Follow the leader(board) with confidence: Estimating p-values from a single test set with item and response variance

Shira Wein
Georgetown Univ.[*]
sw1158@georgetown.edu

Christopher M. Homan
Rochester Inst. Tech.
cmhvcs@rit.edu

Lora Aroyo and **Chris Welty**
Google Research
{l.m.aroyo,cawelty}@gmail.com

Abstract

Among the problems with leaderboard culture in NLP has been the widespread lack of confidence estimation in reported results. In this work, we present a framework and simulator for estimating p-values for comparisons between the results of two systems, in order to understand the confidence that one is actually better (i.e. ranked higher) than the other. What has made this difficult in the past is that each system must itself be evaluated by comparison to a gold standard. We define a null hypothesis that each system's *metric scores* are drawn from the same distribution, using variance found naturally (though rarely reported) in test set items and individual labels on an item (responses) to produce the metric distributions. We create a test set that evenly mixes the responses of the two systems under the assumption the null hypothesis is true. Exploring how to best estimate the true p-value from a single test set under different metrics, tests, and sampling methods, we find that the presence of response variance (from multiple raters or multiple model versions) has a profound impact on p-value estimates for model comparison, and that choice of metric and sampling method is critical to providing statistical guarantees on model comparisons.

1 Introduction

AI and NLP evaluation is facing a scientific reproducibility crisis that, despite increasing awareness, continues to worsen (Gundersen and Kjensmo, 2018). Published results may often show only epsilon improvements to state-of-the-art results, with no effort to estimate whether or not the results are statistically significant. The reasons for this crisis are complex, and it is easy to implicate the culture created by leaderboards (e.g., Wang et al. (2018)).

Our work is motivated by the need to **provide statistical testing alongside NLP results in order to reliably demonstrate model improvement**, as opposed to solely depending on leaderboards. Our work naturally ensues from studies of rater response disagreement (see e.g., (R Artstein; Snow et al., 2008; L Aroyo, 2013; Plank et al., 2014; Fornaciari et al., 2022), among others). Further, the issue of insufficient statistical analysis in NLP work is well-documented, with many ACL papers not reporting statistical significance (Dror et al., 2018). Considering the reliance on system comparison for benchmarking and leaderboards, statistical guarantees that consider the performance of both systems are critical, yet understudied.

Statistical tests for paired data (e.g. McNemar (1947)) are not appropriate for this setting because of their reliance on strong assumptions about the data (Dietterich, 1998b); even extensions of McNemar's test such as the Cochran-Mantel-Haenszel test (Mantel, 1963) only apply when the metric can be applied independently to each item or responder (human or machine), and is then aggregated. Therefore, these metrics are not applicable for this use case, in large part due to three potential challenges: (1) three sets of data are involved in this comparison, (2) there is variance in all three of those sets, and (3) many different metrics are used in NLP evaluation. Moreover, variance can come at the item or response level, due to stochastic inference or training, changes in training data such as cross-validation, or annotator disagreement in gold labels.

We investigate the use of null hypothesis significance tests (NHST) to add a dimension of confidence to NLP evaluations. The purpose of NHST is to determine whether differences between multiple sets of observations are significant, after accounting for sampling variance in the observations. When comparing two NLP systems, each is first compared to a gold standard, resulting in some metric score (e.g. BLEU (Papineni et al., 2002)), and then those metric scores for the two models are compared to each other. While all p-values are esti-

[*] Work completed while interning at Google.

mates, there are many ways to sample and measure the results from a single test set, each producing a different p-value estimate. We explore how to determine which method (of sampling, aggregating, and measuring responses) produces the most accurate p-value estimate from a single test set in comparison to the true/ground truth p-value.

In this work, we present a framework for effective p-value estimation when comparing two systems against gold judgments, with the aim of identifying with statistical rigor if one system outperforms the other. Our findings indicate that the amount of response variance has an impact on p-value estimates, item-wise mean absolute error is consistently a reliable metric, and—while most metrics and sampling methods perform well when machine output is dissimilar—metric choice and sampling method is especially critical when the performance of the two machines is similar.

Our primary contributions include:

- combining, for the first time, the related notions of response disagreement from machines (Gorman and Bedrick, 2019) and from raters (Aroyo and Welty, 2015);
- a new framework for NHST that allows comparisons across different test metrics and sampling strategies;
- a simulator capable of producing informative null hypotheses and computing p-values that account for both item and response variance,
- a thorough evaluation of how well eight metrics and six re-sampling strategies estimate the "true" p-value from a single test, on simulated data; and
- a demonstration of our framework on real-world data.

Our findings give insight into which statistics are most informative when designing NHSTs for contemporary NLP systems, and is applicable to any NLP setting that makes use of comparisons to quantitative gold judgments (e.g. sentiment analysis, semantic similarity, search relevance, translation quality, etc.), when response variance is prevalent. We plan to share our code upon publication.

2 Related Work

Our approach to generating a statistical guarantee associated with the comparison of two NLP systems (a p-value) is rooted in the statistical inference method of NHST. Our formulation also incorporates variance in rater and system responses.

2.1 NHST for evaluation

Existing notions of p-values are built on a null hypothesis $\mathcal{H}_0$ which states that the effect size between the control and test set is zero. The p-value is then the probability that an effect of the observed size or greater would occur under the assumption that $\mathcal{H}_0$ is true. Here, the "control" and "test" sets are the outputs of distinct models that we wish to compare, and the effect size represents the performance of the first system compared to the second on gold standard data.

Dietterich (1998a) considers hypothesis testing on machine learning problems (specifically comparing the performance of two learning algorithms with a small amount of data), but does not consider response variance or accuracy of the p-value estimate. Our approach builds on Dietterich's (1998a); we also observe that the standard null hypotheses do not quite fit the use case of comparing the output of two systems, since the error is the result of a comparison with a third, gold standard, dataset, and we investigate the effect of different sources of variance, as well as different metrics, on the p-value estimate from a single test set.

Søgaard et al. (2014) explore the effects of sample size, covariates (such as sentence length), and the variance introduced by multiple metrics, and conclude that current approaches to p-value tests are not reproducible or sufficient. They suggest that the usual upper bound of $p < 0.05$ is too high, and that $p < 0.0025$ provides a better guarantee that the *false positive rate* is less than 5%. One problem faced in coming to this conclusion was how to determine what the correct p-value actually is. Note that they use the false positive rate as the target of the guarantee, which is an intuitive but completely non-standard approach to hypothesis testing. We address this by utilizing a simulator that is capable of generating thousands of test sets, which then allows us to make a better estimate as to the true p-value, and compare the effects of many more sources of variance.

Related work has surveyed statistical significance testing techniques in NLP systems (Dror et al., 2020) and studied permutation and bootstrapping methods for computing significance tests and confidence intervals on text summarization evaluation metrics (Deutsch et al., 2021). Haroush et al. (2021) observe that out of distribution detection can be recast as a p-value problem, using p-values for inference, not significance testing.

Prior work critical of the utility of the p-value cites the impact of sample size and bias on the level of significance (Sullivan and Feinn, 2012; Thiese et al., 2016), as well as the variability of p-values across samples (Halsey et al., 2015).

Kohavi et al. (2022) examine the misunderstandings and errors related to statistics reported on A/B test experiments, including the erroneous perception that p-value indicates the chance of a false positive. Kohavi et al. (2022) suggest that p-values are widely inaccurately applied even by experts and that intentional efforts need to be made to report meaningful statistical measures.

Though there is some criticism of the use of p-values, we propose that they can be useful in bridging the lack of confidence estimation in NLP system evaluations. Further, we aim to address the effect of variability across samples by using a large number of samples to determine the best approach to p-value generation.

Null hypothesis statistical testing alone as a method for significance testing also does not lead to reproducibility, due to the use in evaluation of inconsistent train-test splitting (Gorman and Bedrick, 2019). We address this as well in our approach by incorporating response variance, discussed in more detail below.

2.2 Response Variance

For each item in a test set, a human rater can provide a *response*, such as a class label or a Likert scale. Prior work indicates the importance of eliciting such responses from multiple raters per item, to account for ambiguity and different perspectives (L Aroyo, 2014; Uma et al., 2021). Regardless of the task, gathering multiple responses results in disagreement. Machine systems also provide a response for each item, and these responses can vary with stochastic training conditions, hyperameter changes, cross-validation, and other causes. System variance can be incorporated into model prediction by merging answers rather than simply ranking (Gondek et al., 2012).

Response variance may be indicative of true features of the data and thus be incorporated into the model (Reidsma and Carletta, 2008). Recent work has indicated that taking a majority vote aggregation may not be effective at resolving/incorporating annotator variance (Davani et al., 2022; Barile et al., 2021).

