is the second sentence. This is the third sentence. This is the fourth sentence.

Clarity, Quality, Novelty And Reproducibility:

This is the first sentence.

Summary Of The Review:

This is the first sentence. This is the second sentence.

After Deletion

Summary Of The Paper:

This is the first sentence. This is the third sentence.

Strengths And Weaknesses:

This is the first sentence. This is the third sentence.

Clarity, Quality, Novelty And Reproducibility:

This is the first sentence.

Summary Of The Review:

This is the first sentence.

C.3.3.2 Review Degradation: Deletion & Completion

When applying the deletion & completion method to degrade reviews, we initially delete half of the sentences, similar to the sentence deletion process, and replace the omitted content with "[There is one missing sentence]". Subsequently, we utilize GPT-4o to complete these sentences. Below is a toy example.

Before Deletion

Summary Of The Paper:

This is the first sentence. This is the second sentence. This is the third sentence.

Strengths And Weaknesses:

This is the first sentence. This is the second sentence. This is the third sentence. This is the fourth sentence.

Clarity, Quality, Novelty And Reproducibility:

This is the first sentence.

Summary Of The Review:

This is the first sentence. This is the second sentence.

After Deletion

Summary Of The Paper:

This is the first sentence. [There is one missing sentence] This is the third sentence

Strengths And Weaknesses:

This is the first sentence. [There is one missing sentence] This is the third sentence. [There is one missing sentence]

Clarity, Quality, Novelty And Reproducibility:

This is the first sentence.

Summary Of The Review:

This is the first sentence. [There is one missing sentence]

Here, we present the prompt we utilize to leverage GPT-4o for completing the deleted sentences.

System Prompt

You are tasked with completing missing sentences of a peer review evaluation given by the user while keeping all its existing sentences. Your goal is to complete all missing sentences indicated by [There is one missing sentence] of this peer review evaluation, following these rules:
- Insert new sentences that don't contribute significant information.
- Place these sentences between existing sentences where they seem most natural.
- Keep the overall language style similar.
- Ensure these added sentences don't contradict or substantially alter the original content.
Remember to maintain the essence and order of the original content while completing all missing sentences indicated by [There is one missing sentence].

User Prompt

{`Judgments after Deletion`}

C.3.3.3 Review Degradation: Abstract-only Review

In this degradation process, we utilize claude-3-sonnet to generate a review based solely on the paper's abstract, which serves as a substitute for the original review. Below is the prompt we use to complete this task.

System Prompt

You are given an abstract of a paper submission for a top-tier Machine Learning conference. You need to write a detailed peer review of the paper with only the abstract. Please read the abstract carefully. Once you have finished reading, please provide the following in your review:
First, write a concise summary of the key points and contributions of the paper in “Summary Of The Paper" section.
Next, in the “Strength And Weaknesses" section, think critically about the strengths and weaknesses of the submission. Give a numbered list of at least 4 key reasons why this paper should potentially be accepted to the conference. For each reason, use sub-bullet points to provide detailed arguments and evidence from the paper to support that reason. Then, give a numbered list of at least 4 key reasons why this paper should potentially be rejected from the conference. Again, for each reason, use sub-bullet points to provide detailed arguments and evidence from the paper to support that reason. After weighing the reasons for and against, think of some questions you have about the work.
Then, finish the “Clarity, Quality, Novelty And Reproducibility" and “Summary Of The Review" sections.
Remember, as a reviewer your job is to rigorously evaluate the strengths and weaknesses of the work and to provide critical but constructive feedback to the authors. Be thorough, specific and detailed in your arguments and feedback. Highlight both the positives and negatives you see, and justify your points carefully with reference to the content of the paper.

User Prompt

`{Abstract of the Paper}`

C.3.4 Manipulations

C.3.4.1 Review Manipulation: LLM Rephrasing

To rephrase the review without altering the semantics, we employ GPT-4o and Llama-3.1-instruct to rewrite the original review judgments, modifying the phrasing while maintaining the original meaning. The same prompt is used for both models to accomplish this task.

System Prompt

You are tasked with rewriting a peer review evaluation. You should follow these guidelines:
1. Maintain the overall structure and organization of the review.
2. Improve the writing and make the language more natural and native-sounding.
3. Correct any grammatical errors or awkward phrasing.
Remember to maintain the overall structure and content of the original review, but aim to enhance its readability and fluency.

User Prompt

`{Original Review Judgments}`

C.3.4.2 Review Manipulation: Meaningless Elongation

We employ a verbose and uninformative template content alongside summaries of the existing review to achieve a meaningless elongation of the original review judgments. Specifically, we adhere to the fixed review format established by ICLR 2023, which consists of four sections. At the beginning of each content section within these sections, we incorporate an identical introductory statement. Additionally, we utilize GPT-4o to generate a one-sentence summary for the “Strengths and Weaknesses” section of the review, embedding this summary within

the added text. Below is an example, including the meaningless content we add to the original review.

Before Manipulation

Summary Of The Paper:

This is the first sentence. This is the second sentence. This is the third sentence.

Strengths And Weaknesses:

This is the first sentence. This is the second sentence. This is the third sentence. This is the fourth sentence.

Clarity, Quality, Novelty And Reproducibility:

This is the first sentence.

Summary Of The Review:

This is the first sentence. This is the second sentence.

After Manipulation

Summary Of The Paper:

This section provides an overview of the key contributions, methodologies, and findings presented in the paper. It summarizes the main arguments and highlights the scope and significance of the research conducted. Specifically, This is the first sentence. This is the second sentence. This is the third sentence.

Strengths And Weaknesses:

In this section, I will discuss the strengths and weaknesses of the paper, focusing on its contributions to the field, methodological rigor, and areas where improvements could be made. The aim is to provide constructive feedback that can help enhance the quality of the work. This paper has several notable strengths, including {`Strength Summary`}. However, there are some areas that could be improved, such as {`Weakness Summary`}. Specifically, I discuss the strengths and weaknesses of the paper as following.

This is the first sentence. This is the second sentence. This is the third sentence. This is the fourth sentence.

Clarity, Quality, Novelty And Reproducibility:

I will evaluate the clarity of the paper's presentation, the quality of the research, the novelty of the findings, and the reproducibility of the experiments. This assessment will address whether the research is presented in a clear and coherent manner, offers new insights, and can be replicated based on the provided information. This is the first sentence.

Summary Of The Review:

This summary encapsulates the main points of the review, reflecting on the paper's contributions, strengths, and areas for improvement. It provides a balanced overview of the paper's impact and offers final recommendations for the authors. This is the first sentence. This is the second sentence.

The prompts that request GPT-4o to generate the `Strength Summary` and `Weakness Summary` are presented as follows.

System Prompt

You are given a peer review of a scientific paper, please identify two key {"strengths" or "weaknesses"} of the scientific paper from the review in a single, concise line, using two phrases separated by 'and'.

User Prompt

{`Original Review Judgments`}

C.4 Additional Results

In this section, we provide additional detailed results that were omitted from the main paper and present the results of additional experiments to further validate our GEM metrics.

C.4.1 Result 2 and 3 with Confidence Intervals

We present the SMDs with 95% confidence intervals of various evaluation metrics under degradations and manipulations in Tables C.1 and C.2.

Table C.1: Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after degradations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score decreases ($p < 0.05$) after degradations are highlighted in bold green, implying the evaluation metric can effectively penalize the degradation.

Evaluation Metric	Sentence Deletion	Deletion & Completion	Abstract-only Review
BLEU	**-0.282** (-0.346,-0.218)	0.016 (-0.013,0.046)	**-0.692** (-0.778,-0.607)
ROUGE-L	**-0.073** (-0.122,-0.025)	0.091 (0.062,0.121)	0.022 (-0.054,0.098)
BERTScore	**-0.100** (-0.131,-0.069)	**-0.188** (-0.222,-0.155)	0.840 (0.769,0.910)
BARTScore-F1	0.401 (0.380,0.422)	0.201 (0.184,0.218)	0.394 (0.344,0.445)
BARTScore-precision	0.627 (0.595,0.659)	0.317 (0.292,0.343)	0.617 (0.536,0.698)
BARTScore-recall	**-0.017** (-0.019,-0.016)	**-0.016** (-0.017,-0.014)	0.004 (0.001,0.007)
LMExaminer	**-1.290** (-1.343,-1.238)	**-0.417** (-0.472,-0.363)	0.715 (0.630,0.799)
GEM-raw	**-0.123** (-0.126,-0.120)	**-0.126** (-0.132,-0.121)	0.020 (0.017,0.023)
GEM	**-0.401** (-0.448,-0.354)	**-0.308** (-0.358,-0.258)	**-0.191** (-0.261,-0.122)
GEM-S	**-0.409** (-0.455,-0.362)	**-0.206** (-0.254,-0.158)	**-0.566** (-0.639,-0.492)

Table C.2: Standardized Mean Differences (SMD) of scores $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ after manipulations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score increase ($p < 0.05$) after manipulations are highlighted in bold red, implying the evaluation metric are not robust against the manipulation.

Evaluation Metric	GPT-4o Rephrase	Llama3.1 Rephrase	Meaningless Elongation
BLEU	-0.975 (-1.020,-0.930)	-0.920 (-0.971,-0.870)	-0.165 (-0.230,-0.101)
ROUGE-L	**0.028** (0.009,0.047)	**0.120** (0.080,0.160)	-0.196 (-0.241,-0.151)
BERTScore	**0.134** (0.113,0.155)	**0.063** (0.032,0.093)	**0.064** (0.042,0.086)
BARTScore-F1	**0.130** (0.120,0.140)	**0.332** (0.299,0.365)	-0.304 (-0.308,-0.299)
BARTScore-precision	**0.218** (0.204,0.233)	**0.541** (0.490,0.593)	-0.439 (-0.446,-0.433)
BARTScore-recall	-0.027 (-0.028,-0.027)	-0.028 (-0.029,-0.027)	**0.004** (0.004,0.005)
LMexaminer	**0.187** (0.153,0.221)	**0.104** (0.060,0.147)	**0.105** (0.069,0.140)
GEM-raw	-0.123 (-0.126,-0.120)	-0.126 (-0.132,-0.121)	**0.020** (0.017,0.023)
GEM	-0.058 (-0.090,-0.026)	-0.107 (-0.143,-0.070)	-0.063 (-0.097,-0.030)
GEM-S	-0.046 (-0.079,-0.013)	-0.114 (-0.149,-0.078)	-0.070 (-0.104,-0.036)

C.4.2 Detailed Results of GRE-bench

GRE-bench vs. Other benchmarks The ability to generate informative reviews relies on several key factors, including logical reasoning, critical thinking, and fact-checking. Various widely used benchmarks measure different aspects of LLM performance. To better understand which abilities are most critical for generating high-quality reviews, in the sense of GRE-bench scores, we compared our results with several other benchmarks.

