# Hierarchical Bayesian Crowdsourcing with Item Difficulty

**Seong Woo Han**[1,†] **Ozan Adıgüzel**[2] **Bob Carpenter**[3]

[1]University of Pennsylvania
[2]Columbia University
[3]Center for Computational Mathematics, Flatiron Institute
[†]Correspondence to seonghan@seas.upenn.edu

## Abstract

In applied statistics and machine learning, the gold standards used for training are often biased and almost always noisy. Dawid and Skene's justifiably popular crowdsourcing model adjusts for rater sensitivity and specificity, but fails to capture distributional properties of rating data gathered for training, which in turn biases training. In this study, we introduce a general purpose measurement-error model with which we can infer consensus categories by adding item-level effects for difficulty, discriminativeness, and guessability. We further show how to constrain the bimodal posterior of these models to avoid adversarial raters. We validate our model's goodness of fit with posterior predictive checks, the Bayesian analogue of $\chi^2$ tests, and assess its predictive accuracy using leave-one-out cross-validation. We illustrate our new model with two well-studied data sets, binary rating data for caries in dental X-rays and implication in natural language.

## 1. Introduction

Crowdsourcing is the process of soliciting ratings (labels, annotations, codes) from a group of respondents for a set of items and combining them for downstream tasks such as training a neural network classifier or a large language model. This note focuses on binary ratings arising in two-way classification problems. In practice, such ratings are often noisy, biased, and inconsistent across raters and items. For example, dentists labeling dental X-rays for caries show low agreement and systematic differences in labeling tendencies (Espeland and Handelman, 1989), and annotators in natural language inference datasets exhibit substantial variability (Snow et al., 2008). As a result, heuristic approaches such as majority voting ignore uncertainty and can lead to biased estimates and degraded performance. Several probabilistic models have been proposed to describe the annotation process. The simplest multinomial model Albert and Dodd (2004) assumes equal annotator reliability and leads to equal-weight voting. Dawid and Skene (1979) introduced annotator-specific sensitivity and specificity, providing a widely used framework for modeling rater reliability. Subsequent work incorporates limited item structure; for example, Beigman and Klebanov (2009) model items as easy or regular, with annotators agreeing on easy items. However, these models do not capture continuous variation in item properties such as difficulty, discrimination, and guessing, nor do they address identifiability issues arising from adversarial raters. More recent applications, such as human feedback for fine-tuning large language models Ouyang et al. (2022); Rafailov et al. (2024), further highlight the need for flexible and interpretable models of noisy annotation; additional applications are discussed in Appendix A.

. We provide code at https://github.com/seongwoohan/bayes-crowdsourcing-irt.

In this work we ask how crowdsourcing models can be extended to better capture the heterogeneity of raters and items while remaining identifiable. To answer this question, the main contributions of the paper are: (i) we propose a hierarchical Bayesian measurement-error model for crowdsourced binary ratings with item-level difficulty, discrimination, and guessing effects, (ii) we resolve the bimodality of rating models by introducing a constraint that excludes adversarial raters, improving identifiability while retaining flexibility near the spammy boundary, and (iii) we provide open-source implementations in Stan (Carpenter et al., 2017), enabling use in R, Python, and Julia.

## 2. Hierarchical Bayesian crowdsourcing model

**Generating ratings.** We consider $I$ items and $J$ raters with $N$ binary ratings $y_n \in \{0, 1\}$ in long format, where each rating is associated with an item $\text{item}_n \in \{1, \ldots, I\}$ and rater $\text{rater}_n \in \{1, \ldots, J\}$. Each item $i$ has a latent category $z_i \in \{0, 1\}$ with prevalence $\pi \in (0, 1)$, $z_i \sim \text{Bernoulli}(\pi)$. For estimation, we marginalize over $z_i$ to form the likelihood used in inference; details are in Appendix B.

Ratings are generated conditionally on $z_i$. Each rater $j$ is characterized by sensitivity and specificity parameters $\alpha_j^{\text{sens}}, \alpha_j^{\text{spec}} \in \mathbb{R}$ on the log-odds scale, governing accuracy on positive and negative items, respectively. Each item $i$ is parameterized by difficulty $\beta_i \in \mathbb{R}$, discrimination $\delta_i > 0$, and guessability $\lambda_i \in (0, 1)$. Difficulty shifts the log-odds of a correct response, so higher $\beta_i$ corresponds to harder items; discrimination scales this difference, so higher $\delta_i$ makes responses more sensitive to the gap between rater ability and item difficulty; and guessability $\lambda_i$ captures the probability of a correct response due to guessing, ensuring a nonzero baseline accuracy even for very difficult items. The full, unreduced model follows the item-response theory three-parameter logistic model generalized with sensitivity and specificity (which we denote IRT-3PL, despite the generalization), where the probability that rater $j$ assigns the correct rating to item $i$ is given by

$$c_n \sim \text{bernoulli}\big(\lambda_i + (1 - \lambda_i)\,\text{logit}^{-1}\big(\delta_i(\alpha_j^k - \beta_i)\big)\big)\,, \tag{1}$$

where $k = \text{sens}$ if $z_i = 1$ and $k = \text{spec}$ if $z_i = 0$. To obtain a distribution over observed ratings, the probability is flipped for negative items so that high accuracy corresponds to a high probability of the correct label. Thus, the rating is given by

$$y_n \sim \begin{cases} \text{Bernoulli}\left(\lambda_i + (1 - \lambda_i)\ \text{logit}^{-1}\left(\delta_i(\alpha_j^{\text{sens}} - \beta_i)\right)\ \right), & \text{if } z_i = 1, \\ \text{Bernoulli}\left(1 - (\lambda_i + (1 - \lambda_i)\ \text{logit}^{-1}\left(\delta_i(\alpha_j^{\text{spec}} - \beta_i)\right)\ )\ \right), & \text{if } z_i = 0. \end{cases} \tag{2}$$

The second case ($z_i = 0$) reduces to the expression $\text{bernoulli}\big((1 - \lambda_i) \cdot \big(1 - \text{logit}^{-1}\big(\delta_i \cdot \big(\alpha_j^{\text{spec}} - \beta_i\big)\big)\big)\big)$.