Prior work has explored the role of variance and data collection in metrics on human annotated datasets (Welty et al., 2019). Homan et al. (2022) provides a framework for analyzing the amount of variance, and types of disagreement, in crowdsourced datasets. Wong et al. (2021) addresses the variance in crowdsourced annotations by presenting a more contextualized measure of inter-rater reliability based on Cohen's kappa. Bayesian models of annotation have also been used and evaluated as potential methods for identifying annotator accuracy and item difficulty (Paun et al., 2018). Recent work has also considered incorporating logical justifications of human viewpoints as a two-dimensional judgment (Draws et al., 2022).

Our simulator produces scores with variance according to different distributions (specified as hyperparameters), allowing us to include response variance in our evaluation.

3 Evaluation Framework

3.1 Problem Formulation

Comparing two NLP systems often involves measuring a baseline B and a candidate A against gold judgments G, to determine whether A is an improvement over B. This comparison is made using a *metric* δ run over a *test set* that is drawn from a population of data. For each item i in the test set, both A and B have a distribution of responses A_i and B_i, and it is possible to have multiple responses for each item. In addition, due to rater disagreement, there is a distribution of human responses, G_i. The metric δ compares each system's responses to the human responses and produces a pair of metric scores, $\delta(A,G)$ and $\delta(B,G)$. Finally, the per-system metric scores are compared to each other so that when $\delta(A,G) > \delta(B,G)$ we can say A is an improvement over B.

The *null hypothesis*, denoted $\mathcal{H}_0$, is that the two sets of responses being compared (i.e. $A_{i,j}$ and $B_{i,j}$, where i is an item and and j is a responses for a given item) *are drawn from the same distribution*. This is compared against an *alternative hypothesis*, denoted $\mathcal{H}_1$, that $A_{i,j}$ and $B_{i,j}$ are true to the underlying distributions from which A, B, and G were drawn, and therefore that the comparison $\delta(A,G)$ and $\delta(B,G)$ is a fair representation of the comparison between A and B. We aim to provide a p-value for this comparison.

By contrast, in the vast majority of NHST settings, A and B are sets of individual responses and there is no notion of variance in i once it is drawn; the only source of variance comes from the sampling of the items. For simple test statistics like mean, a closed-form estimate such as a paired t-test (Student, 1908) will suffice.

However, many metrics used in NLP are not amenable to such closed-form estimates, and the presence of response-level variance means that even many simple metrics cannot be reliably estimated in closed form. Therefore, it is necessary to rely on resampling methods, such as bootstrapping or permutation sampling, to estimate *p*-values. Here we focus on bootstrapping variants, where variance is estimated by resampling from a dataset with replacement.

Usually, the most important design issue in NHST is whether the sample has enough statistical power to detect a difference between A and B when one exists; in our setting, there are two equally fundamental questions: what approach to resampling to use in order to estimate variance, and what metric to use for reliably estimating *p*-values. These design issues led us to the following research questions:

> **RQ1.** Can response-level variance be used to estimate *p*-values?
>
> **RQ2.** What method of sampling response variance generates the most accurate *p*-value?
>
> **RQ3.** What metrics generate the most accurate *p*-value?
>
> **RQ4.** How sensitive are the measurements as two systems' responses draw closer to each other?

3.2 Simulator

To produce and analyze *p*-value estimates from a test set, we built a simulator that operates in three stages. The main idea is to sample a *reference test set* from a known, fixed underlying distribution of items and responses and use a resampling method to estimate the *p*-value of that test set. Then, we use the same underlying distribution to directly estimate the "true" *p*-value of the distribution.

3.2.1 Generating reference test data

First, we generate the reference test data, described in detail in Algorithm 3 in appendix A. The reference test set consists of samples for ground truth (G^{ref}) and two NLP systems (A^{ref} and B^{ref}). Each sample has N items and K responses per item. The responses are continuous values in the interval $[0, 1]$. To construct G^{ref}, for each item i we sample a mean μ_i and standard deviation σ_i from specific uniform distributions. Then, we sample K responses from a normal distribution parameterized by μ_i and σ_i. For A^{ref} and B^{ref}, we use the same sample of means and standard deviations as for G^{ref}, but with μ_i replaced by $\mu_i + \varepsilon_X^i$, where ε_X^i is chosen uniformly at random over the interval $[-\varepsilon_X, \varepsilon_X]$ for $X \in \{A, B\}$, respectively. This process makes the items in the three sets the same, while keeping the responses in each set independent (conditioned on each item i) where the magnitudes of difference in the response distributions are parameterized by ε_A and ε_B.

3.2.2 Sampling for comparative hypothesis testing

Next, we simulate each of the sampling strategies on items and responses. Algorithm 1 describes this process in detail. It takes hyperparameters that specify the item and response *sampling strategies*, respectively (described in §4.1). Here, it is only important to note that our sampling strategies provide rules for resampling from a dataset, such as *sample the items, take all responses* or *sample the items, then sample from the responses for each item*. Algorithm 1 is actually used twice: once for the data needed to estimate the *p*-value based on the reference test set and once for the true *p*-value. In each case it produces data supporting $\mathcal{H}_0$ and $\mathcal{H}_1$.

For the reference test set (`rts`) $\mathcal{H}_1$, we construct three samples corresponding to A^{ref} B^{ref} and G^{ref} by resampling from each according to the given sampling strategy.

For the reference test set $\mathcal{H}_0$, we reuse the sample of G^{ref} constructed for $\mathcal{H}_1$. For A^{ref} and B^{ref}, we operate under the $\mathcal{H}_0$ assumption that they are drawn from the same underlying distribution (when in fact they were drawn from similar distributions, perturbed according to ε_A and ε_B). We do this by first combining A^{ref} and B^{ref} into a single set $A^{\text{ref}}|B^{\text{ref}}$, where each item i in the combined set has all of the responses from both A_i^{ref} and B_i^{ref}. We sample responses for each of A^{ref} and B^{ref} by sampling from $A^{\text{ref}}|B^{\text{ref}}$.

For the true *p*-value (`true`) $\mathcal{H}_1$, Algorithm 1 constructs the samples corresponding to each of A, B and G by ignoring A^{ref}, B^{ref}, and G^{ref} and instead sampling directly from the underlying distribution described in §3.2.1. For the $\mathcal{H}_0$ data, we use the same underlying distribution, except that in order to operate under the $\mathcal{H}_0$ assumption that A^{ref} and B^{ref} are drawn from the same distribution, each item i and response for each of A and B (the process is unchanged for G) is sampled by first uniformly drawing $X \sim \{A, B\}$ and then sampling from the normal distribution parameterized by $(\mu_i + \varepsilon_X^i, \sigma_i)$.

3.2.3 Applying hypothesis tests to (sub)sampled distributions

Finally, for each of the reference test sets and the true distribution, we sample from the distribution M times and feed the output to Algorithm 2, which estimates p-values with respect to a given metric.

Algorithm 1 SAMPLE

```
Input parameters
        $G, A, B$: pointers to reference data or underlying distributions
        $\Phi$: item index sampler
        $\Pi$: response sampler
        $r \in \{\text{rts}, \text{true}\}$ whether to use the input sets for re-sampling or to sample directly from the true underlying distribution.
Results
        $G^*, A^{\text{alt}}, B^{\text{alt}}$: vector (or matrix) samples
        $A^{\varnothing}, B^{\varnothing}$: null hypothesis samples
$j \leftarrow 0$
for all $i \in \Phi(S)$ do
    $G_j^* \leftarrow \Pi_r(G_i)$
    $A_j^{\text{alt}} \leftarrow \Pi_r(A_i)$
    $B_j^{\text{alt}} \leftarrow \Pi_r(B_i)$
    $A_j^{\varnothing} \leftarrow \Pi_r(A_i|B_i)$
    $B_j^{\varnothing} \leftarrow \Pi_r(A_i|B_i)$
    $j \leftarrow j + 1$
end for
```

Algorithm 2 HTEST

```
Input parameters
        $G_j, A_j^{\text{alt}}, B_j^{\text{alt}}, A_j^{\varnothing}, B_j^{\varnothing}, 1 \le j \le M$ constructed from $M$ calls to Algorithm 1
        $\delta$: a test metric
$\alpha \leftarrow 0$
$\beta \leftarrow 0$
for all $j \in \{1, \ldots, M\}$ do
    $\alpha_j = \delta(A_j^{\text{alt}}, G_j) - \delta(B_j^{\text{alt}}, G_j)$
    $\beta_j = \delta(A_j^{\varnothing}, G_j) - \delta(B_j^{\varnothing}, G_j)$
end for
$p \leftarrow 0$
for all $j \in \{1, \ldots, M\}$ do
    $p \leftarrow p + |\{\alpha_{j'} \mid \beta_j < \alpha_{j'}\}|/M$
end for
$p \leftarrow p/M$
```

4 Experiments

We perform a set of experiments on datasets where $N = 1000$ and $K = 5$. These numbers are representative of the number of items in typical test sets and of the numbers of responses in test sets where multiple responses are reported. We consider 6 sampling methods, 8 metrics, and 5 levels of perturbation of system B (we fix the perturbation $\varepsilon_A = 0$ and treat it as an ideal model[1]).