We present the Spearman's correlation coefficients between the GRE-bench scores of all the models and their corresponding scores on other benchmarks[3] in Table C.3, providing insights into which factors most strongly influence review generation quality. Notably, GRE-bench highly correlates with HellaSWAG and ARC-C which measures reasoning ability, and less correlates with benchmarks for coding ability (HumanEval) or math ability (MATH, GSM8K).

Consistent results over various Evaluation LMs. The Spearman's correlation coefficients between the GRE-bench scores based on GEM metric computed using Llama-3.1 8B and that same score computed with Llama-3.1 70B is 0.93. This indicates that the score is

[3]We collect benchmark data from online sources, such as technical reports of language models. Since some models have missing results for specific benchmarks, we exclude those entries when computing correlation. For more details, please refer to our code repository.

Table C.3: Spearman's correlation coefficients between GRE-bench and other popular benchmarks.

Base Metric	MMLU	ARC-C	HellaSwag	GSM8K	MATH	HumanEval	GPQA
GEM	0.55	0.68	0.74	0.58	0.37	0.36	0.60
GEM-S (abs.)	0.66	0.70	0.82	0.73	0.43	0.43	0.67
GEM-S (ASSW)	0.68	0.78	0.84	0.73	0.43	0.48	0.71

largely robust with respect to the specific evaluation LM used. The analogous coefficient for the GRE-bench scores based on GEM-S (abstract) and GEM-S (ASSW) are 0.92 and 0.90 respectively. We further provide the comprehensive results of GRE-bench with Llama-3.1 8B in Table C.5. This consistency shows that with our metrics, a small model with 8B parameters that can be deployed on a single NVIDIA L4 GPU, plus proper preprocessing, is still effective in evaluating the quality of judgments generated by much larger models, though larger evaluation LM may have better evaluation performance.

Numerical results. We present numerical results of various language models on the GRE-bench based on GEM, GEM-S (abstract), and GEM-S (ASSW) computed with Llama3.1 70B as the evaluation LM, alongside their estimated parameter sizes in Table C.4. Note that some estimations of the parameter sizes are from unofficial sources. This information is only to provide context, and none of our conclusions depend on these estimates. The corresponding results of GRE-bench computed with Llama3.1 8B is shown in Table C.5.

Table C.4: Results of GRE-bench based on GEM, GEM-S (abstract), and GEM-S (ASSW) with Llama3.1 70B. In the column of parameter size, the symbol ∼ indicates that this size is estimated. For mix-of-expert (MoE) models, we show the total parameter size instead of the activated parameter size.

Model Name	**Para. Size** (in Billion)	**GRE-bench** GEM	GEM-S(abs.)	GEM-S(ASSW)
deepseek/deepseek-chat	671	19.27	6.88	1.74
deepseek/deepseek-r1	671	17.28	7.23	1.94
openai/gpt-4o-mini (v 20240718)	∼ 8	16.31	5.25	0.52
openai/gpt-4o (v 20240513)	∼ 200	20.08	7.62	2.82
openai/gpt-3.5-turbo (v 20240125)	∼ 20	19.28	6.31	1.40
anthropic/claude-2	∼ 137	20.04	7.17	2.27
anthropic/claude-3-haiku (v 20240307)	∼ 20	20.32	7.63	2.50
anthropic/claude-3-sonnet (v 20240229)	∼ 70	24.11	9.53	4.25
anthropic/claude-3-opus (v 20240229)	∼ 2000	25.77	11.52	5.89
google/gemini-pro (v 002)	∼ 20	14.73	4.09	-0.26
google/gemini-pro-1.5 (v 002)	∼ 175	22.76	9.61	4.47
meta-llama/llama-3-8b-instruct	8	16.68	5.81	1.07
meta-llama/llama-3-70b-instruct	70	18.35	7.23	3.37
mistralai/mixtral-8x7b-instruct (v 0.1)	56	16.86	5.22	1.23
mistralai/mixtral-8x22b-instruct (v 0.1)	176	18.94	5.82	1.64
meta-llama/llama-3.1-8b-instruct	8	18.74	7.60	2.40
meta-llama/llama-3.1-70b-instruct	70	19.95	8.00	3.47
mistralai/mistral-large (v 2407)	123	20.40	7.44	2.72
mistralai/mistral-small (v 2409)	22	18.08	6.70	1.28
mistralai/mistral-tiny (v mistral-7b-0.3)	7	17.21	4.11	0.15

Table C.5: Results of GRE-bench based on GEM, GEM-S (abstract), and GEM-S (ASSW) with Llama3.1 8B. In the column of parameter size, the symbol ∼ indicates that this size is estimated. For mix-of-expert (MoE) models, we show the total parameter size instead of activated parameter size.

Model Name	**Para. Size** (in Billion)	**GRE-bench** GEM	GEM-S(abs.)	GEM-S(ASSW)
deepseek/deepseek-chat	671	47.51	17.40	11.98
deepseek/deepseek-r1	671	47.45	18.89	13.80
openai/gpt-4o-mini (v 20240718)	∼ 8	42.55	14.44	10.25
openai/gpt-4o (v 20240513)	∼ 200	48.01	18.73	12.78
openai/gpt-3.5-turbo (v 20240125)	∼ 20	45.62	15.86	10.72
anthropic/claude-2	∼ 137	55.94	24.29	17.22
anthropic/claude-3-haiku (v 20240307)	∼ 20	49.77	19.58	13.34
anthropic/claude-3-sonnet (v 20240229)	∼ 70	56.49	24.16	16.52
anthropic/claude-3-opus (v 20240229)	∼ 2000	60.94	27.94	20.21
google/gemini-pro (v 002)	∼ 20	40.65	12.86	9.49
google/gemini-pro-1.5 (v 002)	∼ 175	52.94	22.43	16.06
meta-llama/llama-3-8b-instruct	8	43.50	15.49	11.24
meta-llama/llama-3-70b-instruct	70	46.39	18.60	13.68
mistralai/mixtral-8x7b-instruct (v 0.1)	56	45.18	15.61	11.25
mistralai/mixtral-8x22b-instruct (v 0.1)	176	47.29	17.03	11.85
meta-llama/llama-3.1-8b-instruct	8	46.51	18.33	12.69
meta-llama/llama-3.1-70b-instruct	70	48.29	19.59	13.91
mistralai/mistral-large (v 2407)	123	47.84	17.84	12.52
mistralai/mistral-small (v 2409)	22	45.01	15.90	11.10
mistralai/mistral-tiny (v mistral-7b-0.3)	7	43.90	13.34	9.94

C.4.3 Consistent Results over Preprocessing Models

To answer the question *"Does using an open-sourced preprocessing model (rather than GPT4o) still lead to effective GEM metric and similar GRE-bench results?"*, we use a Llama3.2-90B-Vision-Instruct model for preprocessing and re-run all validating experiments mentioned in Section 4.4. We find that most experimental results remain valid except for two manipulation tests. The GRE-bench results also remain highly consistent with the results of using GPT4o preprocessing.

Positive Correlation with Human Annotation GEM still shows a significant positive correlation with human annotation in the proposal peer grading dataset (Section 4.4.1), however, the Spearman's correlation coefficient between GEM scores and instructor grades drops from 0.433 (GPT4o preprocessing) to 0.274 (Llama3.2-90b preprocessing).

Sensitivity to Degradation In the degradation experiment (Section 4.4.2), GEM with Llama3.2-90b preprocessing also shows significant sensitivity to all degradations and even

performs better in "sentence deletion" than using GPT4o to preprocess.

Table C.6: Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after degradations of GEM and GEM-S metrics (preprocessing with Llama-3.2 90B) with 95% confidence intervals in parentheses. Significant score decreases ($p < 0.05$) after degradations are highlighted in bold green, implying the evaluation metric can effectively penalize the degradation.

Evaluation Metric	Sentence Deletion	Deletion & Completion	Abstract-only Review
GEM (Llama Preprocessing)	**-0.566** (-0.625,-0.507)	**-0.317** (-0.371,-0.263)	**-0.078** (-0.154,-0.002)
GEM-S (Llama Preprocessing)	**-0.549** (-0.607,-0.491)	**-0.205** (-0.258,-0.153)	**-0.327** (-0.399,-0.254)

Weaker Robustness than GPT-4o Preprocessing In the manipulation experiments (Section 4.4.3), GEM with Llama3.2-90b preprocessing is robust against "GPT4 rephrase" but fails in the "Llama-3.1 rephrase" and "Meaningless Elongation".

Table C.7: Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after manipulations of GEM and GEM-S metrics (preprocessing with Llama-3.2 90B) with 95% confidence intervals in parentheses. Significant score increases ($p < 0.05$) after manipulations are highlighted in bold red, implying the evaluation metrics are not robust against the manipulation.

Evaluation Metric	GPT-4o Rephrase	Llama3.1 Rephrase	Meaningless Elongation
GEM (Llama Preprocessing)	-0.132 (-0.180,-0.084)	**0.124** (0.075,0.173)	**0.138** (0.086,0.189)
GEM-S (Llama Preprocessing)	-0.126 (-0.173,-0.079)	**0.156** (0.109,0.203)	**0.112** (0.061,0.163)

Consistent GRE-bench Results Based on the GEM metric with llama3.2-90b preprocessing, we re-compute the GRE-bench scores with ICLR 2023 dataset (Section 4.5). The Spearman's correlation coefficient between the GRE-bench scores based on the GEM metric with Llama3.2-90B preprocessing and the scores with GPT-4o preprocessing shown in Table C.5 is 0.89. The analogous coefficient for the GRE-bench scores based on GEM-S (abstract) and GEM-S (ASSW) are 0.85 and 0.83 respectively. All detailed numerical results are provided in Table C.8.

Table C.8: Results of GRE-bench based on GEM, GEM-S (abstract), and GEM-S (ASSW) with Llama3.2 90B preprocessing. In the column of parameter size, the symbol ∼ indicates that this size is estimated.