**Rater model and priors.** A spammy rater is one for whom the rating does not depend on the item being rated. This occurs when $\alpha_j^{\text{sens}} = -\alpha_j^{\text{spec}}$, so that on the probability scale $\text{logit}^{-1}(\alpha_j^{\text{sens}}) = 1 - \text{logit}^{-1}(\alpha_j^{\text{spec}})$ Passonneau and Carpenter (2014). For example, a rater with 40% sensitivity and 60% specificity has a 40% chance of returning a positive rating regardless of the true category.

An adversarial rater is one for which $\alpha_j^{\text{sens}} < -\alpha_j^{\text{spec}}$, implying sensitivity is less than one minus specificity. Such raters are systematically more likely to be incorrect than correct. For example, raters with sensitivity and specificity below 50% are adversarial, whereas those above 50% are informative. To ensure identifiability and avoid symmetric likelihood contributions, we exclude adversarial behavior by reparameterizing sensitivity while retaining flexibility near the spammy boundary,

$$\alpha_j^{\text{sens}} = -\alpha_j^{\text{spec}} + \log(1 + \exp(\eta_j)), \tag{3}$$

where $\eta_j \in \mathbb{R}$. This ensures $\alpha_j^{\text{sens}} > -\alpha_j^{\text{spec}}$ for all raters, while allowing raters arbitrarily close to the spammy regime. We complete the model by placing weakly informative priors on all parameters to regularize the scale of the parameters and aid identifiability, as recommended by Gabry et al.

(2019) and Gelman et al. (2020). For item-level parameters, $\pi \sim \text{Beta}(2,2)$, $\beta_i \sim \text{Normal}(0,1)$, $\delta_i \sim \text{LogNormal}(0, 0.25)$, and $\lambda_i \sim \text{Beta}(2,2)$. To model heterogeneity across raters, we place a hierarchical bivariate normal prior on $(\eta_j, \alpha_j^{\text{spec}})$, where $\eta_j$ controls the distance from the spammy boundary,

$$\begin{pmatrix} \eta_j \\ \alpha_j^{\text{spec}} \end{pmatrix} \sim \text{MVN}(\mu_\alpha,\ \text{diag}(\tau_\alpha)\,\Omega_\alpha\,\text{diag}(\tau_\alpha))\,, \tag{4}$$

with $\mu_\alpha \sim \text{Normal}(0,5)$, $\tau_\alpha \sim \text{LogNormal}(0,1)$, and $\Omega_\alpha \sim \text{LKJ}(2)$. For posterior computation, we use the equivalent non-centered reparameterization Papaspiliopoulos et al. (2007) to decouple latent variables from their scales and mitigates Neal's funnel Neal (2003) for faster and more stable convergence:

$$\begin{pmatrix} \eta_j \\ \alpha_j^{\text{spec}} \end{pmatrix} = \mu_\alpha + \text{diag}(\tau_\alpha) L_{\Omega_\alpha} z_j, \quad z_j \sim \mathcal{N}(\mathbf{0}, I_2). \tag{5}$$

**Model reductions.** By tying or fixing parameters, the full (IRT 3PL + sens/spec) model induces a range of natural submodels, including standard IRT variants and the Dawid–Skene model. Table 1 summarizes the reduction structure via parameter constraints. We consider selected reductions of the full model. Models D, BD, and ABD correspond to tied sensitivity and specificity (i.e., $\alpha^{\text{spec}} = \alpha^{\text{sens}}$), while the remaining models allow separate sensitivity and specificity parameters. Additional model reductions are provided in Appendix C.

| Tag | Reduction | Description |
|---|---|---|
| A | $\lambda_i = 0$ | no guessing items |
| B | $\delta_i = 1$ | equal discrimination items |
| C | $\beta_i = 0$ | equal difficulty items |
| D | $\alpha^{\text{spec}} = \alpha^{\text{sens}}$ | equal error raters |
| E | $\alpha_i = \alpha_j$ | identical raters |

**Table 1.** Orthogonal model reductions.

| Reductions | Probability Correct | Note |
|---|---|---|
| | $\lambda_i + (1-\lambda_i)\,\text{logit}^{-1}(\delta_i(\alpha_j^k - \beta_i))$ | IRT 3PL + sens/spec |
| $A$ | $\text{logit}^{-1}(\delta_i(\alpha_j^k - \beta_i))$ | IRT 2PL + sens/spec |
| $AB$ | $\text{logit}^{-1}(\alpha_j^k - \beta_i)$ | IRT 1PL + sens/spec |
| $ABC$ | $\text{logit}^{-1}(\alpha_j^k)$ | Dawid–Skene |
| $D$ | $\lambda_i + (1-\lambda_i)\,\text{logit}^{-1}(\delta_i(\alpha_j - \beta_i))$ | IRT 3PL |
| $BD$ | $\lambda_i + (1-\lambda_i)\,\text{logit}^{-1}(\alpha_j - \beta_i)$ | IRT 2PL |
| $ABD$ | $\text{logit}^{-1}(\alpha_j - \beta_i)$ | IRT 1PL |

(6)

## 3. Training on probabilistic data

Rating models produce posterior probabilities $\Pr[y_n = 1]$ rather than hard labels. Using these probabilities directly for training preserves uncertainty and avoids bias introduced by collapsing to binary gold standard labels. Consider a classification setting with covariates $x_n \in \mathbb{R}^L$ and binary outcomes $y_n \in \{0,1\}$, modeled via logistic regression, $y_n \sim \text{bernoulli}\big(\text{logit}^{-1}(x_n \cdot \beta)\big)$. In practice, $y_n$ is unobserved and replaced by an estimate of $\Pr[y_n = 1]$, whereas standard approaches instead use hard labels obtained by thresholding or majority vote.