4.1 Sampling strategies for response variance

We experiment with 6 test set sampling methods to calculate a p-value. By implementing these methods, we are able to determine which of these approaches on a single test set best approximates the true p-value.

- Randomly sampling one response. *(all_items, sample(1))* uses all items and randomly selects one response per item, e.g. [0.6,0.4,0.8,0.5,0.4][2] → 0.4.
- Bootstrapping responses. *(all_items, sample(5))* uses all items and samples `n=5` responses per item as in “bootstrapping” (Welty et al., 2019), with replacement, e.g. [0.6,0.4,0.8,0.5,0.4] → [0.6,0.6,0.5,0.4,0.5].
- Bootstrapping items. *(bootstrap_items, all)* bootstrap samples `n=1000` items with replacement and uses all responses for each item.
- Bootstrapping items, selecting one response per item. *(bootstrap_items, first_element)* bootstrap samples `n=1000` items with replacement and selects the first response per item, e.g. [0.6,0.4,0.8,0.5,0.4] → 0.6.[3]
- Bootstrapping items, randomly selecting one response per item. *(bootstrap_items, all)* bootstrap samples `n=1000` items with replacement and randomly selects one response per item, e.g. [0.6,0.4,0.8,0.5,0.4] → 0.4.
- Bootstrapping items, bootstrapping responses. *(bootstrap_items, sample(5))* bootstrap samples `n=1000` items with replacement samples

[1] We fix the perturbation to zero and focus on comparing the sampling methods and metrics under this ideal setting, though varying ε_A in further experimentation will provide additional insight to the generalizability of our results

[2] Here and below, this vector represents the set of $K = 5$ sampled responses associated with each item in our experiment.

[3] Because we are using resampling methods to estimate p-values, using the first item only results in less variance than sampling one item from all five responses. This corresponds to a case in which there is only one response per item.

n=5 responses per item with replacement, e.g. [0.6,0.4,0.8,0.5,0.4] → [0.8,0.6,0.4,0.6,0.5].

4.2 Metrics

We implement 8 metrics to compare the gold scores and the systems output:

- Mean absolute error (MAE). Calculate the error for each item, i.e. the distance (absolute value of the difference) from gold to system responses, then take the mean of the item-wise error. Note that if the size of the response sample per item is greater than 1, the responses per item are aggregated to the mean.
- (Inverse) Mean-squared error (MSE). Mean squared error (inverted so that higher is better) across all items.
- Item-wise metric wins (Wins_{δ}). Compare the system responses to gold for each item using a metric δ, and count the number of items in the set for which each system performs better (i.e. wins). In Table 2, we show the results only for $\mathrm{Wins}_{\mathrm{MAE}}$.
- Cosine distance. First, vectorize each matrix. Transform each from an $n \times k$ to an $nk \times 1$ dimensional matrix. Then $\delta_{\cos}(A,G) = 1 - \frac{A \cdot G}{\|A\|\|G\|}$, $\delta_{\cos}(B,G) = 1 - \frac{B \cdot G}{\|B\|\|G\|}$,
- Aggregated EMD. Mean of each item and earth mover's distance of the entire vertical distribution.
- Aggregated EMD vectorized. Transform each from an $n \times k$ to an $nk \times 1$ dimensional matrix. Then take the earth mover's distance on the entire vectorized distribution.
- Mean of EMDs. Earth mover's distance of each individual item and mean of all of the EMD scores.
- Spearman Rho. Spearman's correlation between the vectors of mean responses per item.

5 Results of simulation study

We examine which of the metrics and sampling methods on a single test set best estimate the true p-value, by calculating the error between the estimated p-value and true p-value across five response distribution perturbations ($\varepsilon_A = 0, \varepsilon_B \in \{0.0, 0.05, 0.1, 0.3, 0.7\}$, *q.v.* Alg. 3).

We expect that as the amount of perturbation applied to system B increases, it should be clearer that the data is drawn from two separate distributions. A metric that is more sensitive to the effect of perturbation/distance should have a **smaller difference between the estimated p-value and true p-value when the perturbation is increased**. Consequently, the metric should have a harder time producing the estimated p-value when the systems are closer together—meaning a larger difference between the estimated p-value and true p-value when the perturbation is less.

Table 1 shows estimation error for each of the sampling methods (minimized across all metrics). The estimation error is the difference between the p-value estimated from a single test set and the true p-value. With $\varepsilon_B = 0.1$, the (all_items, sample(5)), (bootstrap_items, all), and (bootstrap_items, sample(5)) all perform well. These three sampling methods are clearly the best performers for all $\varepsilon > 0$. On the other hand, sampling strategies which reduce the amount of responses per item (i.e. sample(1) and first_element) are not as effective. These findings indicate that **incorporating the variance into the evaluation enables a more accurate statistical comparison**.

Table 2 shows estimation error for each of the metrics (minimized across all sampling methods). To illuminate trends across perturbation levels, Figure 1 visualizes the results from Table 2, and some interesting patterns emerge. As discussed above, we expect a good method to decrease its p-value estimates as the perturbation of B (the x axis) increases.

Multiple metrics (cosine similarity, $\mathrm{Wins}_{\mathrm{MAE}}$, and Spearman Rho) show lower minimum differences at each increasing interval of perturbation. This suggests that these metrics, when operating under unknown conditions / distances between system A and system B, may behave most predictably. $\mathrm{Wins}_{\mathrm{MAE}}$ has the lowest difference in true and estimated p-value for $\varepsilon > 0$, making this the preferred metric.

The least consistent metric is Aggregated EMD vectorized, which increased, decreased, and increased again in minimum difference between estimated and true p-values at increasing levels of perturbation (Table 2).

It is important to note that `p=0.05` is a critical value when considering statistical guarantees, so differences in estimated and true p-values close to or exceeding 0.05 are not acceptable; if the difference in estimated and true p-value is close to 0.05, there is sufficient room for error for it to seem like there *is* evidence of model difference, when in fact there *is not*.

Sampling Method	n	$\varepsilon_B = 0$	$\varepsilon_B = 0.05$	$\varepsilon_B = 0.1$	$\varepsilon_B = .3$	$\varepsilon_B = .7$
(all_items,sample(1))	6	0.04261	0.02185	0.00285	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
(all_items,sample(5))	9	0.02152	0.00289	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
(bootstrap_items, all)	9	0.00621	0.00166	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
(bootstrap_items, first_element)	6	0.04243	0.06268	0.00462	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
(bootstrap_items, sample(1))	6	0.05317	0.00171	0.00184	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
(bootstrap_items, sample(5))	9	0.02094	0.00680	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$

Table 1: Minimum p-value estimation error by sampling method (a tuple of item and response sampler), based on n experiments per method, for five different levels of M2 perturbation (ε_B), with $\varepsilon_A = 0$. n is the number of experiments using a given method (i.e. number of metrics used in combination with this sampling method).

Metric	n	$\varepsilon_B = 0$	$\varepsilon_B = 0.05$	$\varepsilon_B = 0.1$	$\varepsilon_B = 0.3$	$\varepsilon_B = .7$
Cosine Similarity	6	0.13246	0.02128	0.00184	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
Aggregated EMD	6	0.02094	0.00444	0.00342	0.03633	$< \mathbf{10^{-5}}$
Aggregated EMD vectorized	3	0.01807	0.00415	0.00478	0.00808	$< \mathbf{10^{-5}}$
MSE	6	0.00621	0.03206	0.01349	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
MAE	6	0.01071	0.02929	0.00020	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
Wins_{MAE}	9	0.08724	0.00166	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
Mean of EMDs	3	0.02152	0.02721	0.03219	0.00022	$< \mathbf{10^{-5}}$
Spearman Rho	6	0.07934	0.02114	0.01110	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$

Table 2: Minimum p-value estimation error by metric, based on n experiments per metric, for five different levels of M2 perturbation (ε_B), with $\varepsilon_A = 0$. n is the number of experiments using a given metric (i.e. number of sampling methods used in combination with this metric).

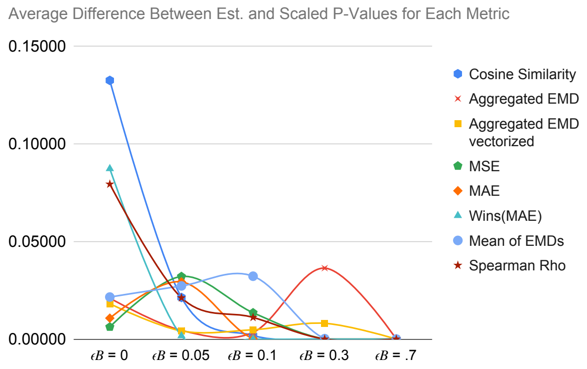

Figure 1: Minimum difference between estimated p-score and true p-score for each of the 8 metrics, at the 5 levels of perturbation.