Model Name	**Para. Size** (in Billion)	**GRE-bench** GEM	GEM-S(abs.)	GEM-S(ASSW)
openai/gpt-4o-mini (v 20240718)	∼ 8	45.76	19.58	18.27
openai/gpt-4o (v 20240513)	∼ 200	50.17	23.12	20.55
openai/gpt-3.5-turbo (v 20240125)	∼ 20	48.56	20.92	19.04
anthropic/claude-2	∼ 137	54.61	26.07	23.93
anthropic/claude-3-haiku (v 20240307)	∼ 20	52.70	24.87	22.18
anthropic/claude-3-sonnet (v 20240229)	∼ 70	54.83	25.82	22.78
anthropic/claude-3-opus (v 20240229)	∼ 2000	62.08	33.13	30.53
google/gemini-pro (v 002)	∼ 20	46.75	21.33	20.43
google/gemini-pro-1.5 (v 002)	∼ 175	56.36	27.40	24.44
meta-llama/llama-3-8b-instruct	8	47.36	20.96	19.86
meta-llama/llama-3-70b-instruct	70	55.62	29.61	27.98
mistralai/mixtral-8x7b-instruct (v 0.1)	56	50.53	22.83	21.08
mistralai/mixtral-8x22b-instruct (v 0.1)	176	51.57	23.84	21.79
meta-llama/llama-3.1-8b-instruct	8	51.04	24.51	22.28
meta-llama/llama-3.1-70b-instruct	70	50.26	23.68	21.73
mistralai/mistral-large (v 2407)	123	51.81	23.59	21.20
mistralai/mistral-small (v 2409)	22	50.22	22.99	20.58
mistralai/mistral-tiny (v mistral-7b-0.3)	7	48.42	20.04	18.86

Overall, as more advanced models can preprocess reports more accurately, the GEM metric will be more accurate and manipulation-resistant with superior preprocessing. Considering the frequent release and update of LLMs, utilizing more advanced LLMs for preprocessing enhances both the accuracy and robustness of the GEM metric. Consequently, our GEM metrics stand to benefit from the continuous advancements in LLM technology.

C.4.4 Validation on Yelp Restaurant Review Dataset

To further validate our GEM metrics' effectiveness in evaluating judgments without gold-standard references in scenarios in addition to peer review. We use an open-access online business review data from Yelp, and repeat the same workflow in Sections 4.4.2 and 4.4.3, to check how sensitive GEM metrics are to degradations and how robust GEM metrics are to manipulations.

The results are similar to those observed in experiments on ICLR dataset (Sections 4.4.2 and 4.4.3). Our GEM and GEM-S demonstrate significant sensitivity to degradations and robustness against manipulations.

Yelp Review We randomly sample 300 restaurants from the entire dataset which have at least 50 reviews with at least 1000 characters. For each restaurant, we randomly sample 3 customer reviews among all reviews with at least 1000 characters. Yelp reviews quality varies much more than ICLR reviews, thus, we use length as a heuristic filter to get reasonable quality reviews.

GEM Setup We use GPT-4o for text preprocessing, and use Llama-3.1 8B as the evaluation-LM. For GEM-S, we use the categories of the restaurant as the synopsis, which is given by the Yelp dataset.

GEM demonstrates sensitivity to degradations. We reuse the *Sentence Deletion* and *Deletion & Completion* degradation methods described in Section 4.4.2. However, since restaurant reviews lack "abstract", the *Abstract-only Review* degradation cannot be applied. Instead, we propose *Random Replacement* degradation, where we degrade the original review x_i by replacing it with a random review x_i' of another randomly selected restaurant.

Table C.9 presents the standardized mean differences (SMD) with 95% confidence intervals for GEM and baseline metrics before and after these degradations. The results align with those observed in the ICLR dataset (Section 4.4.2): GEM and GEM-S consistently exhibit significant SMD across all three degradations. Additionally, while LMExaminer is more sensitive to degradations like *Sentence Deletion* and *Deletion & Completion*, which impair presentation quality, GEM shows greater sensitivity to the *Random Replacement* degradation, which severely compromises semantic informativeness.

GEM demonstrates robustness to manipulation. Similarly, we reuse the *GPT-4o Rephrase* and *Llama-3.1 Rephrase* manipulations to test whether our metric is robust against specific language styles. However, since Yelp restaurant reviews lack a structured format like

Table C.9: Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after degradations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score decreases ($p < 0.05$) after degradations are highlighted in bold green, implying the evaluation metric can effectively penalize the degradation.

Evaluation Metric	Sentence Deletion	Deletion & Completion	Random Replacement
BLEU	**-0.348** (-0.436,-0.260)	**-0.113** (-0.200,-0.026)	**-0.231** (-0.355,-0.107)
ROUGE-L	**-0.677** (-0.765,-0.589)	0.009 (-0.064,0.081)	**-0.345** (-0.462,-0.229)
BERTscore	**-0.318** (-0.359,-0.276)	0.011 (-0.032,0.054)	**-0.722** (-0.836,-0.609)
BARTscore-F1	0.744 (0.705,0.784)	0.457 (0.427,0.487)	-0.042 (-0.163,0.079)
BARTscore-precision	0.877 (0.831,0.924)	0.536 (0.501,0.571)	-0.023 (-0.159,0.113)
BARTscore-recall	**-0.015** (-0.018,-0.012)	**-0.014** (-0.017,-0.011)	**-0.046** (-0.051,-0.041)
LMExaminer	**-1.245** (-1.328,-1.162)	**-0.466** (-0.550,-0.382)	**-0.204** (-0.340,-0.068)
GEM	**-0.386** (-0.459,-0.314)	**-0.238** (-0.318,-0.159)	**-0.516** (-0.624,-0.408)
GEM-S	**-0.361** (-0.436,-0.287)	**-0.195** (-0.277,-0.114)	**-0.283** (-0.385,-0.182)

ICLR peer reviews (as illustrated in Figure 4.4), it is unclear how to construct a suitable *Meaningless Elongation* manipulation.

Table C.10 shows the standardized mean differences (SMD) with 95% confidence intervals for GEM and baseline metrics before and after these manipulations. The results remain similar to those observed in Section 4.4.3. GEM metrics remain robust against both types of manipulation, while LMExaminer shows a significant increase in scores after LLM rephrasing.

Table C.10: Standardized Mean Differences (SMD) of scores $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ after manipulations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score increases ($p < 0.05$) after manipulations are highlighted in bold red, implying the evaluation metrics are not robust against the manipulation.

Evaluation Metric	GPT-4o Rephrase	Llama3.1 Rephrase
BLEU	**0.100** (0.014,0.185)	0.046 (-0.063,0.156)
ROUGE-L	0.020 (-0.037,0.076)	**0.083** (0.008,0.157)
BERTscore	-0.016 (-0.041,0.008)	-0.098 (-0.133,-0.064)
BARTscore-F1	**0.316** (0.287,0.346)	**0.425** (0.371,0.478)
BARTscore-precision	**0.371** (0.337,0.405)	**0.508** (0.445,0.571)
BARTscore-recall	-0.012 (-0.013,-0.010)	-0.016 (-0.018,-0.014)
LMExaminer	**0.300** (0.240,0.361)	**0.453** (0.387,0.518)
GEM	0.021 (-0.035,0.076)	0.010 (-0.055,0.075)
GEM-S	0.020 (-0.037,0.077)	-0.012 (-0.080,0.055)

C.4.5 Exploration of Properties of LLM Estimated Distribution

An interesting question raised by our anonymous reviewer is:

As we know $\Pr[x, y] = \Pr[x] \cdot \Pr[y|x] = \Pr[y] \cdot \Pr[x|y]$*, does this symmetry also apply to the LLM estimated probability, i.e.,* $\Pr_{\text{LLM}}[x] \cdot \Pr_{\text{LLM}}[y|x] = \Pr_{\text{LLM}}[y] \cdot \Pr_{\text{LLM}}[x|y]$*?*

To answer this question, we randomly select 500 papers from ICLR 2023 and 1874 corresponding reviews in total. For each pair of reviews x, y for the same paper, we use our method to compute $\log \Pr_{\text{LLM}}[x|y]$, $\log \Pr_{\text{LLM}}[y|x]$, $\log \Pr_{\text{LLM}}[x]$, and $\log \Pr_{\text{LLM}}[y]$. For those 5328 pairs, we compute the Spearman correlation coefficient between $\Pr_{\text{LLM}}[x] \cdot \Pr_{\text{LLM}}[x|y] := \exp(\log \Pr_{\text{LLM}}[x|y] + \log \Pr_{\text{LLM}}[y])$ and $\Pr_{\text{LLM}}[y] \cdot \Pr_{\text{LLM}}[y|x] := \exp(\log \Pr_{\text{LLM}}[y|x] + \log \Pr_{\text{LLM}}[x])$. The Spearman correlation coefficient is 0.943, which indicates that they are highly positively correlated, despite minor differences. Potential noise in the estimation process may account for these differences.

The results revealed a high positive correlation between these two values, although they were not precisely equal. Despite this difference, the empirical results in our paper indicate that this does not negatively impact the desired properties, including the accuracy and manipulation resistance.

C.5 GEM / GEM-S with Fine-tuned Evaluation-LM

We not only use the original Llama-3.1-8B-Instruct as the Evaluation-LM, but also fine-tune it to better estimate the conditional and marginal probabilities. In this section, we provide the details of how we fine-tune the model, as well as the experimental results.

C.5.1 Implementation

We fine-tune the original Llama-3.1-8B-Instruct model using 1,000 papers and their corresponding reviews from ICLR 2023 (which are separate from the papers used for experiments). Let $\texttt{gen}(\text{metric}, x, y)$ represent the function that generates the full prompt for predicting judgments (both input and output), where $x = \texttt{Null}$ indicates that we are estimating the marginal probability. For each paper, we randomly selected three reviews, denoted as r_1, r_2, and r_3, along with their preprocessed versions $\hat{r}_1$, $\hat{r}_2$, and $\hat{r}_3$. From each paper, we generated nine data points for GEM and nine data points for GEM-S as follows: $\texttt{gen}(\texttt{GEM(-S)}, \hat{r}_1, \hat{r}_2)$, $\texttt{gen}(\texttt{GEM(-S)}, \hat{r}_1, \hat{r}_3)$, $\texttt{gen}(\texttt{GEM(-S)}, \hat{r}_2, \hat{r}_1)$, $\texttt{gen}(\texttt{GEM(-S)}, \hat{r}_2, \hat{r}_3)$, $\texttt{gen}(\texttt{GEM(-S)}, \hat{r}_3, \hat{r}_1)$, $\texttt{gen}(\texttt{GEM(-S)}, \hat{r}_3, \hat{r}_2)$, $\texttt{gen}(\texttt{GEM(-S)}, \texttt{Null}, \hat{r}_1)$, $\texttt{gen}(\texttt{GEM(-S)}, \texttt{Null}, \hat{r}_2)$, $\texttt{gen}(\texttt{GEM(-S)}, \texttt{Null}, \hat{r}_3)$.