To evaluate the impact of probabilistic training, we conduct a synthetic experiment (details in Appendix D). The resulting error norms in estimating $\beta$ are shown in Figure 1. The plot shows that training with the probabilities is much better than taking the category with the highest probability. The best approach is training a linear regression based on the log odds, with the noisy version of the same approach not far behind. Weighted training with logistic regression is not quite as good. Randomly selecting a category according to the generative model is not as good as using the weights directly, but it still dominates taking the best category.

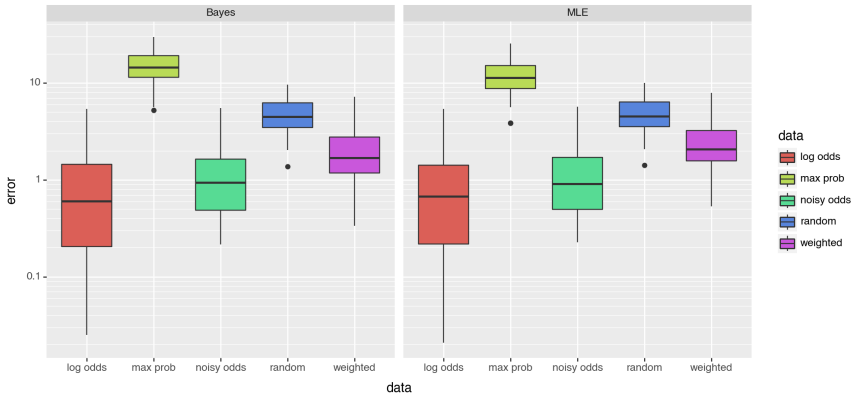


**Figure 1.** The L2 norm of parameter estimation error, $||\widehat{\theta} - \theta||_2$, for different approaches to training with probabilities (lower is better): (log odds) training a linear regression on the log odds, (max prob) assigning the highest probability category, (noisy odds) add standard normal noise to the log odds approach, (random) assign a random category according to the probability, (weighted) train a weighted logistic regression. Estimates and variability are consistent between Bayesian posterior means with a normal prior (left) or ridge-penalized maximum likelihood estimates (right). Regression is 32-dimensional, with correlated inputs and 1024 training data points. Results show standard bar-and-whisker plots over 32 trials with paired random $x, \beta$.

## 4. Model evaluation

The posteriors of all 7 distinct models were estimated using the No-U-Turn sampler, an adaptive Hamiltonian Monte Carlo method (Hoffman et al., 2014; Betancourt, 2017), and evaluated on two datasets using posterior predictive checks (PPC) for goodness-of-fit and leave-one-out cross-validation (LOO) for predictive accuracy (Gelman et al., 1996; Vehtari et al., 2017) (details in Appendix E). The first consists of 5 dentists rating roughly 4000 dental X-rays for caries (Espeland and Handelman, 1989), and the second involves nearly 200 Mechanical Turkers rating subsets of roughly 3000 sentence pairs for entailment (Snow et al., 2008). Models were implemented in Stan (v2.33) and fit using CmdStanPy (v1.20) with default settings. All runs used four chains with 1000 warmup and 1000 sampling iterations, and achieved split-$\widehat{R} < 1.01$ for all parameters, indicating convergence (Gelman et al., 2013). Figure 2 shows posterior predictive $p$-values and $\text{elpd}_{\text{loo}}$ for all models, where values are summarized by their mean and standard deviation for posterior predictive checks and by standard errors for leave-one-out cross-validation, with error bars reflecting simulation variability and estimation uncertainty, respectively. Posterior predictive $p$-values assess how well model-simulated ratings match the observed data, with values near 0.5 indicating good fit. Among models that pass these checks, $\text{elpd}_{\text{loo}}$ provides a measure of predictive accuracy, where higher values indicate better performance. Results for the full set of 18 model reductions (Appendix C) are provided in Appendix G.

For the caries data, the IRT-2PL + sens/spec provides the best overall performance, achieving the highest $\text{elpd}_{\text{loo}}$. The IRT-1PL + sens/spec and IRT-3PL + sens/spec models perform similarly but slightly worse in predictive accuracy. Posterior predictive checks reinforce this pattern. While all sens/spec models provide reasonable fits, the IRT-3PL + sens/spec model exhibits increased variability, suggesting that the additional guessing parameter introduces complexity and is not well supported by the data. In contrast, models without sensitivity and specificity (i.e., standard IRT 1PL–3PL) perform substantially worse in both posterior predictive fit and predictive accuracy. The Dawid–Skene model, in particular, fails to capture the ratings distribution adequately, with poor posterior predictive behavior for rating-level summaries. Overall, these results demonstrate that explicitly modeling rater sensitivity and specificity is critical, providing an improvement over both classical IRT models and Dawid–Skene. For the Mechanical Turk data, the IRT-2PL + sens/spec