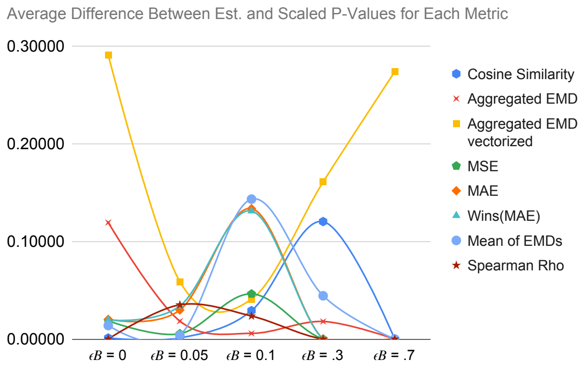

Figure 2: For our application to real-world data: minimum difference between estimated p-score and true p-score for each of the 8 metrics and 5 levels of perturbation.

6 Application to real-world data

To apply our method on actual data, we need the item and response data for the ground truth and the two machines (G^{ref}, A^{ref}, and B^{ref}, respectively). For our example, we chose Kumar et al. (2021), a dataset of 107,620 social media comments that are labeled by five annotators each on the toxicity of each comment, using a 5-level Likert scale from 0–4. We randomly sampled 1000 items from it for G^{ref}, normalizing the annotations into [0,1], yielding possible responses $\{0, 0.2, 0.4, 0.6, 0.8\}$.

Next, we match the hyperparameters of Algorithm 3 to the actual underlying distributions. We assume that each response $G_{i,k}$ is drawn from a normal distribution with a specific mean and standard deviation for each item, as before, except rather than assuming they come from uniform distributions as in Algorithm 3 we now take parameterized models `foldednormal([0, 0.28])` and `triangular`([-0.05, 0.21, 0.45] for the means and standard deviations, respectively, fitted to the 107,620-comment dataset. We visually inspect the histograms to determine the probabilistic model

Sampling Method	n	$\varepsilon_B = 0$	$\varepsilon_B = 0.05$	$\varepsilon_B = 0.1$	$\varepsilon_B = .3$	$\varepsilon_B = .7$
(all_items,sample(1))	6	0.00108	0.00545	0.02909	0.01037	$< \mathbf{10^{-5}}$
(all_items,sample(5))	9	0.02020	0.00390	0.00585	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
(bootstrap_items, all)	9	0.00014	0.00120	0.00604	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
(bootstrap_items, first_element)	6	0.10096	0.03511	0.02359	0.01801	$< \mathbf{10^{-5}}$
(bootstrap_items, sample(1))	6	0.00193	0.00893	0.02965	0.00958	$< \mathbf{10^{-5}}$
(bootstrap_items, sample(5))	9	0.00406	0.00265	0.03939	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$

Table 3: On real toxicity data: minimum p-value estimation error by sampling method, a tuple of item and response sampler, based on n experiments per method, for five different levels of M2 perturbation (ε_B), with $\varepsilon_A = 0$.

Metric	n	$\varepsilon_B = 0$	$\varepsilon_B = 0.05$	$\varepsilon_B = 0.1$	$\varepsilon_B = .3$	$\varepsilon_B = .7$
Cosine Similarity	6	0.00108	0.00120	0.02909	0.12034	$< \mathbf{10^{-5}}$
Aggregated EMD	6	0.11932	0.01835	0.00585	0.01801	$< \mathbf{10^{-5}}$
Aggregated EMD vectorized	3	0.29085	0.05877	0.04086	0.16090	0.27378
MSE	6	0.01866	0.00545	0.04629	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
MAE	6	0.02020	0.02973	0.13357	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
Wins_{MAE}	9	0.01942	0.03391	0.131146	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$
Mean of EMDs	3	0.01382	0.00390	0.14331	0.04434	$< \mathbf{10^{-5}}$
Spearman Rho	6	0.00014	0.03511	0.02359	$< \mathbf{10^{-5}}$	$< \mathbf{10^{-5}}$

Table 4: On real toxicity data: minimum p-value estimation error by metric, based on n experiments per metric, for five different levels of M2 perturbation (ε_B), with $\varepsilon_A = 0$.

to use, and then choose the hyperparameters that minimizes the mean absolute error between the observed data distributions and those predicted by the models. This process is described in appendix B. We also assume that, after sampling from a normal distribution the results in the range $[0, 0.2)$ are converted to 0.2, those in the range $[0.2, 0.4)$ are converted to 0.4 etc. to simulate the discrete nature of Likert responses.

With these parameters set, we can run the framework described in §3.2, with the toxicity dataset sample as our reference test set and simulated system responses to choose the best metric and sampling method to use on G^{ref}.

We expect to see results similar to those from the pure simulation study, although the fact that responses are now discrete, rather than continuous, there will be sharper differences in performance between different values of ε_B.

The results on the toxicity dataset (Table 3, Table 4, Figure 2) exhibit some of the same patterns seen in the pure simulation results. (Bootstrap_items, all) is the best sampling strategy, and the strategies that take only one response per item seem to do the worst. Among the metrics, Spearman Rho has the best overall performance.

However, it should be noted that for $\varepsilon_B \in \{0, 0.05, 0.1\}$ the maximum amount of perturbation is relatively small compared to the 0.2 interval between successive elements in the response domain. There is not much observable difference in the performance between A and B until $\varepsilon_B = 0.3$. At this point, many of the metrics do well. Another interesting pattern in the metric results is that Spearman Rho is among the better performers in most cases, particularly for $\varepsilon_B \in \{0.3, 0.7\}$.

7 Discussion

These experiments suggest answers to our four research questions:

RQ1. Can response-level variance be used to estimate p-values? Yes. In Table 1 and Table 3 we see that (bootstrap_items, first_element)—the only sampling method that does not make use of response variance—generally performs poorly in the three lowest perturbation settings, and becomes competitive with the best approach only at $\varepsilon_B = 0.7$. The response variance appears to make the measurements more sensitive to smaller differences between evaluated systems.

RQ2. What method of sampling response variance generates the most accurate p-value? The most promising sampling method is (bootstrap_items, all) (Table 1 and Table 3).

RQ3. What metrics generate the most accurate p-value? In the purely simulated data, Wins_{MAE} is the best (Table 2). In the toxicity dataset, MSE, MAE, Wins_{MAE}, and Spearman Rho all do very well for $\varepsilon_B \geq 0.3$, and MSE and Spearman Rho do better than the rest for smaller perturbation levels (Table 4).

RQ4. How sensitive are the measurements as two systems' response distributions draw closer to each other? On the purely simulated data, Wins_{MAE} is the most consistent metric when considering sensitivity to distance between system A and system B (Figure 1).

Compared to the purely simulated data, the real toxicity dataset exhibits a much sharper difference among the performance of the better metrics as ε_B increases. This is likely due to binning the responses into five discrete levels, meaning that levels of perturbation that are detectable in a continuous domain (which the purely simulated data has) are negligible over a discrete domain (which the toxicity data has) when they are much smaller than the size of each bin. However, as the perturbation levels approach or exceed the bin size, the binning quite suddenly creates starker differences in the toxicity dataset than in the purely simulated data (Table 2 and Table 4).

Our results suggest that, of the methods explored here, the Wins_{MAE} metric, in combination with the (bootstrap_items, all) sampling technique, provides the most effective p-value estimate on a single test set. That Wins_{MAE} performed poorly on $\varepsilon_B = 0$ (or, on the toxicity dataset, $\varepsilon_B \leq 0.1$) should not distract from its superior performance for other choices of ε_B. Recall that $\varepsilon_B = 0$ (or, on the toxicity dataset, $\varepsilon_B \leq 0.1$) means that the null hypothesis is (effectively) true (Hung et al., 1997; Boos and Stefanski, 2011; Colquhoun, 2014) and p-values are very large, so larger errors are less critical.

Compared to MAE, Wins_{MAE}'s counting wins likely outperforms taking the mean due to the small sample of responses taken for each item (i.e., no more than five for each). With such a small sample it is hard to estimate the mean with any degree of precision, so when these means are aggregated over all 1000 items, this lack of precision accumulates. Even though we cannot reliably estimate the mean with two samples, in comparing two samples of size five, it is still possible to tell when one mean is likely greater than the other: a win is a binary, whereas the mean is a continuous, variable, so the mean carries more information. Thus, at lower sample sizes it is harder to estimate.

Our results suggest that, among the metrics and sampling methods studied here, the choice of best metric is independent of the choice of sampling method, and vice versa.

8 Conclusion

8.1 Overview & Findings

In this work we address the lack of statistical rigor in system evaluation and propose a framework to help tackle this problem. Here, we constructed a statistical approach to comparing two systems against gold/human judgments. After developing a simulator to test the utility of sampling methods and metrics on many test sets, we experimented with 6 sampling methods, 8 metrics, and 5 levels of distance between system A (proposed system) and system B (baseline). We find that sampling methods which incorporate variance perform better, and that Wins_{MAE} and Spearman Rho are reliable metrics.