In total, this process yield 18,000 input-output pairs. We used the Unsloth framework to fine-tune the model, running on a physical server equipped with a single NVIDIA A100 GPU, provided by Google Colab. Here is the settings we use for fine-tuning:

```
"model_config": {
    "base_model":"unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit", # The base model
    "max_seq_length": 4096, # The maximum sequence length
    "load_in_4bit": True, # Load the model in 4-bit
},
"lora_config": {
    "r": 16, # The number of LoRA layers 8, 16, 32, 64
    "lora_alpha":16, # The alpha value for LoRA
    "lora_dropout":0, # The dropout value for LoRA
},
"training_config": {
    "per_device_train_batch_size": 2, # The batch size
    "gradient_accumulation_steps": 4, # The gradient accumulation steps
    "warmup_steps": 5, # The warmup steps
    "max_steps":0, # The maximum steps
    "num_train_epochs": 3, # The number of training epochs
    "learning_rate": 2e-4, # The learning rate
    "optim" :"adamw_8bit", # The optimizer
    "weight_decay" : 0.01,  # The weight decay
    "lr_scheduler_type": "linear", # The learning rate scheduler
    "seed" : 42, # The seed
}
```

C.5.2 Experimental Results

Table C.11: Standardized Mean Differences (SMD) $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ of scores after degradations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score decreases ($p < 0.05$) after degradations are highlighted in bold green, implying the evaluation metric can effectively penalize the degradation.

Evaluation Metric	Sentence Deletion	Deletion & Completion	Abstract-only Review
GEM-raw	**-0.123** (-0.126,-0.120)	**-0.126** (-0.132,-0.121)	0.020 (0.017,0.023)
GEM	**-0.401** (-0.448,-0.354)	**-0.308** (-0.358,-0.258)	**-0.191** (-0.261,-0.122)
GEM-S	**-0.409** (-0.455,-0.362)	**-0.206** (-0.254,-0.158)	**-0.566** (-0.639,-0.492)
GEM-finetune	**-0.155** (-0.199,-0.110)	**-0.135** (-0.180,-0.090)	-0.043 (-0.106,0.020)
GEM-S-finetune	**-0.192** (-0.244,-0.140)	**-0.099** (-0.148,-0.049)	**-0.355** (-0.422,-0.288)

Table C.12: Standardized Mean Differences (SMD) of scores $d := (\mu' - \mu)/\sigma_{\text{pooled}}$ after manipulations of various evaluation metrics with 95% confidence intervals in parentheses. Significant score increase ($p < 0.05$) after manipulations are highlighted in bold red, implying the evaluation metric are not robust against the manipulation.

Evaluation Metric	GPT-4o Rephrase	Llama3.1 Rephrase	Meaningless Elongation
GEM-raw	-0.123 (-0.126,-0.120)	-0.126 (-0.132,-0.121)	**0.020** (0.017,0.023)
GEM	-0.058 (-0.090,-0.026)	-0.107 (-0.143,-0.070)	-0.063 (-0.097,-0.030)
GEM-S	-0.046 (-0.079,-0.013)	-0.114 (-0.149,-0.078)	-0.070 (-0.104,-0.036)
GEM-finetune	-0.014 (-0.044,0.015)	-0.010 (-0.044,0.024)	-0.056 (-0.088,-0.025)
GEM-S-finetune	-0.026 (-0.062,0.010)	-0.045 (-0.085,-0.004)	-0.074 (-0.112,-0.037)

C.5.3 GRE-bench based on finetuned GEM and GEM-S

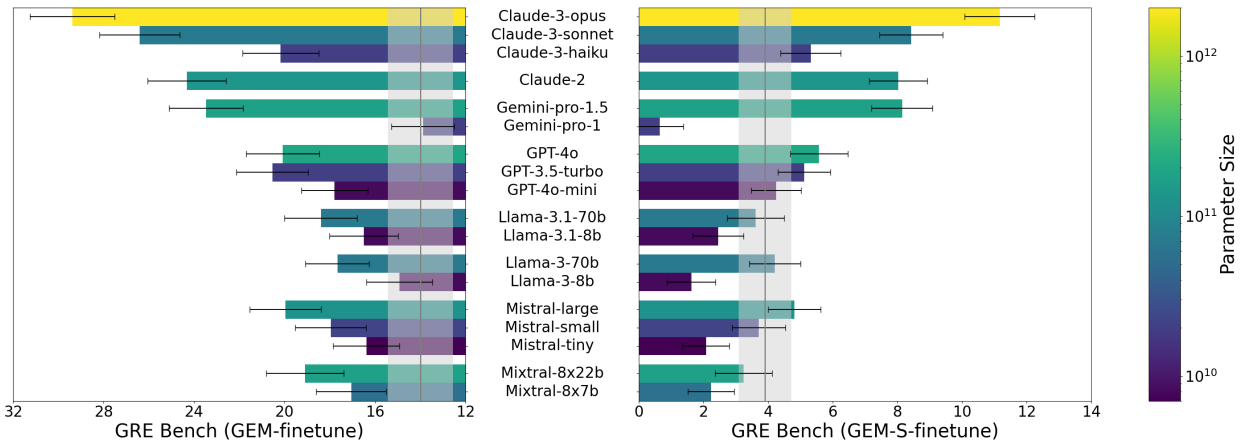


Figure C.1: Results of GRE-bench based on finetuned GEM and GEM-S with 90% confidence intervals vs. model parameter size. The grey line represents the average human baseline, with the 90% confidence interval shaded in grey.

APPENDIX D

Appendix for Aligned Scoring Mechanisms

D.1 Test Principle Details

In this section, we provide the details of degradation and manipulation strategies used in our experiments.

Table D.1: Degradation strategies.

Strategy	**Description**	**Provenance**
Random Replacement	Replace the original candidate with a response sampled from a different task within the same task family.	Lu et al. [2024]
Sentence Deletion	Delete every other sentence while preserving section headers, ordering, and visible format.	Xu et al. [2025]
Deletion & Completion	Delete every other sentence, mark each gap, then ask a helper LLM to fill the gaps using only the remaining text and the original format.	Xu et al. [2025]
Surface Report	Regenerate the response using only weak context, such as the abstract for peer review or the headline plus lead paragraph for summarization.	Robertson and Koyejo [2025], Xu et al. [2025]
Ultra-Concise Compression	Summarize under a severe length cap (e.g., 10% of the original), so that specific evidence and caveats must be dropped.	Robertson and Koyejo [2025]
Opinion Flip	Reverse the overall evaluative verdict (e.g., accept → reject, positive → negative) while preserving the surface structure and topical vocabulary of the original candidate.	Robertson and Koyejo [2025]

Table D.2: Manipulation strategies.

Strategy	Description	Provenance
Rephrase	Ask an LLM to rewrite the candidate in a specific style while preserving meaning and high-level structure.	Xu et al. [2025]
Meaningless Elongation	Append the same fixed, semantically empty filler sentences to every candidate report without adding task-relevant information.	Xu et al. [2025]
Opinion Shift (Positive)	Rewrite the candidate to preserve structure and coverage while systematically shifting stance toward more favorable sentiment.	Robertson and Koyejo [2025]
Opinion Shift (Negative)	Rewrite the candidate to preserve structure and coverage while systematically shifting stance toward less favorable sentiment.	Robertson and Koyejo [2025]
Opinion Shift (Neutral)	Rewrite claims as hedged, noncommittal statements (e.g., "the method is effective" → "the method may have some merit"), flattening the evaluative signal without changing topical content.	This work
Opinion Shift (Extreme)	Amplify all claims to their strongest possible form (e.g., "a reasonable approach" → "a groundbreaking contribution"), testing whether a metric rewards confidence or extremity as a proxy for quality.	This work

D.2 Approximate Manipulation Robustness of the f-MI Metrics

The information processing inequality in Proposition 5.3 guarantees that any manipulation $\sigma : \mathcal{X} \to \mathcal{X}$ of a candidate report can only reduce the true f-MI:

$$I_f(\sigma(X); Y) \leq I_f(X; Y).$$

However, the score is computed using a fixed estimator and distribution model, which in practice amounts to using a restricted or suboptimal critic rather than the population optimum. As a result, the estimated score need not satisfy DPI exactly, even when the underlying information measure does. This motivates the following approximate notion of robustness: for a class Σ of manipulations, suppose

$$\sup_{\sigma \in \Sigma} |\mathbb{E}\left[S(\sigma(X), Y)\right] - I_f(\sigma(X); Y)| \leq \varepsilon.$$

Then for every $\sigma \in \Sigma$,

$$\mathbb{E}\left[S(\sigma(X), Y)\right] \leq \mathbb{E}\left[S(X, Y)\right] + 2\varepsilon.$$

Indeed,

$$\mathbb{E}\left[S(\sigma(X), Y)\right] \leq I_f(\sigma(X); Y) + \varepsilon \leq I_f(X; Y) + \varepsilon \leq \mathbb{E}\left[S(X, Y)\right] + 2\varepsilon,$$

where the middle inequality is exactly Proposition 5.3. Thus, any manipulation that decreases the true f-mutual information by more than 2ε is still penalized by the estimated score in expectation.

The assumption requires the estimation error to be uniformly small over the entire manipulation class Σ. This is a strong condition: it asks the critic to generalize not only to clean reports but also to all manipulated variants. In practice, ε may be large for manipulation strategies that produce out-of-distribution inputs for the critic. Empirical validation (Section 5.3) is therefore essential.

D.3 TV, f-variational, Statement-level, LLM-oracle Metric

This method instantiates the scoring framework with

$$F = \text{TV}, \qquad E = f\text{-variational}, \qquad D = \text{LLM-as-a-Judge (statement-level) denoted as } D_\Psi,$$

where D_Ψ is induced by a reference decomposition Ψ.

We keep the candidate at the report level and decompose each reference into statements:

$$\Psi(y) = \Big(\psi_1(y), \ldots, \psi_{K(y)}(y)\Big). \tag{D.1}$$

We instantiate the local critic t and aggregation rule A from Section 5.4.4 as follows. For each candidate–statement pair, an LLM judge returns a centered binary local critic

$$t : \mathcal{V}^{\leq L} \times \mathcal{V}^{\leq L} \to \left\{-\frac{1}{2}, \frac{1}{2}\right\}, \tag{D.2}$$

where $t(x, w) = \frac{1}{2}$ if candidate x supports or aligns with statement w, and $t(x, w) = -\frac{1}{2}$ otherwise. We use mean aggregation, so the induced report-level critic is

$$T_\Psi(x, y) = \frac{1}{K(y)} \sum_{k=1}^{K(y)} t\Big(x, \psi_k(y)\Big). \tag{D.3}$$

Since each summand lies in $[-\frac{1}{2}, \frac{1}{2}]$, T_Ψ satisfies the constraint in the TV variational estimator.