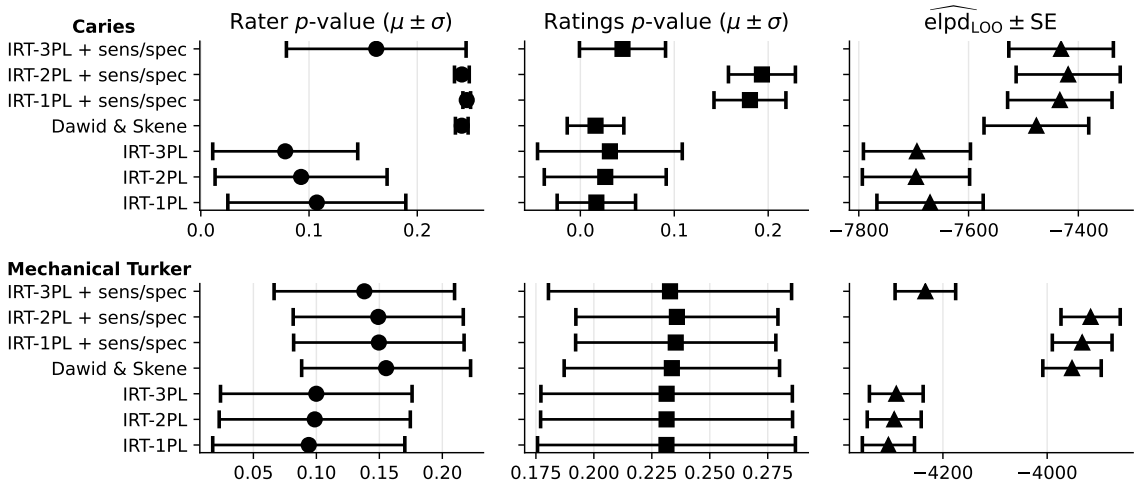


**Figure 2.** Posterior predictive $p$-values and leave-one-out cross-validation $\text{elpd}_{\text{loo}}$ across Caries and Mechanical Turk datasets for 7 models. Posterior predictive values are summarized by mean and standard deviation, while $\text{elpd}_{\text{loo}}$ values are reported with standard errors.

again achieves the best predictive performance in terms of $\text{elpd}_{\text{loo}}$. Posterior predictive checks show that the rater $p$-values are broadly similar across the IRT models with sensitivity and specificity and the Dawid–Skene model. However, models incorporating sensitivity and specificity consistently outperform those without it in predictive accuracy. None of the models provide a strong fit for the ratings $p$-values. The main difference between the Caries and Mechanical Turk datasets is that there are many more Turkers providing far fewer ratings each, and there are more spammy or low-quality raters among the Turkers, which makes the inference problem substantially noisier.

## 5. Dawid and Skene's posterior predictions underestimate dispersion

To demonstrate how posterior predictions underestimate dispersion, we compare Dawid and Skene's baseline with three item-level models: difficulty, plus discrimination, and plus guessing with sensitivity and specificity in Amazon Turker data. The goodness-of-fit tests are similar, so in Figure 3, we plot the expected number of marginal positive votes for each model along with the actual data. The Dawid and Skene model has an inflated number of middling positive votes compared to the data; in contrast the IRT-3PL + sens/spec model underestimates them. However, posterior predictive inference for IRT-2PL + sens/spec and IRT-1PL + sens/spec more closely matches the actual data. This alignment enhances the model's ability to reflect true voter consensus, thereby producing a distribution of votes that more accurately mirrors

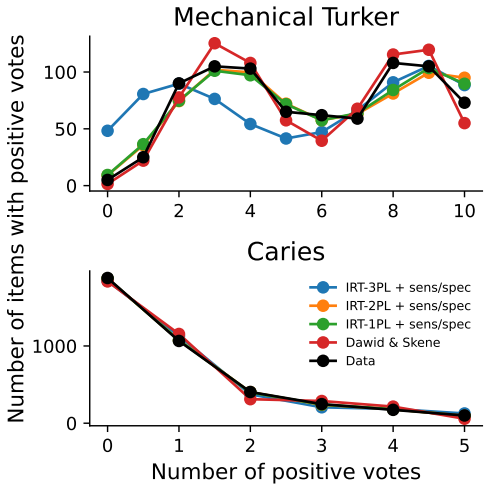


**Figure 3.** Distribution of positive votes per item, demonstrating the varying levels of dispersion captured by each model.

observed voting patterns. For the caries data, the story is similar. IRT-1,2,3PL + sens/spec closely match the actual data, while the Dawid and Skene model again inflates the number of middling positive votes compared to the real data.

## 6. Conclusion

Until this work, crowdsourcing models for categorical responses, such as Dawid–Skene (Dawid and Skene, 1979), do not account for item-level variation or structured rater behavior, leading to systematic misfit. Our results demonstrate this through posterior predictive checks and leave-one-out cross-validation, where Dawid–Skene fails to capture observed rating patterns. By incorporating item difficulty, discrimination, and guessability within a hierarchical Bayesian framework, our models achieve improved fit and predictive performance. Explicit modeling of rater sensitivity and specificity further yields more accurate and well-calibrated predictions. To ensure identifiability, we restrict to cooperative solutions, excluding adversarial modes where raters perform below chance, which are less realistic in typical crowdsourcing settings (Passonneau and Carpenter, 2014). Potential extensions of the current framework are discussed in Appendix F. The bottom line is that most training efforts in machine learning are not making the most of their human feedback and can be improved by applying the structured probabilistic crowdsourcing models.

## References

ACT, Inc. ACT [American College Testing] technical manual. Technical report, ACT, Inc., 2024. URL https://www.act.org/content/dam/act/unsecured/documents/ACT_Technical_Manual.pdf.

Paul S Albert and Lori E Dodd. A cautionary note on the robustness of latent class models for estimating diagnostic error without a gold standard. *Biometrics*, 60(2):427–435, 2004.

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008.

Eyal Beigman and Beata Beigman Klebanov. Learning with annotation noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 280–287, 2009.

Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus A Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76, 2017.

Xi Chen, Paul N Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 193–202, 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.

Mark A Espeland and Stanley L Handelman. Using latent class models to characterize and assess relative error in discrete measurements. *Biometrics*, pages 587–599, 1989.

Kevin S Flannagan, Brian M Mills, and Robert L Goldstone. The psychophysics of home plate umpire calls. *Scientific Reports*, 14(1):2735, 2024.

Anton K Formann. Latent class model diagnostics—a review and some proposals. *Computational Statistics & Data Analysis*, 41(3-4):549–559, 2003.

Jonah Gabry, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. Visualization in Bayesian workflow. *Journal of the Royal Statistical Society, Series A: Statistics in Society*, 182 (2):389–402, 2019.

Andrew Gelman and Jennifer Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press, 2007.

Andrew Gelman, Xiao-Li Meng, and Hal Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, pages 733–760, 1996.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, Third edition, 2013.

Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. Bayesian workflow. *arXiv preprint arXiv:2011.01808*, 2020.

Matthew D Hoffman, Andrew Gelman, et al. The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

Balaji Lakshminarayanan and Yee Whye Teh. Inferring ground truth from multi-annotator ordinal data: a probabilistic approach. *arXiv*, 1305.0015, 2013.