8.2 Recommendations for Practitioners

While this testing regime is our general recommendation for future work evaluating NLP systems, our findings indicate that evaluation protocol requires tuning to the specific task and data. Generally, our results show that incorporating variance into sampling strategy enables more rigorous statistical evaluation, and both Wins_{MAE} and Spearman Rho are metrics which seem to be strong in their sensitivity to perturbation.

These methods are useful for designing an experiment, as they can indicate an optimal metric or sampling strategy, as well as number of necessary items or annotators for the task.

Beyond specific recommendations for metrics and sampling methods, our results demonstrate that machine similarity (distance in distribution between the baseline and the proposed system), sampling method, and metric chosen affect leaderboard performance, and statistical guarantees should be provided when claiming that a proposed model outperforms an existing model.

8.3 Future Work

In future work, we would like to consider further hyperparameters, such as the effect of number of responses on the measurement sensitivity, categorical responses as opposed to continuous numerical data, and different item and response distributions. In the latter case, we believe that understanding the item and response distributions of an evaluated system will be an important element in choosing sampling strategies and metrics.

Limitations

As the contributions of this work include a framework and preliminary experimentation, there are a number of constraints that we leave to future work. Firstly, we considered only one family of response distributions. We chose normal distributions because their behavior is well-understood and they are easy to work with. However, the structural similarities between normal distributions and the best performing metrics—namely, absolute error—suggests that, more generally, the best test metrics for NHST may vary depending on the underlying response distributions. Therefore, we recommend that use of our framework should potentially vary depending on the dataset being considered, and might have other distributions commonly found in model and gold standard items and responses, such as exponential or multinomial distributions.

Similarly, we only considered p-value estimators that are based on bootstrap sampling. Implementation of our framework in future use would benefit from matching the estimator to the test metric. For instance, permutation tests are the most common way to estimate p-values for Spearman correlation, and analytical tests such as Student's or MacNemar's, which are commonly used even when the underlying assumptions on which they are based are not likely to hold (as, we expect is the case here). As such, the sampling method could change based on which metric is best for the task/data.

Acknowledgements

We thank anonymous reviewers for their helpful feedback. This work is supported by Google Research and a Clare Boothe Luce Scholarship.

References

Lora Aroyo and Chris Welty. 2015. Truth is a lie: Crowd Truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24.

Francesco Barile, Shabnam Najafian, Tim Draws, Oana Inel, Alisa Rieger, Rishav Hada, and Nava Tintarev. 2021. Toward benchmarking group explanations: Evaluating the effect of aggregation strategies versus explanation. In *Perspectives@ RecSys*.

Dennis D Boos and Leonard A Stefanski. 2011. P-value precision and reproducibility. *The American Statistician*, 65(4):213–221.

David Colquhoun. 2014. An investigation of the false discovery rate and the misinterpretation of p-values. *Royal Society open science*, 1(3):140216.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A statistical analysis of summarization evaluation metrics using resampling methods. *Transactions of the Association for Computational Linguistics*, 9:1132–1146.

Thomas G Dietterich. 1998a. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Thomas G Dietterich. 1998b. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine learning*, 32:1–22.

Tim Draws, Oana Inel, Nava Tintarev, Christian Baden, and Benjamin Timmermans. 2022. Comprehensive viewpoint representations for a deeper understanding of user interactions with debated topics. In *ACM SIGIR Conference on Human Information Interaction and Retrieval*, pages 135–145.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Rotem Dror, Lotem Peled-Cohen, Segev Shlomov, and Roi Reichart. 2020. Statistical significance testing for natural language processing. *Synthesis Lectures on Human Language Technologies*, 13(2):1–116.

Tommaso Fornaciari, Alexandra Uma, Massimo Poesio, and Dirk Hovy. 2022. Hard and soft evaluation of NLP models with BOOtSTrap SAmpling - BooStSa. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 127–134, Dublin, Ireland. Association for Computational Linguistics.

D. C. Gondek, A. Lally, A. Kalyanpur, J. W. Murdock, P. A. Duboue, L. Zhang, Y. Pan, Z. M. Qiu, and C. Welty. 2012. A framework for merging and ranking of answers in deepqa. *IBM Journal of Research and Development*, 56(3.4):14:1–14:12.

Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791, Florence, Italy. Association for Computational Linguistics.

Odd Erik Gundersen and Sigbjørn Kjensmo. 2018. State of the art: Reproducibility in artificial intelligence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Lewis G Halsey, Douglas Curran-Everett, Sarah L Vowler, and Gordon B Drummond. 2015. The fickle p value generates irreproducible results. *Nature methods*, 12(3):179–185.

Matan Haroush, Tzviel Frostig, Ruth Heller, and Daniel Soudry. 2021. A statistical framework for efficient out of distribution detection in deep neural networks. In *International Conference on Learning Representations*.

Christopher Homan, Tharindu Cyril Weerasooriya, Lora Mois Aroyo, and Chris Welty. 2022. Annotator response distributions as a sampling frame. In *LREC Workshop on Perspectivist NLP*.

HM James Hung, Robert T O'Neill, Peter Bauer, and Karl Kohne. 1997. The behavior of the p-value when the alternative hypothesis is true. *Biometrics*, pages 11–22.

Ron Kohavi, Alex Deng, and Lukas Vermeer. 2022. A/b testing intuition busters: Common misunderstandings in online controlled experiments. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3168–3177.

Deepak Kumar, Patrick Gage Kelley, Sunny Consolvo, Joshua Mason, Elie Bursztein, Zakir Durumeric, Kurt Thomas, and Michael Bailey. 2021. Designing toxic content classification for a diversity of perspectives. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*, pages 299–318.

C Welty L Aroyo. 2013. Crowd truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard. In *Web Science 2013*.

C Welty L Aroyo. 2014. The three sides of crowdtruth.

Nathan Mantel. 1963. Chi-square tests with one degree of freedom; extensions of the mantel-haenszel procedure. *Journal of the American Statistical Association*, 58(303):690–700.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

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

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 507–511, Baltimore, Maryland. Association for Computational Linguistics.

M Poesio R Artstein. Inter-coder agreement for computational linguistics. *Computational linguistics*, 34(4):555–596.

Dennis Reidsma and Jean Carletta. 2008. Squibs: Reliability measurement without limits. *Computational Linguistics*, 34(3):319–326.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. 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, Honolulu, Hawaii. Association for Computational Linguistics.

Anders Søgaard, Anders Johannsen, Barbara Plank, Dirk Hovy, and Hector Martínez Alonso. 2014. What's in a p-value in NLP? In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 1–10, Ann Arbor, Michigan. Association for Computational Linguistics.

Student. 1908. The probable error of a mean. *Biometrika*, pages 1–25.

Gail M Sullivan and Richard Feinn. 2012. Using effect size—or why the p value is not enough. *Journal of graduate medical education*, 4(3):279–282.

Matthew S Thiese, Brenden Ronna, and Ulrike Ott. 2016. P value interpretations and considerations. *Journal of thoracic disease*, 8(9):E928.

Alexandra N Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Chris Welty, Praveen Paritosh, and Lora Aroyo. 2019. Metrology for ai: From benchmarks to instruments. *arXiv preprint arXiv:1911.01875*.

Ka Wong, Praveen Paritosh, and Lora Aroyo. 2021. Cross-replication reliability – an empirical approach to interpreting inter-rater reliability.

A Algorithm used by the simulation framework

Here we include formalizations of an algorithms used in our work. In Algorithm 3 we specify the process for generating the reference test data.

Algorithm 3 GENTESTSET

```
Input parameters
        $N$: test set size
        $K$: number of responses per item
        $\varepsilon_A$: Perturbation of $A$ scores from $G$
        $\varepsilon_B$: Perturbation of $B$ scores from $G$
Results
        $\mu_i$: Response means per item
        $\sigma_i$: Response standard deviations per item
        $G$: item, response matrix for human annotations
        $A$: item, response matrix for test system
        $B$: item, response matrix for baseline system
for all $i \in [0, N)$ do
    $\mu_i \sim$ uniform([0,1])
    $\sigma_i \sim$ uniform([0,.2])
    $\nu_A \sim$ uniform($[-\varepsilon_A, \varepsilon_A]$)
    $\nu_B \sim$ uniform($[-\varepsilon_B, \varepsilon_B]$)
    for all $k \in [0, k)$ do
        $G_{i,k} \sim$ normal($\mu_i, \sigma_i$)
        $A_{i,k} \sim$ normal($\mu_i + \nu_0, \sigma_i$)
        $B_{i,k} \sim$ normal($\mu_i + \nu_1, \sigma_i$)
    end for
end for
```

B Fitting the mean and standard deviation models to the toxicity dataset

To fit the distribution of the simulated system responses to the dataset, we take the mean and standard deviation of the responses of each item in the dataset. We then inspect histograms of theses values. We noted that the distribution of the item-wise means (Figure 3, left) seems to follow a folded normal distribution that has been *clamped* to the range $[0, .8]$ (i.e., values falling outside that range are assigned to the nearest value in the range, namely 0 or 1). The standard deviations (Figure 4, right) seem to follow a triangular distribution clamped to the range $[0, \infty]$ (i.e., only values less than 0 are reassigned).