For task i, let $\mathcal{R}_i^+$ be positive references paired with the same task, and let $\mathcal{R}_i^-$ be negative references sampled from the marginal reference distribution P_Y. We define the sampled

contrastive score

$$\hat{S}_{\mathrm{CTV}}(x_i;\mathcal{R}_i^+,\mathcal{R}_i^-)=\frac{1}{|\mathcal{R}_i^+|}\sum_{y\in\mathcal{R}_i^+}T_\Psi(x_i,y)-\frac{1}{|\mathcal{R}_i^-|}\sum_{y\in\mathcal{R}_i^-}T_\Psi(x_i,y). \tag{D.4}$$

Assume that, conditional on task ψ_i, the references in $\mathcal{R}_i^+$ are drawn i.i.d. from $P_{Y|\psi_i}$, and the references in $\mathcal{R}_i^-$ are drawn i.i.d. from P_Y, independently of (x_i,ψ_i). Then

$$\mathbb{E}\left[\hat{S}_{\mathrm{CTV}}(x_i;\mathcal{R}_i^+,\mathcal{R}_i^-)\mid x_i,\psi_i\right]=\mathbb{E}_{Y\sim P_{Y|\psi_i}}\Big[T_\Psi(x_i,Y)\Big]-\mathbb{E}_{Y\sim P_Y}\Big[T_\Psi(x_i,Y)\Big]. \tag{D.5}$$

If (X_i,Y_i) is generated by first sampling a task and then sampling the candidate and reference independently conditional on that task, averaging Eq. (D.5) over (X_i,ψ_i) yields

$$\mathbb{E}\Big[\hat{S}_{\mathrm{CTV}}(X_i;\mathcal{R}_i^+,\mathcal{R}_i^-)\Big]=\mathbb{E}_{P_{XY}}\Big[T_\Psi(X,Y)\Big]-\mathbb{E}_{P_X\otimes P_Y}\Big[T_\Psi(X,Y)\Big]. \tag{D.6}$$

Because T_Ψ factors through $\Psi(Y)$ and satisfies $\|T_\Psi\|_\infty\le 1/2$, the TV f-variational form (Equation (5.2)) applied to the transformed pair $(X,\Psi(Y))$ gives

$$\mathbb{E}\Big[\hat{S}_{\mathrm{CTV}}(X_i;\mathcal{R}_i^+,\mathcal{R}_i^-)\Big]\le I_{\mathrm{TV}}\Big(X;\Psi(Y)\Big)\le I_{\mathrm{TV}}(X;Y).$$

D.4 Implementation Details

D.4.1 Implementation

We instantiate the token-autoregressive model with `Llama-3.1-8B`, and adopt the preprocessing process from Lu et al. [2024], Xu et al. [2025] with `claude-haiku-4.5`.

We instantiate the LLM-oracle with `claude-haiku-4.5`, queried through OpenRouter API at temperature 0 with a single sample per prompt. In the f-variational settings, the oracle output is used directly as the critic $T(x,y)$. In the direct-estimation settings, the oracle returns a 7-point ordinal supporting judgment, which we map to {1/8, 1/4, 1/2, 1, 2, 4, 8} and treat as a discretized surrogate for the density ratio estimate $\hat{r}(x,y)$. For statement-level representation, we first decompose the reference y into atomic claims using a dataset-specific decomposition prompt, and then score the resulting (candidate, reference statement) pairs with the same claude-haiku-4.5 model.

Unless otherwise noted, each candidate is scored against one positive reference, and, for f-variational metrics, four negative references sampled from other tasks in the same dataset group and takes average to reduce noise. A group consists of same-kind tasks within a

dataset, such as the same class in Peer Grading-WH or the same topic category (e.g., math or technology) in LFQA-E. ICLR 2026, Peer Grading-XLSK, SummEval, and SPACE are treated as a single group.

D.4.2 Computational Resources

All experiments were conducted on a high-performance workstation with the following specifications:

Hardware. The machine is equipped with dual Intel Xeon Platinum 8470Q CPUs (totaling 104 physical cores and 208 threads), 1.0 TiB of system RAM, and one NVIDIA RTX PRO 6000 (Blackwell architecture) GPU with 96 GB of device memory.

Local Inference. For metrics using the Llama-3.1-8B model, local inference was performed with an average processing speed of approximately 0.5s/it. The total local compute time required for the entire study (excluding API latency) was approximately 6 GPU hours.

API Usage and Reliability. Evaluations involving Claude-Haiku-4.5 were conducted via the OpenRouter API.

- Configuration: All API calls used a default max token limit of 4000 and a temperature setting of 0 to ensure deterministic and complete responses.
- Retry Logic: To handle potential network instability or API timeouts, we implemented a robust retry mechanism with a maximum of 4 attempts per request; a call was marked as a failure only if all retries were exhausted.
- Error Handling: In cases where specific perturbation methods (for degradation or manipulation tests) encountered API errors or generation failures, the system was designed to gracefully fall back to a no-op (no operation), ensuring the stability of the overall evaluation pipeline.

Execution Time. Due to varying dataset sizes, the total experimental duration per dataset ranged from 20 minutes to 1 hour.

D.5 Detailed Degradation and Manipulation Statistics

Table D.3: Degradation perturbation method results. Each cell reports paired standardized change $d_z \pm$ 95% CI half-width. Red values mark failed degradation criteria ($p < 0.05$).

Metric	PG-WH	PG-XLSK	ICLR26	SummEval	SPACE	LFQA-E	MedAESQA
			deletion_and_completion				
MI-based Mechanisms							
KL-Direct-Autoreg.	−0.70±0.15	−0.64±0.17	−0.51±0.15	−0.62±0.15	−0.64±0.15	−0.15±0.14	−0.33±0.14
TVD-FVar-Report	−0.04±0.14	−0.25±0.15	−0.22±0.14	−0.21±0.14	−0.43±0.14	−0.22±0.14	+0.19±0.14
TVD-FVar-Statement	−0.27±0.14	−0.33±0.16	−0.21±0.14	−0.63±0.15	−0.48±0.15	−0.28±0.14	−0.25±0.14
Non-MI baselines							
ROUGE-L	+0.09±0.14	+0.22±0.15	−0.97±0.17	−0.09±0.14	−0.52±0.15	−0.48±0.15	−0.32±0.14
BLEU	+0.44±0.14	+0.12±0.15	−0.84±0.16	+0.08±0.14	−0.38±0.14	−0.29±0.14	−0.30±0.14
BERTScore	−4.55±0.46	+0.62±0.17	−0.05±0.14	+0.61±0.15	−0.61±0.15	+0.08±0.14	+0.04±0.14
LLM-J / Claude-Haiku-4.5	−0.45±0.14	−0.31±0.16	−0.35±0.14	−0.30±0.14	−0.48±0.15	−0.40±0.14	−0.02±0.14
LLM-J / Claude-Sonnet-4.5	−0.51±0.15	−0.29±0.16	−0.41±0.14	−0.52±0.15	−0.65±0.15	−0.49±0.15	−0.21±0.14
LLM-J / GPT-5-mini	−0.34±0.14	−0.25±0.15	−0.36±0.14	−0.34±0.14	−0.60±0.15	−0.21±0.14	−0.13±0.14
LLM-J / GPT-4o-mini	+0.08±0.14	+0.38±0.16	−0.15±0.14	−0.17±0.14	−0.14±0.14	−0.17±0.14	−0.24±0.14
			opinion_flip				
MI-based Mechanisms							
KL-Direct-Autoreg.	−0.47±0.14	−0.68±0.17	−0.70±0.15	–	–	–	–
TVD-FVar-Report	−0.17±0.14	−0.02±0.15	−0.05±0.14	–	–	–	–
TVD-FVar-Statement	−0.60±0.15	−0.25±0.16	−0.26±0.14	–	–	–	–
Non-MI baselines							
ROUGE-L	+0.31±0.14	−1.30±0.21	+0.19±0.14	–	–	–	–
BLEU	+0.79±0.16	−0.82±0.18	−0.19±0.14	–	–	–	–
BERTScore	−4.37±0.45	−1.16±0.20	+0.39±0.14	–	–	–	–
LLM-J / Claude-Haiku-4.5	−0.57±0.15	−0.53±0.16	−0.46±0.15	–	–	–	–
LLM-J / Claude-Sonnet-4.5	−0.67±0.15	−0.60±0.17	−0.56±0.15	–	–	–	–
LLM-J / GPT-5-mini	−0.70±0.15	−0.33±0.16	−0.45±0.15	–	–	–	–
LLM-J / GPT-4o-mini	−0.19±0.14	+0.18±0.15	+0.05±0.14	–	–	–	–
			random_replacement				
MI-based Mechanisms							
KL-Direct-Autoreg.	−0.25±0.14	−0.79±0.18	−1.42±0.20	−1.55±0.21	−0.83±0.16	−1.14±0.18	−0.55±0.15
TVD-FVar-Report	−0.36±0.14	−1.25±0.20	−6.31±0.64	−1.75±0.22	−0.97±0.17	−2.15±0.25	−1.99±0.24
TVD-FVar-Statement	−0.63±0.15	−0.76±0.17	−1.25±0.18	−1.43±0.20	−0.77±0.16	−1.49±0.20	−1.90±0.23
Non-MI baselines							
ROUGE-L	+0.01±0.14	−0.25±0.16	−0.99±0.17	−1.58±0.21	−0.77±0.16	−0.98±0.17	−1.47±0.20
BLEU	+0.01±0.14	−0.16±0.15	−0.31±0.14	−0.77±0.16	−0.67±0.15	−0.66±0.15	−0.67±0.15
BERTScore	−4.03±0.42	−0.33±0.16	−1.10±0.18	−2.15±0.25	−1.04±0.17	−1.41±0.20	−1.98±0.24
LLM-J / Claude-Haiku-4.5	−0.44±0.14	−1.01±0.19	−0.86±0.16	−1.76±0.22	−0.88±0.16	−1.89±0.23	−2.12±0.25
LLM-J / Claude-Sonnet-4.5	−0.49±0.15	−1.18±0.20	−2.82±0.31	−2.25±0.26	−0.88±0.16	−1.99±0.24	−2.09±0.25
LLM-J / GPT-5-mini	−0.39±0.14	−0.86±0.18	−2.46±0.28	−2.13±0.25	−0.74±0.16	−2.15±0.25	−2.24±0.26
LLM-J / GPT-4o-mini	−0.23±0.14	−0.07±0.15	+0.06±0.14	−2.64±0.29	−0.50±0.15	−2.26±0.26	−1.78±0.22
			sentence_deletion				
MI-based Mechanisms							
KL-Direct-Autoreg.	−0.72±0.15	−1.24±0.20	−1.34±0.19	−0.49±0.15	−0.69±0.15	−0.39±0.14	−0.35±0.14
TVD-FVar-Report	−0.05±0.14	−0.26±0.16	−0.20±0.14	−0.15±0.14	−0.35±0.14	−0.22±0.14	−0.03±0.14
TVD-FVar-Statement	−0.30±0.14	−0.32±0.16	−0.51±0.15	−0.68±0.15	−0.46±0.15	−0.29±0.14	−0.60±0.15
Non-MI baselines							

Continued on next page

Table D.3: Degradation perturbation method results (continued). Each cell reports paired standardized change $d_z \pm$ 95% CI half-width. Red values mark failed degradation criteria ($p < 0.05$).