R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 1959.

Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282, 2012.

Radford M Neal. Slice sampling. *The annals of statistics*, 31(3):705–767, 2003.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Keton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Omiros Papaspiliopoulos, Gareth O Roberts, and Martin Sköld. A general framework for the parametrization of hierarchical models. *Statistical Science*, pages 59–73, 2007.

Rebecca J Passonneau and Bob Carpenter. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, 2:311–326, 2014.

Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. Comparing Bayesian models of annotation. *Transactions of the Association for Computational Linguistics*, 6:571–585, 2018.

Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of machine learning research*, 11(4), 2010.

Simon Rogers, Mark Girolami, and Tamara Polajnar. Semi-parametric analysis of multi-rater data. *Statistics and Computing*, 20:317–334, 2010.

Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. Corpus annotation through crowdsourcing: Towards best practice guidelines. In *The International Conference on Language Resources and Evaluation*, pages 859–866, 2014.

Hagit Shatkay, Andrey Rzhetsky, and W John Wilbur. Searching for high-utility text in the biomedical literature. *BioLINK SIG: Linking Literature, Information and Knowledge for Biology*, 2005.

Padhraic Smyth, Usama Fayyad, Michael Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labelling of Venus images. *Advances in neural information processing systems*, 7, 1994.

Rion Snow, Brendan O'connor, Dan Jurafsky, and Andrew Y Ng. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 254–263, 2008.

James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and computing*, 27:1413–1432, 2017.

## A. Applications of crowdsourcing models

There are several tasks to which crowdsourcing models are applied, and for every one of them a rating model improves performance over heuristic baselines. For example, rating models outperform majority voting for category training and outperform indirect measurements like inter-annotator agreement statistics to measure task difficulty (see, e.g., Artstein and Poesio (2008), Sabou et al. (2014), McHugh (2012)).

**Inferring a gold-standard data set.** The first and foremost application of crowdsourcing is to generate a "gold standard" data set, where a single category (or label) is assigned to each item. In terms of generating representative data, it is best to sample data according to its probability (i.e., follow the generative model) rather than to choose the "best" rating for each item according to a heuristic such as highest probability. The second section of this paper shows that it is better for downstream accuracy to train with a probabilistic corpus that retains information about rating uncertainty. In cases where it is impractical to use a probabilistic corpus with weighted training, we show why it is far better to sample labels according to their posterior probability in the rating model than to choose the "best" label. In particular, we will demonstrate that majority voting schemes among raters are suboptimal compared to sampling, which is in turn dominated by training with the probabilities (a kind of Rao-Blackwellization). If a classifier for the data is available, an even better approach is to jointly train a classifier and rating model, as shown by Raykar et al. (2010).

**Inferring population prevalence.** The second most common application of crowdsourcing is to understand the probability of positivity among items in the population represented by the crowdsourcing data. This is particularly common in epidemiology, where the probability of positive outcomes is the prevalence of the disease in the (sub)population (Albert and Dodd, 2004). It can also be used to analyze the prevalence of hate speech on a social media site or bias in televised news, the prevalence of volcanoes on Venus (Smyth et al., 1994), or the prevalence of positive reviews for a restaurant.

**Understanding and improving the coding standard.** The third most common application of crowdsourcing is to understand the coding standard, which is the rubric under which rating is carried out. Traditionally, this has been measured through inter-annotator agreement (Artstein and Poesio, 2008). In contrast, rating models provide finer-grained analysis of rater accuracy in terms of sensitivity and specificity, as well as the information gain expected from a rating by a specific rater (Passonneau and Carpenter, 2014).

**Understanding and improving raters.** What is the mean sensitivity and specificity and how does it vary among raters? Are sensitivity and specificity anticorrelated or correlated in the population? This understanding can be fed back to the raters themselves for ongoing training. For example, American baseball umpires have extensive feedback on how they call balls and strikes as measured against a very accurate machine's call, which has led to much higher accuracy and consistency among umpires Flannagan et al. (2024). Understanding rater populations, such as those available through Mechanical Turk or Upwork, is important when managing raters for multiple crowdsourcing tasks. For example, it is straightforward with a rating model to infer the proportion of spammers and the proportion of high quality raters. The number of raters required for high quality joint ratings may also be assessed (Passonneau and Carpenter, 2014).

**Understanding the items.** A fifth task, which has received relatively little attention in the crowdsourcing literature, is to understand the structure of the population of items. For example, which items are difficult to rate and why? Which items simply have too little signal to be consistently rated? Which items lie near the decision boundary and which are far away? Which items have high discrimination and why? A discriminative item is one which cleanly separates high ability from low ability raters in their ability to rate it correctly. Understanding the items is the primary focus of educational test design, where the items are test questions and the raters are students. Test questions are selected for standardized tests like the ACT (American College Testing) based on having high discrimination and a useful range of difficulties (ACT, Inc., 2024).

## B. Marginal likelihood

A crowdsourcing model generates latent discrete categories $z_i \in \{0, 1\}$ for each item. For both optimization and sampling, it is convenient to marginalize the complete likelihood $p(y, z \mid \pi, \alpha, \beta, \delta, \lambda)$ to the rating likelihood $p(y \mid \pi, \alpha, \beta, \delta, \lambda)$. The marginalization calculation is efficient because it is factored by data item. Letting $\theta = \pi, \alpha, \beta, \delta, \lambda$ be the full set of continuous parameters, the trick is to rearrange the long-form data by item, then marginalize out the discrete parameters, resulting in the likelihood

$$p(y \mid \theta) = \prod_{i=1}^{I} \sum_{z_i=0}^{1} p(z_i \mid \theta) \cdot \prod_{n:\text{item}_n=i} p(y_n \mid z_i, \theta). \tag{7}$$