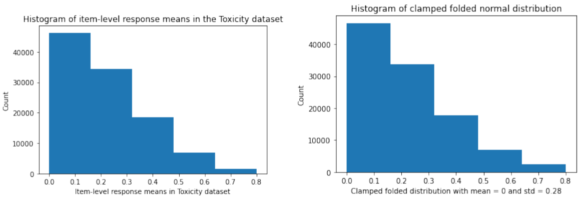

Figure 3: Distribution of item-level response means in the Toxicity dataset (left), and from a 1000-item sample of a folded normal distribution with mean .20 and standard deviation of 0.16, where values greater than zero have been assigned to 0 and values greater than one have been assigned to 0.8.

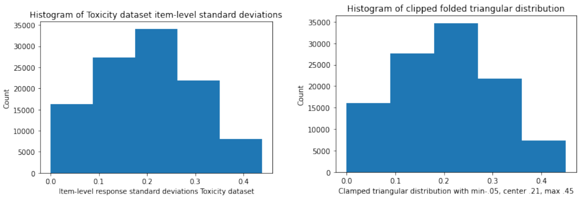

Figure 4: Distribution of item-level response standard deviations in the Toxicity dataset (left), which has a mean of 0.19 and standard deviation of 0.11 and from a 1000-item sample of a triangular distribution with minimum −0.05, apex 0.21, and maximum 0.45, where values greater than zero have been assigned to zero and values greater than one have been assigned to one.

We generated means in our simulator by sampling from a folded normal distribution clamped to $[0, .8]$. Using grid search, we found that assigning this distribution a mean of 0.0 and standard deviation 0.28 minimized the mean absolute error (MAE) between the bars of the histograms in Figure 3. Similarly, a triangular distribution clamped to $[0, 1.0]$ and with minimum, apex, and maximum of −0.05, 0.21 and 0.45 minimized the (MAE) between the bars of the histograms in Figure 4. Both MAE scores were estimated to be around 2000, which is very small considering that the dataset has 107,620 items.

C Complete Results

Here we include the full results for our experiments on both simulated and real data.

Sampling method	Metric	Estimated p	True p	diff
(all_items sample(5))	MAE	0.410587	0.496402	-0.085815
(all_items sample(1))	MAE	0.381939	0.497279	-0.115340
(bootstrap_items sample(5))	MAE	0.427611	0.496402	-0.068791
(bootstrap_items sample(1))	MAE	0.423322	0.476487	-0.053165
(bootstrap_items first_element)	MAE	0.024045	0.492053	-0.468008
(bootstrap_items all)	MAE	0.485691	0.496402	-0.010711
(all_items sample(5))	Wins(MAE)	0.371267	0.510317	-0.139050
(all_items sample(1))	Wins(MAE)	0.367486	0.496606	-0.129120
(bootstrap_items sample(5))	Wins(MAE)	0.374182	0.510317	-0.136135
(bootstrap_items sample(1))	Wins(MAE)	0.403785	0.493224	-0.089439
(bootstrap_items first_element)	Wins(MAE)	0.083394	0.479193	-0.395799
(bootstrap_items all)	Wins(MAE)	0.423081	0.510317	-0.087236
(all_items sample(5))	Wins(MAE) (vectorized)	0.221291	0.504003	-0.282712
(bootstrap_items sample(5))	Wins(MAE) (vectorized)	0.251065	0.483919	-0.232854
(bootstrap_items all)	Wins(MAE) (vectorized)	0.383131	0.500626	-0.117495
(all_items sample(5))	MSE	0.430407	0.48391	-0.053503
(all_items sample(1))	MSE	0.356333	0.497763	-0.141430
(bootstrap_items sample(5))	MSE	0.452586	0.48391	-0.031324
(bootstrap_items sample(1))	MSE	0.400131	0.478923	-0.078792
(bootstrap_items first_element)	MSE	0.012927	0.49465	-0.481723
(bootstrap_items all)	MSE	0.490123	0.48391	0.006213
(all_items sample(5))	Spearman Rho	0.355155	0.494165	-0.139010
(all_items sample(1))	Spearman Rho	0.369992	0.488593	-0.118601
(bootstrap_items sample(5))	Spearman Rho	0.374628	0.494165	-0.119537
(bootstrap_items sample(1))	Spearman Rho	0.412421	0.491757	-0.079336
(bootstrap_items first_element)	Spearman Rho	0.007716	0.497834	-0.490118
(bootstrap_items all)	Spearman Rho	0.397471	0.494165	-0.096694
(all_items sample(5))	EMD Agg	0.4807	0.502944	-0.022244
(all_items sample(1))	EMD Agg	0.4634	0.506008	-0.042608
(bootstrap_items sample(5))	EMD Agg	0.482	0.502944	-0.020944
(bootstrap_items sample(1))	EMD Agg	0.4174	0.504515	-0.087115
(bootstrap_items first_element)	EMD Agg	0.4391	0.481533	-0.042433
(bootstrap_items all)	EMD Agg	0.4574	0.502944	-0.045544
(all_items sample(5))	EMD Agg (vectorized)	0.4418	0.495472	-0.053672
(bootstrap_items sample(5))	EMD Agg (vectorized)	0.4283	0.495472	-0.067172
(bootstrap_items all)	EMD Agg (vectorized)	0.4774	0.495472	-0.018072
(all_items sample(5))	Mean Agg	0.4723	0.493816	-0.021516
(bootstrap_items sample(5))	Mean Agg	0.4495	0.493816	-0.044316
(bootstrap_items all)	Mean Agg	0.3281	0.493816	-0.165716
(all_items sample(5))	COS (vectorized)	0.186133	0.478781	-0.292648
(all_items sample(1))	COS (vectorized)	0.308011	0.493675	-0.185664
(bootstrap_items sample(5))	COS (vectorized)	0.206125	0.493335	-0.287210
(bootstrap_items sample(1))	COS (vectorized)	0.348773	0.481228	-0.132455
(bootstrap_items first_element)	COS	0.014585	0.497565	-0.482980
(bootstrap_items all)	COS (vectorized)	0.154479	0.463897	-0.309418

Table 5: Full results on the purely simulated data for $\varepsilon_B = 0$.

Sampling method	Metric	Estimated p	True p	diff
(all_items sample(5))	MAE	0.331591	0.067622	0.263969
(all_items sample(1))	MAE	0.416351	0.350643	0.065708
(bootstrap_items sample(5))	MAE	0.340759	0.067622	0.273137
(bootstrap_items sample(1))	MAE	0.407312	0.378019	0.029293
(bootstrap_items first_element)	MAE	0.418707	0.356032	0.062675
(bootstrap_items all)	MAE	0.19662	0.067622	0.128998
(all_items sample(5))	Wins(MAE)	0.044571	0.001909	0.042662
(all_items sample(1))	Wins(MAE)	0.109459	0.087604	0.021855
(bootstrap_items sample(5))	Wins(MAE)	0.046853	0.001909	0.044944
(bootstrap_items sample(1))	Wins(MAE)	0.111549	0.113261	-0.001712
(bootstrap_items first_element)	Wins(MAE)	0.040036	0.10596	-0.065924
(bootstrap_items all)	Wins(MAE)	0.003566	0.001909	0.001657
(all_items sample(5))	Wins(MAE) (vectorized)	0.004189	0.001297	0.002892
(bootstrap_items sample(5))	Wins(MAE) (vectorized)	0.008305	0.001504	0.006801
(bootstrap_items all)	Wins(MAE) (vectorized)	0.010192	0.001759	0.008433
(all_items sample(5))	MSE	0.479884	0.262248	0.217636
(all_items sample(1))	MSE	0.482863	0.450803	0.032060
(bootstrap_items sample(5))	MSE	0.484121	0.262248	0.221873
(bootstrap_items sample(1))	MSE	0.512241	0.464467	0.047774
(bootstrap_items first_element)	MSE	0.116947	0.442982	-0.326035
(bootstrap_items all)	MSE	0.486482	0.262248	0.224234
(all_items sample(5))	Spearman Rho	0.458313	0.263784	0.194529
(all_items sample(1))	Spearman Rho	0.486023	0.442924	0.043099
(bootstrap_items sample(5))	Spearman Rho	0.473643	0.263784	0.209859
(bootstrap_items sample(1))	Spearman Rho	0.485766	0.464627	0.021139
(bootstrap_items first_element)	Spearman Rho	0.211795	0.434099	-0.222304
(bootstrap_items all)	Spearman Rho	0.485429	0.263784	0.221645
(all_items sample(5))	EMD Agg	0.4788	0.483241	-0.004441
(all_items sample(1))	EMD Agg	0.513	0.482578	0.030422
(bootstrap_items sample(5))	EMD Agg	0.5077	0.483241	0.024459
(bootstrap_items sample(1))	EMD Agg	0.4974	0.517659	-0.020259
(bootstrap_items first_element)	EMD Agg	0.3833	0.489908	-0.106608
(bootstrap_items all)	EMD Agg	0.5365	0.483241	0.053259
(all_items sample(5))	EMD Agg (vectorized)	0.4774	0.495554	-0.018154
(bootstrap_items sample(5))	EMD Agg (vectorized)	0.3859	0.495554	-0.109654
(bootstrap_items all)	EMD Agg (vectorized)	0.4914	0.495554	-0.004154
(all_items sample(5))	Mean Agg	0.2238	0.195486	0.028314
(bootstrap_items sample(5))	Mean Agg	0.2387	0.195486	0.043214
(bootstrap_items all)	Mean Agg	0.2227	0.195486	0.027214
(all_items sample(5))	COS (vectorized)	0.471735	0.390999	0.080736
(all_items sample(1))	COS (vectorized)	0.475717	0.449534	0.026183
(bootstrap_items sample(5))	COS (vectorized)	0.46214	0.375208	0.086932
(bootstrap_items sample(1))	COS (vectorized)	0.480865	0.459581	0.021284
(bootstrap_items first_element)	COS	0.132421	0.437445	-0.305024
(bootstrap_items all)	COS (vectorized)	0.35917	0.384993	-0.025823