Metric	PG-WH	PG-XLSK	ICLR26	SummEval	SPACE	LFQA-E	MedAESQA
ROUGE-L	−0.06±0.14	+0.85±0.18	−0.37±0.14	−0.04±0.14	−0.17±0.14	−0.64±0.15	−0.17±0.14
BLEU	−0.23±0.14	+0.50±0.16	−0.70±0.15	+0.02±0.14	−0.27±0.14	−0.63±0.15	−0.40±0.14
BERTScore	−4.19±0.43	+1.03±0.19	−0.16±0.14	+0.11±0.14	−0.29±0.14	−0.10±0.14	+0.17±0.14
LLM-J / Claude-Haiku-4.5	−0.43±0.14	−1.01±0.19	−0.80±0.16	−0.37±0.14	−0.55±0.15	−0.83±0.16	−0.44±0.15
LLM-J / Claude-Sonnet-4.5	−0.56±0.15	−1.34±0.21	−1.40±0.20	−0.49±0.15	−0.62±0.15	−0.90±0.16	−0.53±0.15
LLM-J / GPT-5-mini	−0.61±0.15	−1.37±0.21	−1.14±0.18	−0.50±0.15	−0.74±0.16	−0.76±0.16	−0.56±0.15
LLM-J / GPT-4o-mini	−0.68±0.15	−1.30±0.21	−0.88±0.16	−0.56±0.15	−0.57±0.15	−0.93±0.17	−0.87±0.16
`surface_report`							
MI-based Mechanisms							
KL-Direct-Autoreg.	–	−0.57±0.16	−0.02±0.14	−0.26±0.14	−0.99±0.17	+0.05±0.14	−0.09±0.14
TVD-FVar-Report	–	−0.96±0.18	−0.08±0.14	−0.07±0.14	−0.78±0.16	+0.01±0.14	−0.01±0.14
TVD-FVar-Statement	–	−0.51±0.16	−0.24±0.14	+0.04±0.14	−0.66±0.15	+0.13±0.14	−0.01±0.14
Non-MI baselines							
ROUGE-L	–	−2.45±0.31	−0.88±0.16	−0.13±0.14	−1.75±0.22	−0.18±0.14	−0.40±0.14
BLEU	–	−2.07±0.27	−1.19±0.18	+0.58±0.15	−0.95±0.17	−0.03±0.14	−0.46±0.15
BERTScore	–	−2.65±0.32	+0.14±0.14	+1.91±0.23	−3.02±0.33	+0.20±0.14	−0.31±0.14
LLM-J / Claude-Haiku-4.5	–	−1.08±0.19	+0.14±0.14	+0.40±0.14	−1.33±0.19	+0.71±0.15	+0.88±0.16
LLM-J / Claude-Sonnet-4.5	–	−1.17±0.20	+0.04±0.14	+0.26±0.14	−1.74±0.22	+0.60±0.15	+0.83±0.16
LLM-J / GPT-5-mini	–	−1.08±0.19	+0.02±0.14	+0.09±0.14	−0.94±0.17	+0.51±0.15	+0.75±0.16
LLM-J / GPT-4o-mini	–	−0.79±0.17	+0.92±0.17	+1.18±0.18	−0.57±0.15	+0.85±0.16	+0.72±0.16
`ultra_concise_compression`							
MI-based Mechanisms							
KL-Direct-Autoreg.	−0.73±0.15	−1.51±0.22	−1.51±0.20	−0.82±0.16	−1.09±0.18	−0.13±0.14	+0.02±0.14
TVD-FVar-Report	−0.09±0.14	−0.59±0.17	+0.01±0.14	+0.17±0.14	−0.48±0.15	−0.03±0.14	+0.28±0.14
TVD-FVar-Statement	−0.26±0.14	−0.33±0.16	−0.30±0.14	−1.07±0.17	−0.65±0.15	−0.33±0.14	−0.20±0.14
Non-MI baselines							
ROUGE-L	−2.04±0.24	−2.93±0.35	−2.90±0.32	−0.86±0.16	−1.17±0.18	−2.67±0.30	−1.58±0.21
BLEU	−1.53±0.20	−2.16±0.28	−2.64±0.29	−0.90±0.16	−0.87±0.16	−1.25±0.19	−1.04±0.17
BERTScore	−4.67±0.47	−3.16±0.37	−2.01±0.24	+1.10±0.18	−1.26±0.19	−0.44±0.15	−0.43±0.14
LLM-J / Claude-Haiku-4.5	−0.40±0.14	−1.10±0.19	−0.39±0.14	−0.34±0.14	−0.68±0.15	−0.76±0.16	−0.79±0.16
LLM-J / Claude-Sonnet-4.5	−0.49±0.15	−1.33±0.21	−0.96±0.17	−0.77±0.16	−0.87±0.16	−0.89±0.16	−0.69±0.15
LLM-J / GPT-5-mini	−0.26±0.14	−1.22±0.20	−1.03±0.17	−0.57±0.15	−0.82±0.16	−0.84±0.16	−0.64±0.15
LLM-J / GPT-4o-mini	−0.86±0.16	−1.48±0.22	−0.78±0.16	−1.13±0.18	−1.00±0.17	−1.46±0.20	−1.69±0.22

Table D.4: Manipulation perturbation method results. Each cell reports paired standardized change $d_z \pm$ 95% CI half-width. Red values mark failed manipulation criteria ($p < 0.05$).

Metric	PG-WH	PG-XLSK	ICLR26	SummEval	SPACE	LFQA-E	MedAESQA
`meaningless_elongation`							
MI-based Mechanisms							
KL-Direct-Autoreg.	+0.17±0.14	−0.03±0.15	+0.01±0.14	−0.06±0.14	+0.05±0.14	+0.00±0.14	−0.20±0.14
TVD-FVar-Report	+0.05±0.14	−0.04±0.15	−0.02±0.14	+0.12±0.14	−0.20±0.14	−0.04±0.14	+0.06±0.14
TVD-FVar-Statement	+0.11±0.14	−0.06±0.15	−0.06±0.14	+0.03±0.14	+0.05±0.14	+0.09±0.14	−0.08±0.14

Continued on next page

Table D.4: Manipulation perturbation method results (continued). Each cell reports paired standardized change $d_z \pm$ 95% CI half-width. Red values mark failed manipulation criteria ($p < 0.05$).

Metric	PG-WH	PG-XLSK	ICLR26	SummEval	SPACE	LFQA-E	MedAESQA
Non-MI baselines							
ROUGE-L	+0.46±0.14	−1.62±0.23	−2.24±0.26	−1.24±0.18	−1.33±0.19	−0.38±0.14	−0.66±0.15
BLEU	+0.35±0.14	−1.28±0.21	−0.02±0.14	−0.93±0.17	−0.67±0.15	+0.20±0.14	+0.11±0.14
BERTScore	−4.79±0.49	−2.45±0.31	−0.59±0.15	−0.66±0.15	−2.42±0.28	−0.91±0.16	−0.81±0.16
LLM-J / Claude-Haiku-4.5	+0.28±0.14	−0.54±0.16	−0.04±0.14	+0.07±0.14	−0.51±0.15	−0.12±0.14	−0.30±0.14
LLM-J / Claude-Sonnet-4.5	+0.24±0.14	−1.26±0.20	+0.32±0.14	+0.07±0.14	−0.63±0.15	−0.18±0.14	−0.28±0.14
LLM-J / GPT-5-mini	+0.08±0.14	+0.26±0.16	+0.07±0.14	−0.04±0.14	−0.09±0.14	+0.09±0.14	−0.02±0.14
LLM-J / GPT-4o-mini	+0.39±0.14	+0.71±0.17	+0.38±0.14	−0.28±0.14	−0.41±0.14	+0.00±0.14	−0.12±0.14
`opinion_shift_negative`							
MI-based Mechanisms							
KL-Direct-Autoreg.	+0.01±0.14	−0.04±0.15	+0.52±0.15	–	–	–	–
TVD-FVar-Report	+0.03±0.14	−0.07±0.15	−0.04±0.14	–	–	–	–
TVD-FVar-Statement	−0.18±0.14	−0.23±0.15	−0.27±0.14	–	–	–	–
Non-MI baselines							
ROUGE-L	−0.46±0.14	−2.36±0.30	−1.39±0.19	–	–	–	–
BLEU	−0.09±0.14	−2.05±0.27	−0.94±0.17	–	–	–	–
BERTScore	−4.24±0.44	−2.80±0.34	−0.55±0.15	–	–	–	–
LLM-J / Claude-Haiku-4.5	+0.29±0.14	−0.21±0.15	+0.07±0.14	–	–	–	–
LLM-J / Claude-Sonnet-4.5	+0.18±0.14	−0.32±0.17	+0.11±0.14	–	–	–	–
LLM-J / GPT-5-mini	+0.33±0.14	+0.36±0.16	+0.10±0.14	–	–	–	–
LLM-J / GPT-4o-mini	+0.42±0.14	+0.22±0.15	+0.58±0.15	–	–	–	–
`opinion_shift_neutral`							
MI-based Mechanisms							
KL-Direct-Autoreg.	−0.40±0.14	−0.07±0.15	+0.30±0.14	–	–	−0.01±0.14	−0.16±0.14
TVD-FVar-Report	+0.05±0.14	+0.02±0.15	−0.04±0.14	–	–	+0.00±0.14	−0.04±0.14
TVD-FVar-Statement	−0.07±0.14	−0.19±0.15	−0.32±0.14	–	–	+0.01±0.14	−0.14±0.14
Non-MI baselines							
ROUGE-L	+0.19±0.14	−1.24±0.20	−1.43±0.20	–	–	−0.33±0.14	−0.28±0.14
BLEU	+0.56±0.15	−1.42±0.22	−1.80±0.22	–	–	−0.13±0.14	−0.18±0.14
BERTScore	−4.60±0.47	−1.72±0.24	−0.73±0.16	–	–	−0.25±0.14	−0.42±0.14
LLM-J / Claude-Haiku-4.5	+0.27±0.14	+0.72±0.17	+0.65±0.15	–	–	+0.52±0.15	+0.49±0.15
LLM-J / Claude-Sonnet-4.5	+0.22±0.14	+0.61±0.21	+0.68±0.15	–	–	+0.32±0.14	+0.19±0.14
LLM-J / GPT-5-mini	+0.21±0.14	+0.90±0.18	+0.46±0.15	–	–	+0.43±0.14	+0.21±0.14
LLM-J / GPT-4o-mini	+0.87±0.16	+1.41±0.22	+0.91±0.17	–	–	+0.75±0.16	+0.36±0.14
`opinion_shift_positive`							
MI-based Mechanisms							
KL-Direct-Autoreg.	−0.66±0.15	−0.32±0.16	+0.28±0.14	–	–	–	–
TVD-FVar-Report	−0.06±0.14	−0.02±0.15	−0.08±0.14	–	–	–	–
TVD-FVar-Statement	−0.19±0.14	−0.21±0.15	+0.02±0.14	–	–	–	–
Non-MI baselines							
ROUGE-L	−0.25±0.14	−1.59±0.23	−0.98±0.17	–	–	–	–
BLEU	+0.15±0.14	−1.46±0.22	−0.60±0.15	–	–	–	–
BERTScore	−4.70±0.48	−1.77±0.24	−0.13±0.14	–	–	–	–
LLM-J / Claude-Haiku-4.5	−0.21±0.14	+0.48±0.16	+0.01±0.14	–	–	–	–
LLM-J / Claude-Sonnet-4.5	−0.25±0.14	+0.43±0.20	+0.20±0.14	–	–	–	–