We start with the complete data likelihood for ratings $y$ and latent categories $z$,

$$p(y, z \mid \theta) = \prod_{i=1}^{I} p(z_i \mid \theta) \cdot \prod_{n=1}^{N} p(y_n \mid z, \theta) \tag{8}$$

and then rearrange terms by item,

$$p(y, z \mid \theta) = \prod_{i=i}^{I} \left( p(z_i \mid \theta) \cdot \prod_{n:\text{item}_n=i} p(y_n \mid z_i, \theta) \right). \tag{9}$$

On a per item basis, the marginalization is tractable, yielding Equation 7. Computational inference requires working on the log scale, where the log marginal likelihood of the rating data in the full model is given by

$$\begin{aligned} \log p(y \mid \theta) &= \log \textstyle\prod_{i=i}^{I} \sum_{z_i=0}^{1} p(z_i \mid \theta) \cdot \prod_{n:\text{item}_n=i} p(y_n \mid z_i, \theta). \\ &= \textstyle\sum_{i=1}^{I} \text{logSumExp}_{z_i=0}^{1} \left( \log p(z_i \mid \theta) + \sum_{n:\text{item}_n=i} \log p(y_n \mid z_i, \theta) \right), \end{aligned} \tag{10}$$

where

$$\text{logSumExp}_{n=1}^{N} \ell_n = \log \sum_{n=1}^{N} \exp(\ell_n) \tag{11}$$

is the numerically stable log-scale analogue of addition.

## C. Full model reductions

**Tied sensitivity and specificity.** First, we consider models which do not distinguish sensitivity and specificity. Such models should be used when the categories are not intrinsically ordered (e.g., rating two consumer brands for preference). All of these models other than the last (ABCDE) assumes raters have varying accuracy.

| Reductions | Probability Correct | Note |
|---|---|---|
| $D$ | $\lambda_i + (1 - \lambda_i) \cdot \text{logit}^{-1}(\delta_i \cdot (\alpha_j - \beta_i))$ | IRT 3PL |
| $CD$ | $\lambda_i + (1 - \lambda_i) \cdot \text{logit}^{-1}(\delta_i \cdot \alpha_j)$ | |
| $BD$ | $\lambda_i + (1 - \lambda_i) \cdot \text{logit}^{-1}(\alpha_j - \beta_i)$ | IRT 2PL |
| $BCD$ | $\lambda_i + (1 - \lambda_i) \cdot \text{logit}^{-1}(\alpha_j)$ | |
| $AD$ | $\text{logit}^{-1}(\delta_i \cdot (\alpha_j - \beta_i))$ | |
| $ACD$ | $\text{logit}^{-1}(\delta_i \cdot \alpha_j)$ | |
| $ABD$ | $\text{logit}^{-1}(\alpha_j - \beta_i)$ | IRT 1PL |
| $ABCD$ | $\text{logit}^{-1}(\alpha_j)$ | |
| $ABCDE$ | $\text{logit}^{-1}(\alpha)$ | |

(12)

**Free sensitivity and specificity.** The following models introduce separate parameters for sensitivity and specificity rather than assuming they are the same. Only the last model (ABCE) does not distinguish rater abilities.

| Reductions | Probability Correct | Note |
|---|---|---|
| | $\lambda_i + (1-\lambda_i) \cdot \text{logit}^{-1}(\delta_i \cdot (\alpha_j^k - \beta_i))$ | IRT 3PL + sens/spec |
| $C$ | $\lambda_i + (1-\lambda_i) \cdot \text{logit}^{-1}(\delta_i \cdot \alpha_j^k)$ | |
| $BC$ | $\lambda_i + (1-\lambda_i) \cdot \text{logit}^{-1}(\alpha_j^k)$ | |
| $A$ | $\text{logit}^{-1}(\delta_i \cdot (\alpha_j^k - \beta_i))$ | IRT 2PL + sens/spec |
| $AC$ | $\text{logit}^{-1}(\delta_i \cdot \alpha_j^k)$ | |
| $AB$ | $\text{logit}^{-1}(\alpha_j^k - \beta_i)$ | IRT 1PL + sens/spec |
| $ABC$ | $\text{logit}^{-1}(\alpha_j^k)$ | Dawid/Skene |
| $ABCE$ | $\text{logit}^{-1}(\alpha^k)$ | |

(13)

**No rater effects.** The last model, which is common in epidemiology (Albert and Dodd, 2004), includes item effects without any rater effects.

| Reductions | Probability Correct |
|---|---|
| $ABDE$ | $\text{logit}^{-1}(-\beta_i)$ |

(14)

**Redundant parameter models.** The remaining thirteen models are redundant in the sense that fixing their non-identifiability issues reduces to a model with a single item effect.

| Reductions | Probability Correct |
|---|---|
| $E$ | $\lambda_i + (1-\lambda_i) \cdot \text{logit}^{-1}(\delta_i \cdot (\alpha^k - \beta_i))$ |
| $DE$ | $\lambda_i + (1-\lambda_i) \cdot \text{logit}^{-1}(\delta_i \cdot (\alpha - \beta_i))$ |
| $CE$ | $\lambda_i + (1-\lambda_i) \cdot \text{logit}^{-1}(\delta_i \cdot \alpha^k)$ |
| $CDE$ | $\lambda_i + (1-\lambda_i) \cdot \text{logit}^{-1}(\delta_i \cdot \alpha)$ |
| $BE$ | $\lambda_i + (1-\lambda_i) \cdot \text{logit}^{-1}(\alpha^k - \beta_i)$ |
| $BDE$ | $\lambda_i + (1-\lambda_i) \cdot \text{logit}^{-1}(\alpha - \beta_i)$ |
| $BCE$ | $\lambda_i + (1-\lambda_i) \cdot \text{logit}^{-1}(\alpha^k)$ |
| $BCDE$ | $\lambda_i + (1-\lambda_i) \cdot \text{logit}^{-1}(\alpha)$ |
| $AE$ | $\text{logit}^{-1}(\delta_i \cdot (\alpha^k - \beta_i))$ |
| $ADE$ | $\text{logit}^{-1}(\delta_i \cdot (\alpha - \beta_i))$ |
| $ACE$ | $\text{logit}^{-1}(\delta_i \cdot \alpha^k)$ |
| $ACDE$ | $\text{logit}^{-1}(\delta_i \cdot \alpha)$ |
| $ABE$ | $\text{logit}^{-1}(\alpha^k - \beta_i)$ |