Table 6: Full results on the purely simulated data for $\varepsilon_B = 0.05$. A result of “0” means that the p-value was less than the simulator’s minimum level of precision (10^{-5}).

Sampling method	Metric	Estimated p	True p	diff
(all_items sample(5))	MAE	0.003156	2.00E-06	0.003154
(all_items sample(1))	MAE	0.112564	0.115949	-0.003385
(bootstrap_items sample(5))	MAE	0.006948	2.00E-06	0.006946
(bootstrap_items sample(1))	MAE	0.120493	0.112452	0.008041
(bootstrap_items first_element)	MAE	0.253229	0.11249	0.140739
(bootstrap_items all)	MAE	0.000204	2.00E-06	0.000202
(all_items sample(5))	Wins(MAE)	3.10E-05	0	0.000031
(all_items sample(1))	Wins(MAE)	0.002333	0.005179	-0.002846
(bootstrap_items sample(5))	Wins(MAE)	5.90E-05	0	0.000059
(bootstrap_items sample(1))	Wins(MAE)	0.00328	0.006268	-0.002988
(bootstrap_items first_element)	Wins(MAE)	0.002366	0.006986	-0.004620
(bootstrap_items all)	Wins(MAE)	0	0	0.000000
(all_items sample(5))	Wins(MAE) (vectorized)	0	0	0.000000
(bootstrap_items sample(5))	Wins(MAE) (vectorized)	0	0	0.000000
(bootstrap_items all)	Wins(MAE) (vectorized)	0	0	0.000000
(all_items sample(5))	MSE	0.164925	0.010974	0.153951
(all_items sample(1))	MSE	0.330706	0.317212	0.013494
(bootstrap_items sample(5))	MSE	0.191154	0.010974	0.180180
(bootstrap_items sample(1))	MSE	0.341757	0.303606	0.038151
(bootstrap_items first_element)	MSE	0.207481	0.298607	-0.091126
(bootstrap_items all)	MSE	0.061281	0.010974	0.050307
(all_items sample(5))	Spearman Rho	0.167256	0.009594	0.157662
(all_items sample(1))	Spearman Rho	0.329187	0.318091	0.011096
(bootstrap_items sample(5))	Spearman Rho	0.191409	0.009594	0.181815
(bootstrap_items sample(1))	Spearman Rho	0.330523	0.301009	0.029514
(bootstrap_items first_element)	Spearman Rho	0.143159	0.313247	-0.170088
(bootstrap_items all)	Spearman Rho	0.099042	0.009594	0.089448
(all_items sample(5))	EMD Agg	0.4212	0.480088	-0.058888
(all_items sample(1))	EMD Agg	0.4815	0.484922	-0.003422
(bootstrap_items sample(5))	EMD Agg	0.5018	0.480088	0.021712
(bootstrap_items sample(1))	EMD Agg	0.4958	0.501032	-0.005232
(bootstrap_items first_element)	EMD Agg	0.3286	0.505215	-0.176615
(bootstrap_items all)	EMD Agg	0.36	0.480088	-0.120088
(all_items sample(5))	EMD Agg (vectorized)	0.4289	0.433678	-0.004778
(bootstrap_items sample(5))	EMD Agg (vectorized)	0.4751	0.433678	0.041422
(bootstrap_items all)	EMD Agg (vectorized)	0.4194	0.433678	-0.014278
(all_items sample(5))	Mean Agg	0.1104	0.032391	0.078009
(bootstrap_items sample(5))	Mean Agg	0.1454	0.032391	0.113009
(bootstrap_items all)	Mean Agg	0.0002	0.032391	-0.032191
(all_items sample(5))	COS (vectorized)	0.149166	0.130691	0.018475
(all_items sample(1))	COS (vectorized)	0.293829	0.322725	-0.028896
(bootstrap_items sample(5))	COS (vectorized)	0.183086	0.131847	0.051239
(bootstrap_items sample(1))	COS (vectorized)	0.305439	0.307283	-0.001844
(bootstrap_items first_element)	COS	0.228971	0.307595	-0.078624
(bootstrap_items all)	COS (vectorized)	0.111144	0.1314	-0.020256

Table 7: Full results on the purely simulated data for $\varepsilon_B = 0.1$. A result of "0" means that the p-value was less than the simulator's minimum level of precision (10^{-5}).

Sampling method	Metric	Estimated p	True p	diff
(all_items sample(5))	MAE	0	0	0.000000
(all_items sample(1))	MAE	0	0	0.000000
(bootstrap_items sample(5))	MAE	0	0	0.000000
(bootstrap_items sample(1))	MAE	0	0	0.000000
(bootstrap_items first_element)	MAE	0	0	0.000000
(bootstrap_items all)	MAE	0	0	0.000000
(all_items sample(5))	Wins(MAE)	0	0	0.000000
(all_items sample(1))	Wins(MAE)	0	0	0.000000
(bootstrap_items sample(5))	Wins(MAE)	0	0	0.000000
(bootstrap_items sample(1))	Wins(MAE)	0	0	0.000000
(bootstrap_items first_element)	Wins(MAE)	0	0	0.000000
(bootstrap_items all)	Wins(MAE)	0	0	0.000000
(all_items sample(5))	Wins(MAE) (vectorized)	0	0	0.000000
(bootstrap_items sample(5))	Wins(MAE) (vectorized)	0	0	0.000000
(bootstrap_items all)	Wins(MAE) (vectorized)	0	0	0.000000
(all_items sample(5))	MSE	0	0	0.000000
(all_items sample(1))	MSE	9.00E-06	0.000237	-0.000228
(bootstrap_items sample(5))	MSE	0	0	0.000000
(bootstrap_items sample(1))	MSE	5.20E-05	0.000272	-0.000220
(bootstrap_items first_element)	MSE	0.009451	0.000117	0.009334
(bootstrap_items all)	MSE	0	0	0.000000
(all_items sample(5))	Spearman Rho	0	0	0.000000
(all_items sample(1))	Spearman Rho	3.00E-06	0.000263	-0.000260
(bootstrap_items sample(5))	Spearman Rho	0	0	0.000000
(bootstrap_items sample(1))	Spearman Rho	8.30E-05	0.000105	-0.000022
(bootstrap_items first_element)	Spearman Rho	0.012638	0.000111	0.012527
(bootstrap_items all)	Spearman Rho	0	0	0.000000
(all_items sample(5))	EMD Agg	0.2976	0.141411	0.156189
(all_items sample(1))	EMD Agg	0.294	0.257674	0.036326
(bootstrap_items sample(5))	EMD Agg	0.3763	0.141411	0.234889
(bootstrap_items sample(1))	EMD Agg	0.2968	0.252052	0.044748
(bootstrap_items first_element)	EMD Agg	0.3665	0.291064	0.075436
(bootstrap_items all)	EMD Agg	0.2933	0.141411	0.151889
(all_items sample(5))	EMD Agg (vectorized)	0.0198	0.027877	-0.008077
(bootstrap_items sample(5))	EMD Agg (vectorized)	0.1785	0.027877	0.150623
(bootstrap_items all)	EMD Agg (vectorized)	0.0983	0.027877	0.070423
(all_items sample(5))	Mean Agg	0.0047	8.40E-05	0.004616
(bootstrap_items sample(5))	Mean Agg	0.0094	8.40E-05	0.009316
(bootstrap_items all)	Mean Agg	0.0003	8.40E-05	0.000216
(all_items sample(5))	COS (vectorized)	0	0	0.000000
(all_items sample(1))	COS (vectorized)	2.50E-05	0.000425	-0.000400
(bootstrap_items sample(5))	COS (vectorized)	0	0	0.000000
(bootstrap_items sample(1))	COS (vectorized)	0.000111	0.000373	-0.000262
(bootstrap_items first_element)	COS	0.017425	0.000224	0.017201
(bootstrap_items all)	COS (vectorized)	0	0	0.000000

Table 8: Full results on the purely simulated data for $\varepsilon_B = 0.3$. A result of “0” means that the p-value was less than the simulator’s minimum level of precision (10^{-5}).