Continued on next page

Table D.4: Manipulation perturbation method results (continued). Each cell reports paired standardized change $d_z \pm$ 95% CI half-width. Red values mark failed manipulation criteria ($p < 0.05$).

Metric	PG-WH	PG-XLSK	ICLR26	SummEval	SPACE	LFQA-E	MedAESQA
LLM-J / GPT-5-mini	+0.01±0.14	+0.88±0.18	+0.23±0.14	–	–	–	–
LLM-J / GPT-4o-mini	+1.17±0.18	+1.80±0.25	+0.98±0.17	–	–	–	–
`opinion_shift_strong`							
MI-based Mechanisms							
KL-Direct-Autoreg.	+0.03±0.14	+0.25±0.15	+0.38±0.14	–	–	+0.07±0.14	−0.14±0.14
TVD-FVar-Report	+0.01±0.14	−0.06±0.15	−0.05±0.14	–	–	−0.15±0.14	+0.03±0.14
TVD-FVar-Statement	+0.00±0.14	−0.06±0.15	−0.07±0.14	–	–	−0.01±0.14	+0.02±0.14
Non-MI baselines							
ROUGE-L	+0.52±0.15	−2.31±0.29	−1.63±0.21	–	–	−0.38±0.14	−0.10±0.14
BLEU	+0.67±0.15	−2.06±0.27	−1.95±0.24	–	–	−0.18±0.14	−0.28±0.14
BERTScore	−4.52±0.46	−3.25±0.38	−0.84±0.16	–	–	−0.33±0.14	−0.55±0.15
LLM-J / Claude-Haiku-4.5	+0.23±0.14	+0.35±0.16	+0.52±0.15	–	–	+0.45±0.15	+0.42±0.14
LLM-J / Claude-Sonnet-4.5	+0.21±0.14	+0.32±0.20	+0.58±0.15	–	–	+0.20±0.14	+0.10±0.14
LLM-J / GPT-5-mini	+0.19±0.14	+0.76±0.17	+0.39±0.14	–	–	+0.29±0.14	+0.15±0.14
LLM-J / GPT-4o-mini	+0.65±0.15	+1.29±0.21	+0.91±0.17	–	–	+0.59±0.15	+0.29±0.14
`rephrase`							
MI-based Mechanisms							
KL-Direct-Autoreg.	−0.12±0.14	−0.04±0.15	+0.18±0.14	−0.19±0.14	−0.61±0.15	+0.02±0.14	−0.23±0.14
TVD-FVar-Report	−0.01±0.14	−0.02±0.15	−0.12±0.14	−0.11±0.14	−0.09±0.14	−0.17±0.14	+0.07±0.14
TVD-FVar-Statement	+0.00±0.14	−0.20±0.15	−0.25±0.14	−0.09±0.14	−0.05±0.14	−0.08±0.14	−0.20±0.14
Non-MI baselines							
ROUGE-L	+0.51±0.15	−2.10±0.27	−1.57±0.21	−0.19±0.14	−0.75±0.16	−0.30±0.14	−0.31±0.14
BLEU	+0.71±0.15	−1.96±0.26	−1.97±0.24	+0.56±0.15	−0.65±0.15	−0.17±0.14	−0.33±0.14
BERTScore	−4.50±0.46	−2.71±0.33	−0.74±0.16	+2.71±0.30	−0.69±0.15	+0.00±0.14	−0.85±0.16
LLM-J / Claude-Haiku-4.5	+0.21±0.14	+0.29±0.16	+0.61±0.15	+0.41±0.14	+0.24±0.14	+0.31±0.14	+0.45±0.15
LLM-J / Claude-Sonnet-4.5	+0.23±0.14	+0.33±0.16	+0.38±0.14	+0.41±0.14	+0.00±0.14	+0.02±0.14	+0.03±0.14
LLM-J / GPT-5-mini	+0.15±0.14	+0.34±0.16	+0.24±0.14	+0.12±0.14	+0.01±0.14	+0.21±0.14	+0.08±0.14
LLM-J / GPT-4o-mini	+0.65±0.15	+1.26±0.20	+0.77±0.16	+0.78±0.16	+0.45±0.15	+0.43±0.14	+0.41±0.14

BIBLIOGRAPHY

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.

Arpit Agarwal, Debmalya Mandal, David C Parkes, and Nisarg Shah. Peer prediction with heterogeneous users. In Proceedings of the 2017 ACM Conference on Economics and Computation, pages 81–98. ACM, June 2017.

Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. Journal of the Royal Statistical Society: Series B (Methodological), 28(1):131–142, 1966.

Chittaranjan Andrade. Mean difference, standardized mean difference (smd), and their use in meta-analysis: as simple as it gets. The Journal of clinical psychiatry, 81(5):11349, 2020.

Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. Extractive opinion summarization in quantized transformer spaces. Transactions of the Association for Computational Linguistics, 9:277–293, 2021.

Itai Arieli, Yakov Babichenko, and Rann Smorodinsky. Robust forecast aggregation. Proceedings of the National Academy of Sciences, 115(52):E12135–E12143, 2018.

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.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. arXiv preprint arXiv:2112.00861, 2021.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, et al. Benchmarking foundation models with language-model-as-an-examiner. Advances in Neural Information Processing Systems, 36, 2024.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In Proceedings of the European conference on computer vision (ECCV), pages 456–473, 2018.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. arXiv preprint arXiv:1801.04062, 2018.

David Blackwell. Comparison of experiments. In Proceedings of the second Berkeley symposium on mathematical statistics and probability, volume 2, pages 93–103. University of California Press, 1951.

David Blackwell. Equivalent comparisons of experiments. The annals of mathematical statistics, pages 265–272, 1953.

Leonard Boussioux, Jacqueline N Lane, Miaomiao Zhang, Vladimir Jacimovic, and Karim R Lakhani. The crowdless future? how generative ai is shaping the future of human crowdsourcing. The Crowdless Future, 2023.

Glenn W Brier. Verification of forecasts expressed in terms of probability. Monthly weather review, 78(1):1–3, 1950.

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.

Peng Cao, Yilun Xu, Yuqing Kong, and Yizhou Wang. Max-mig: an information theoretic approach for joint learning from crowds. arXiv preprint arXiv:1905.13436, 2019.

Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. Art or artifice? large language models and the false promise of creativity. In Proceedings of the CHI Conference on Human Factors in Computing Systems, pages 1–34, 2024.

Jenny J Chen, Natala J Menezes, Adam D Bradley, and T North. Opportunities for crowdsourcing research on amazon mechanical turk. Interfaces, 5(3):1, 2011.

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.

François Chollet. On the measure of intelligence. arXiv preprint arXiv:1911.01547, 2019.

Kenneth Church and Patrick Hanks. Word association norms, mutual information, and lexicography. Computational linguistics, 16(1):22–29, 1990.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

J Cohen. Statistical power analysis for the behavioral sciences. 1988.

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

Thomas M Cover and Joy A Thomas. Elements of information theory. John Wiley & Sons, 1999.

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.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. Journal of the Royal Statistical Society: Series C (Applied Statistics), 28(1):20–28, 1979.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Benchmark probing: Investigating data leakage in large language models. In NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

Tjibbe Donker. The dangers of using large language models for peer review. The Lancet Infectious Diseases, 2023.

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.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. Transactions of the Association for Computational Linguistics, 9:391–409, 2021.

Boi Faltings, Radu Jurca, and Goran Radanovic. Peer truth serum: incentives for crowd-sourcing measurements and opinions. arXiv preprint arXiv:1704.05269, 2017.

Yuchen Fan, Chen Ling, Xin Zhong, Shuo Zhang, Heng Zhou, Yuchen Zhang, Mingyu Liang, Chengxing Xie, Ermo Hua, Zhizhou He, et al. Lfqa-e: Carefully benchmarking long-form qa evaluation. In The Fourteenth International Conference on Learning Representations, 2026.

Robert M Fano and David Hawkins. Transmission of information: A statistical theory of communications. American Journal of Physics, 29(11):793–794, 1961.

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.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. arXiv preprint arXiv:2302.04166, 2023.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. Bias and fairness in large language models: A survey. Computational linguistics, 50(3):1097–1179, 2024.

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.

Catherine A Gao, Frederick M Howard, Nikolay S Markov, Emma C Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T Pearson. Comparing scientific abstracts generated by chatgpt to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. BioRxiv, pages 2022–12, 2022.

Xi Alice Gao, James R. Wright, and Kevin Leyton-Brown. Incentivizing evaluation with peer prediction and limited access to ground truth. Artificial Intelligence, 275:618–638, 2019. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2019.03.004. URL https://www.sciencedirect.com/science/article/pii/S0004370219301559.

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.