(15)

## D. Synthetic experiment details

For the experiment, we generate a synthetic data set of covariates $x_i \sim \text{normal}(0, \Sigma)$, where the positive definite covariance matrix is defined by $\Sigma_{m,n} = \rho^{|m-n|}$, with $\rho = 0.9$. The result is highly correlated covariates; uncorrelated covariates show the same trend. We evaluate five approaches to estimation: (max prob) take $y_n$ to be 1 if $\text{logit}^{-1}(x_n \cdot \beta) > \frac{1}{2}$, (log odds) train a linear regression with outcome $\log x_n \cdot \beta$, (noisy odds) log odds with standard normal noise, $\text{logit}^{-1}(x_n \cdot \beta + \epsilon_n)$, with $\epsilon_n \sim \text{normal}(0, 1)$, (random) randomly generate $y_n \sim \text{bernoulli}\big(\text{logit}^{-1}(x_n \cdot \beta)\big)$ according to its probability distribution, and (weighted) train a weighted logistic regression with outcome 1 and weight $\text{logit}^{-1}(x_n \cdot \beta)$ *and* outcome 0 and weight $1 - \text{logit}^{-1}(x_n \cdot \beta)$.

We provide both Bayesian (posterior mean) and frequentist (penalized maximum likelihood) estimates. For the Bayesian setting, we use a standard normal prior $\beta_k \sim \text{normal}(0, 1)$ and in the frequentist setting we use ridge regression, with penalty function $\frac{1}{2} \cdot \beta^\top \cdot \beta$ to match the Bayesian prior.

## E. Evaluation metrics

**Posterior predictive checks** We use posterior predictive checks (PPC) for goodness-of-fit testing (Gelman et al., 1996). They are the Bayesian analogue of the $\chi^2$ tests widely employed in epidemiology (Albert and Dodd, 2004), which test a "null" of the fitted model against the observed data (Formann, 2003). PPCs work by generating replicated data from the fitted model and comparing statistics from the replicated data to those from the original data. The posterior predictive distribution for replications $y^{\text{rep}}$ of the original data $y$ given model parameters $\theta$ is $p(y^{\text{rep}} \mid y) = \mathbb{E}\left[p(y^{\text{rep}} \mid \theta) \mid y\right] = \int p(y^{\text{rep}} \mid \theta) \cdot p(\theta \mid y)\ \mathrm{d}\theta$.

If a model fits well, test statistics $s(\cdot)$ should take similar values in both original and replicated data sets. This is summarized by a Bayesian $p$-value-like statistic, $\Pr[s(y^{\text{rep}}) \geq s(y) \mid y] = \int \mathrm{I}(s(y^{\text{rep}}) \geq s(y)) \cdot p(y^{\text{rep}} \mid y)\ \mathrm{d}y^{\text{rep}}$. Any choice of statistic guided by the quantities of interest in the model itself can be used. We focus on marginal positive votes per rater and per item as statistics, because these directly reflect rater-specific biases (sensitivity and specificity) and item-level difficulty. They are simple to interpret, align with classical $\chi^2$ tests, and effectively reject models that omit difficulty when items show variation in difficulty.

**Leave-one-out cross-validation** We use an accurate approximation of leave-one-out cross-validation (LOO) for predictive accuracy (Vehtari et al., 2017). LOO provides a fine-grained view of predictive performance, especially useful for model comparison and refinement.

LOO estimates out-of-sample predictive fit by evaluating the model's performance on each data point, leaving out one observation at a time. This provides an accurate measure of how well the model generalizes to unseen data. The expected log pointwise predictive density ($\text{elpd}_{\text{loo}}$) is computed as $\text{elpd}_{\text{loo}} = \sum_{i=1}^{n} \log p(y_i \mid y_{-i})$, where $p(y_i \mid y_{-i})$ is the predictive density of observation $y_i$, excluding the $i$-th observation. This involves estimating $p(y_i \mid y_{-i}) = \int p(y_i \mid \theta)\, p(\theta \mid y_{-i}) d\theta$, which can be calculated very efficiently using Pareto-smoothed importance sampling given a single model fit (Vehtari et al., 2017).

## F. Extensions

**Including covariate information** Parameterizing on the log odds scale makes it straightforward to add features (i.e., covariates) in addition to the baseline random effects for items or raters. The simple models presented here can be thought of as intercept-only versions of more general models.

Features for raters might include demographic information such as education, location, native language, age, education level, whether they are an AI and which one, etc. Features for items can be used to inform difficulty, discrimination and guessability. For example, a covariate might indicate the number of options in a multiple choice test, the length of sentences used for inference, or the grade level of the textbook from which the item was culled, when a person's last dental checkup was, etc.

If item-level or rater-level covariates are available, they may be used to inform the parameters in the usual way through a regression in the form of a generalized linear model (Gelman and Hill, 2007). For example, suppose there are item-level covariates $x_i \in \mathbb{R}^K$. With a parameter vector $\gamma \in \mathbb{R}^K$, the generative model for the category of an item may be extended to a logistic regression, $z_i \sim \text{bernoulli}\big(x_i^\top \cdot \gamma\big)$.

**$K$-way categorical rating** A natural extension is to $K$-way categorical ratings, such as classifying a dog image by species, classifying an article in a newspaper by topic, rating a movie on a one to five scale, classifying a doctor's visit with an ICD-10 code, and so on. Most of the work on ratings has been in this more general categorical setting. With more than two categories, sensitivity and specificity are replaced with categorical responses based on the latent true category. Discrimination and guessing act the same way, but difficulty must be replaced with a more general notion of a categorical item level effect, which may represent either focused alternatives (e.g., a border collie is confusable with an Irish shepherd) or diffuse (e.g., can't tell what's in the image).