Naman Goel and Boi Faltings. Deep bayesian trust: A dominant and fair incentive mechanism for crowd. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 1996–2003, 2019.

Shahriar Golchin and Mihai Surdeanu. Time travel in llms: Tracing data contamination in large language models. arXiv preprint arXiv:2308.08493, 2023.

Alexander Goldberg, Ivan Stelmakh, Kyunghyun Cho, Alice Oh, Alekh Agarwal, Danielle Belgrave, and Nihar B Shah. Peer reviews of peer reviews: A randomized controlled trial and other experiments. arXiv preprint arXiv:2311.09497, 2023.

Irving John Good. Rational decisions. Journal of the Royal Statistical Society: Series B (Methodological), 14(1):107–114, 1952.

Yongkang Guo, Jason D Hartline, Zhihuan Huang, Yuqing Kong, Anant Shah, and Fang-Yi Yu. Algorithmic robust forecast aggregation. arXiv preprint arXiv:2401.17743, 2024.

Deepak Gupta, Davis Bartels, and Dina Demner-Fushman. a dataset of medical questions paired with automatically generated answers and evidence-supported references. Scientific Data, 12(1):1035, 2025.

Jason D Hartline, Liren Shan, Yingkai Li, and Yifan Wu. Optimal scoring rules for multi-dimensional effort. arXiv preprint arXiv:2211.03302, 2022.

Arlo D Hendrickson and Robert J Buehler. Proper scores for probability forecasters. The Annals of Mathematical Statistics, 42(6):1916–1921, 1971.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300, 2020.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670, 2018.

Mohammad Hosseini and Serge Horbach. Fighting reviewer fatigue or amplifying bias? considerations and recommendations for use of chatgpt and other large language models in scholarly peer review. Research Integrity and Peer Review, 8, 05 2023. doi: 10.1186/s41073-023-00133-5.

Radu Jurca and Boi Faltings. Enforcing truthful strategies in incentive compatible reputation mechanisms. In International Workshop on Internet and Network Economics, pages 268–277. Springer, 2005.

Pei Ke, Bosi Wen, Zhuoer Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, et al. Critiquellm: Scaling llm-as-critic for effective and explainable evaluation of large language model generation. arXiv preprint arXiv:2311.18702, 2023.

Yuqing Kong. Dominantly Truthful Multi-task Peer Prediction with a Constant Number of Tasks, pages 2398–2411. doi: 10.1137/1.9781611975994.147. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611975994.147.

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 2024a. ISSN 0004-5411. doi: 10.1145/3638239. URL https://doi.org/10.1145/3638239.

Yuqing Kong. Dominantly truthful peer prediction mechanisms with a finite number of tasks. Journal of the ACM, 71(2):1–49, 2024b.

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

Yuqing Kong and Grant Schoenebeck. Water from two rocks: Maximizing the mutual information. In Proceedings of the 2018 ACM Conference on Economics and Computation, pages 177–194, 2018b.

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.

Yuqing Kong, Yunqi Li, Yubo Zhang, Zhihuan Huang, and Jinzhao Wu. Eliciting thinking hierarchy without a prior. Advances in Neural Information Processing Systems, 35:13329–13341, 2022.

Ilia Kuznetsov, Osama Mohammed Afzal, Koen Dercksen, Nils Dycke, Alexander Goldberg, Tom Hope, Dirk Hovy, Jonathan K Kummerfeld, Anne Lauscher, Kevin Leyton-Brown, et al. What can natural language processing do for peer review? arXiv preprint arXiv:2405.06563, 2024.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7:453–466, 2019.

Nathan Lambert. Llama 2: an incredible open llm, 2023. URL https://www.interconnects.ai/p/llama-2-from-meta?sd=pf.

Yingkai Li, Jason D Hartline, Liren Shan, and Yifan Wu. Optimization of scoring rules. In Proceedings of the 23rd ACM Conference on Economics and Computation, pages 988–989, 2022.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. Transactions on Machine Learning Research, 2022.

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.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out, pages 74–81, 2004.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers), pages 3214–3252, 2022.

Ryan Liu and Nihar B. Shah. Reviewergpt? an exploratory study on using large language models for paper reviewing, 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.

Yuxuan Lu and Yuqing Kong. Calibrating "cheap signals" in peer review without a prior. Advances in Neural Information Processing Systems, 36, 2024.

Yuxuan Lu, Shengwei Xu, Yichi Zhang, Yuqing Kong, and Grant Schoenebeck. Eliciting informative text evaluations with large language models. arXiv preprint arXiv:2405.15077, 2024.

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

Ani Nenkova and Rebecca J Passonneau. Evaluating content selection in summarization: The pyramid method. In Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004, pages 145–152, 2004.

Eric Neyman, Georgy Noarov, and S Matthew Weinberg. Binary scoring rules that incentivize precision. In Proceedings of the 22nd ACM Conference on Economics and Computation, pages 718–733, 2021.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. IEEE Transactions on Information Theory, 56(11):5847–5861, 2010.

Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. arXiv preprint arXiv:1907.07355, 2019.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B Hashimoto. Proving test set contamination in black box language models. arXiv preprint arXiv:2310.17623, 2023.

Vishakh Padmakumar and He He. Unsupervised extractive summarization using pointwise mutual information. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 2505–2512, 2021.

Asa B Palley and Jack B Soll. Extracting the wisdom of crowds when information is shared. Management Science, 65(5):2291–2309, 2019.

Yuqi Pan, Zhaohua Chen, and Yuqing Kong. Robust decision aggregation with second-order information. arXiv preprint arXiv:2311.14094, 2023.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations. arXiv preprint arXiv:2404.13076, 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pages 311–318, 2002.

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

Dražen Prelec, H Sebastian Seung, and John McCoy. A solution to the single-question crowd wisdom problem. Nature, 541(7638):532–535, 2017.

Goran Radanovic and Boi Faltings. A robust bayesian truth serum for non-binary signals. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, AAAI'13, page 833–839. AAAI Press, 2013.

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.

Goran Radanovic, Boi Faltings, and Radu Jurca. Incentives for effort in crowdsourcing using the peer truth serum. ACM Transactions on Intelligent Systems and Technology (TIST), 7(4):1–28, 2016.

Vyas Raina, Adian Liusie, and Mark Gales. Is llm-as-a-judge robust? investigating universal adversarial attacks on zero-shot llm assessment. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 7499–7517, 2024.

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

Zachary Robertson. Gpt4 is slightly helpful for peer-review assistance: A pilot study, 2023.

Zachary Robertson and Sanmi Koyejo. Let's measure information step-by-step: Llm-based evaluation beyond vibes. arXiv preprint arXiv:2508.05469, 2025.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. arXiv preprint arXiv:2310.18018, 2023.

Michele Salvagno, Fabio Silvio Taccone, Alberto Giovanni Gerli, et al. Can artificial intelligence help for scientific writing? Critical care, 27(1):1–5, 2023.

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 Trans. Econ. Comput., 10(4), feb 2023a. ISSN 2167-8375. doi: 10.1145/3565560. URL https://doi.org/10.1145/3565560.

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, 2023b.

Grant Schoenebeck, Fang-Yi Yu, and Yichi Zhang. Information elicitation from rowdy crowds. In Proceedings of the Web Conference 2021, WWW '21, page 3974–3986, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3449840. URL https://doi.org/10.1145/3442381.3449840.

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.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. Nature, 631(8022):755–759, Jul 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07566-y. URL https://doi.org/10.1038/s41586-024-07566-y.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, et al. A roadmap to pluralistic alignment. arXiv preprint arXiv:2402.05070, 2024.

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

Cecilia Superchi, José Antonio González, Ivan Solà, Erik Cobo, Darko Hren, and Isabelle Boutron. Tools used to assess the quality of peer review reports: a methodological systematic review. BMC medical research methodology, 19:1–14, 2019.

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

Mike Thelwall. Can chatgpt evaluate research quality? Journal of Data and Information Science, 2024.

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.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

Susan Van Rooyen, Nick Black, and Fiona Godlee. Development of the review quality instrument (rqi) for assessing peer reviews of manuscripts. Journal of clinical epidemiology, 52(7):625–629, 1999.

MATTEO VENANZI, William Teacy, Alexander Rogers, and Nicholas Jennings. Weather sentiment - amazon mechanical turk dataset, 2015. URL https://eprints.soton.ac.uk/376543/.

Jeroen PH Verharen. *ChatGPT* identifies gender disparities in scientific peer review. eLife, 12:RP90230, nov 2023. ISSN 2050-084X. doi: 10.7554/eLife.90230. URL https://doi.org/10.7554/eLife.90230.

Luis Von Ahn and Laura Dabbish. Labeling images with a computer game. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 319–326, 2004.

Luis Von Ahn and Laura Dabbish. Designing games with a purpose. Communications of the ACM, 51(8):58–67, 2008.

Bo Waggoner and Yiling Chen. Output agreement mechanisms and common knowledge. In Proceedings of the AAAI Conference on Human Computation and Crowdsourcing, volume 2, pages 220–226, 2014.

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.

James R Wright, Chris Thornton, and Kevin Leyton-Brown. Mechanical ta: Partially automated high-stakes peer grading. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education, pages 96–101, 2015.

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.

Shengwei Xu, Yuxuan Lu, Grant Schoenebeck, and Yuqing Kong. Benchmarking llms' judgments with no gold standard. In The Thirteenth International Conference on Learning Representations, 2025.

Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. Advances in neural information processing systems, 32, 2019.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: Evaluating generated text as text generation. Advances in Neural Information Processing Systems, 34:27263–27277, 2021.

Hedayat Zarkoob, Hu Fu, and Kevin Leyton-Brown. Report-sensitive spot-checking in peer-grading systems. arXiv preprint arXiv:1906.05884, 2019.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? arXiv preprint arXiv:1905.07830, 2019.

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.

Dun Zhang. stella_en_400m_v5. https://huggingface.co/dunzhang/stella_en_400M_v5, 2024.

Peter Zhang and Yiling Chen. Elicitability and knowledge-free elicitation with peer prediction. In Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, pages 245–252, 2014.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675, 2019.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. Benchmarking large language models for news summarization. Transactions of the Association for Computational Linguistics, 12:39–57, 2024.

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.

Yichi Zhang, Shengwei Xu, Grant Schoenebeck, and David Pennock. Stochastically dominant peer prediction. Advances in Neural Information Processing Systems, 38:151632–151664, 2026.

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.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. Advances in Neural Information Processing Systems, 36: 46595–46623, 2023a.

Lianmin Zheng, Ying Sheng, Wei-Lin Chiang, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Chatbot arena: Benchmarking llms in the wild with elo ratings. Blog post, May, 2023b.