**Population-level models** With enough raters, these models may also be extended hierarchically to make population-level inferences about the distribution of rater abilities or item difficulties (Paun et al., 2018). Several of the crowdsourcing tasks may be combined to select raters and items to rate online with active learning, which is a form of reinforcement learning. With a hierarchical model, inference may be expanded to new raters (Paun et al., 2018).

**Ordered, count, and other data** It is also straightforward to extend a rating model to ordered responses such as Likert scales (Lakshminarayanan and Teh, 2013; Rogers et al., 2010; Shatkay et al., 2005), rank ordering (Chen et al., 2013; Rafailov et al., 2024), counts (the "textbook" case of crowdsourcing is estimating the number of jelly beans in a container (Surowiecki, 2005)), proportions/probabilities, distances, or pairs of real numbers such as planetary locations (Smyth et al., 1994). All that needs to change is the response model and the representation of the latent truth—the idea of getting noisy ratings and inferring a ground truth remains. As an example, Smyth had raters mark images of Venus for volcano locations (Smyth et al., 1994). The true location is represented as a latitude and longitude and rater responses can be multivariate normal centered around the true, but unknown, location. For ordinal ratings, an ordinal logit model of the truth may be used (Rogers et al., 2010). For comparisons, the Bradley-Terry model can be used (Bradley and Terry, 1952), and for ranking, the Plackett-Luce generalization (Plackett, 1975; Luce, 1959), as used in direct preference optimization for fine-tuning large language models (Rafailov et al., 2024).

**Joint estimation of a classifier** When item-level covariates are available, Raykar et al. (2010) provide an approach to jointly estimating the parameters of the rating model and the classifier. In essence, the prevalence model is updated to a logistic regression classifier and the the classifier participates jointly in rating along with the raters.

## G. Full evaluation results

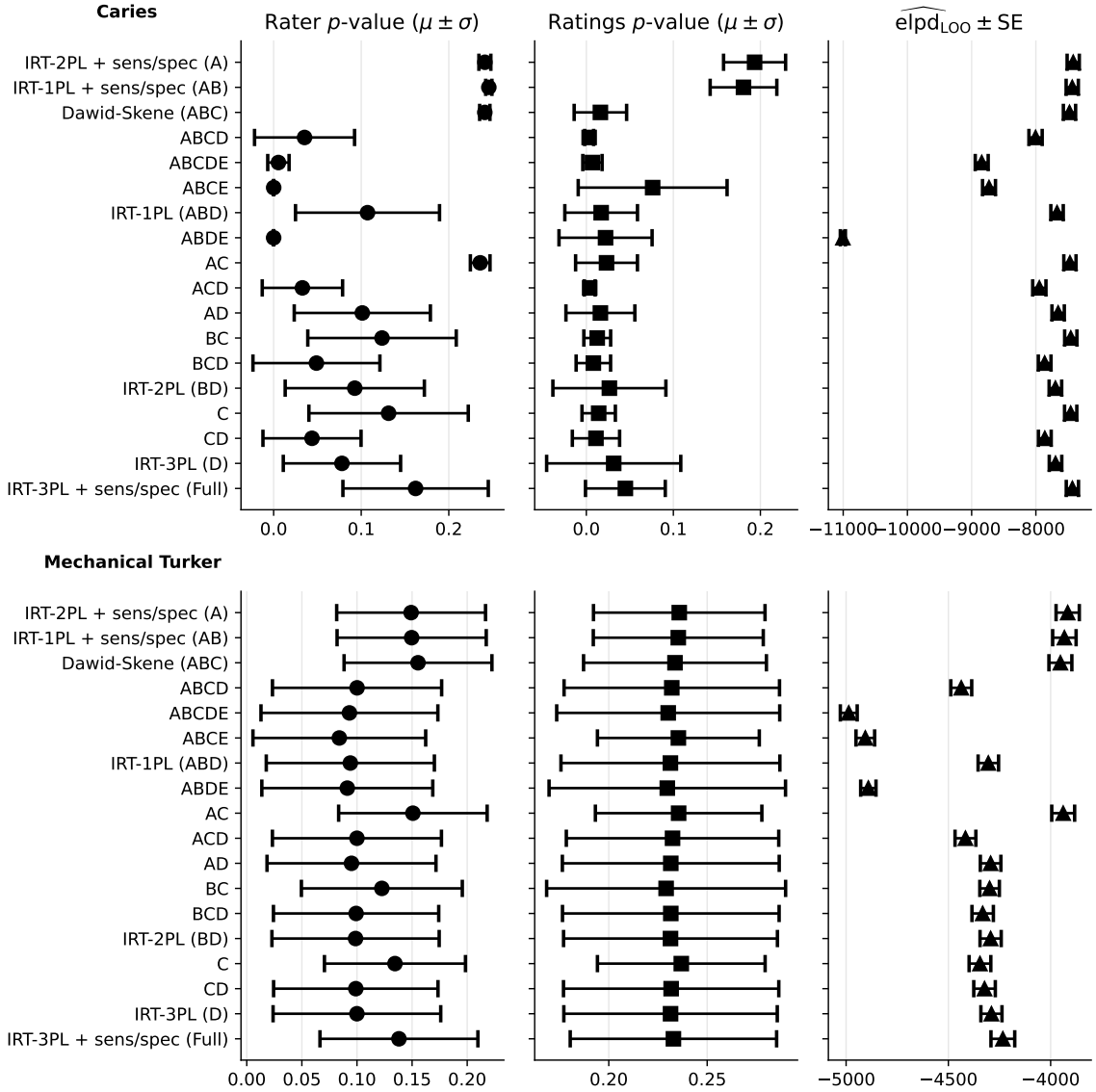


**Figure 4.** Posterior predictive $p$-values and leave-one-out cross-validation $\text{elpd}_{\text{loo}}$ across Caries and Mechanical Turk datasets for different 18 models. Posterior predictive values are summarized by mean and standard deviation, while $\text{elpd}_{\text{loo}}$ values are reported with standard errors.