Sampling method	Metric	Estimated p	True p	diff
(all_items sample(5))	MAE	0	0	0.000000
(all_items sample(1))	MAE	0	0	0.000000
(bootstrap_items sample(5))	MAE	0	0	0.000000
(bootstrap_items sample(1))	MAE	0	0	0.000000
(bootstrap_items first_element)	MAE	0	0	0.000000
(bootstrap_items all)	MAE	0	0	0.000000
(all_items sample(5))	Wins(MAE)	0	0	0.000000
(all_items sample(1))	Wins(MAE)	0	0	0.000000
(bootstrap_items sample(5))	Wins(MAE)	0	0	0.000000
(bootstrap_items sample(1))	Wins(MAE)	0	0	0.000000
(bootstrap_items first_element)	Wins(MAE)	0	0	0.000000
(bootstrap_items all)	Wins(MAE)	0	0	0.000000
(all_items sample(5))	Wins(MAE) (vectorized)	0	0	0.000000
(bootstrap_items sample(5))	Wins(MAE) (vectorized)	0	0	0.000000
(bootstrap_items all)	Wins(MAE) (vectorized)	0	0	0.000000
(all_items sample(5))	MSE	0	0	0.000000
(all_items sample(1))	MSE	0	0	0.000000
(bootstrap_items sample(5))	MSE	0	0	0.000000
(bootstrap_items sample(1))	MSE	0	0	0.000000
(bootstrap_items first_element)	MSE	0	0	0.000000
(bootstrap_items all)	MSE	0	0	0.000000
(all_items sample(5))	Spearman Rho	0	0	0.000000
(all_items sample(1))	Spearman Rho	0	0	0.000000
(bootstrap_items sample(5))	Spearman Rho	0	0	0.000000
(bootstrap_items sample(1))	Spearman Rho	0	0	0.000000
(bootstrap_items first_element)	Spearman Rho	0	0	0.000000
(bootstrap_items all)	Spearman Rho	0	0	0.000000
(all_items sample(5))	EMD Agg	0	2.00E-06	-0.000002
(all_items sample(1))	EMD Agg	0	0.00047	-0.000470
(bootstrap_items sample(5))	EMD Agg	0	2.00E-06	-0.000002
(bootstrap_items sample(1))	EMD Agg	0.0049	0.000828	0.004072
(bootstrap_items first_element)	EMD Agg	0.0006	0.000672	-0.000072
(bootstrap_items all)	EMD Agg	0.0068	2.00E-06	0.006798
(all_items sample(5))	EMD Agg (vectorized)	0	0	0.000000
(bootstrap_items sample(5))	EMD Agg (vectorized)	0	0	0.000000
(bootstrap_items all)	EMD Agg (vectorized)	0	0	0.000000
(all_items sample(5))	Mean Agg	0	0	0.000000
(bootstrap_items sample(5))	Mean Agg	0	0	0.000000
(bootstrap_items all)	Mean Agg	0	0	0.000000
(all_items sample(5))	COS (vectorized)	0	0	0.000000
(all_items sample(1))	COS (vectorized)	0	0	0.000000
(bootstrap_items sample(5))	COS (vectorized)	0	0	0.000000
(bootstrap_items sample(1))	COS (vectorized)	0	0	0.000000
(bootstrap_items first_element)	COS	0	0	0.000000
(bootstrap_items all)	COS (vectorized)	0	0	0.000000

Table 9: Full results on the purely simulated data for $\varepsilon_B = 0.7$. A result of “0” means that the p-value was less than the simulator’s minimum level of precision (10^{-5}).

Sampling method	Metric	Estimated p	True p	diff
(all_items sample(5))	MAE	0.45934	0.479535	-0.020195
(all_items sample(1))	MAE	0.437902	0.499429	-0.061527
(bootstrap_items sample(5))	MAE	0.450871	0.479526	-0.028655
(bootstrap_items sample(1))	MAE	0.456307	0.485554	-0.029247
(bootstrap_items first_element)	MAE	0.266924	0.492477	-0.225553
(bootstrap_items all)	MAE	0.397145	0.479537	-0.082392
(all_items sample(5))	Wins(MAE)	0.43663	0.497893	-0.061263
(all_items sample(1))	Wins(MAE)	0.457158	0.50005	-0.042892
(bootstrap_items sample(5))	Wins(MAE)	0.416203	0.496002	-0.079799
(bootstrap_items sample(1))	Wins(MAE)	0.467904	0.487324	-0.019420
(bootstrap_items first_element)	Wins(MAE)	0.211593	0.496652	-0.285059
(bootstrap_items all)	Wins(MAE)	0.365914	0.496656	-0.130742
(all_items sample(5))	Wins(MAE) (vectorized)	0.384364	0.476789	-0.092425
(bootstrap_items sample(5))	Wins(MAE) (vectorized)	0.394187	0.469652	-0.075465
(bootstrap_items all)	Wins(MAE) (vectorized)	0.34856	0.480752	-0.132192
(all_items sample(5))	MSE	0.457059	0.479458	-0.022399
(all_items sample(1))	MSE	0.433271	0.507752	-0.074481
(bootstrap_items sample(5))	MSE	0.460799	0.47946	-0.018661
(bootstrap_items sample(1))	MSE	0.449009	0.480377	-0.031368
(bootstrap_items first_element)	MSE	0.298127	0.492372	-0.194245
(bootstrap_items all)	MSE	0.498544	0.479457	0.019087
(all_items sample(5))	Spearman Rho	0.493405	0.473137	0.020268
(all_items sample(1))	Spearman Rho	0.471087	0.50108	-0.029993
(bootstrap_items sample(5))	Spearman Rho	0.49305	0.472682	0.020368
(bootstrap_items sample(1))	Spearman Rho	0.478448	0.480376	-0.001928
(bootstrap_items first_element)	Spearman Rho	0.26784	0.501546	-0.233706
(bootstrap_items all)	Spearman Rho	0.472451	0.472311	0.000140
(all_items sample(5))	EMD Agg	0.3172	0.492373	-0.175173
(all_items sample(1))	EMD Agg	0.3247	0.48702	-0.162320
(bootstrap_items sample(5))	EMD Agg	0.384	0.503324	-0.119324
(bootstrap_items sample(1))	EMD Agg	0.3562	0.48553	-0.129330
(bootstrap_items first_element)	EMD Agg	0.2671	0.486825	-0.219725
(bootstrap_items all)	EMD Agg	0.3417	0.490803	-0.149103
(all_items sample(5))	EMD Agg (vectorized)	0.1856	0.496253	-0.310653
(bootstrap_items sample(5))	EMD Agg (vectorized)	0.2054	0.496253	-0.290853
(bootstrap_items all)	EMD Agg (vectorized)	0.1448	0.496253	-0.351453
(all_items sample(5))	Mean Agg	0.3894	0.479924	-0.090524
(bootstrap_items sample(5))	Mean Agg	0.4661	0.479924	-0.013824
(bootstrap_items all)	Mean Agg	0.1139	0.479924	-0.366024
(all_items sample(5))	COS (vectorized)	0.47414	0.49728	-0.023140
(all_items sample(1))	COS (vectorized)	0.486513	0.487594	-0.001081
(bootstrap_items sample(5))	COS (vectorized)	0.478938	0.483	-0.004062
(bootstrap_items sample(1))	COS (vectorized)	0.495858	0.481335	0.014523
(bootstrap_items first_element)	COS	0.398975	0.49993	-0.100955
(bootstrap_items all)	COS (vectorized)	0.508019	0.497584	0.010435

Table 10: Full results on the toxicity data for $\varepsilon_B = 0$.

Sampling method	Metric	Estimated p	True p	diff
(all_items sample(5))	MAE	0.451311	0.236154	0.215157
(all_items sample(1))	MAE	0.451132	0.421398	0.029734
(bootstrap_items sample(5))	MAE	0.443935	0.236095	0.207840
(bootstrap_items sample(1))	MAE	0.472664	0.416839	0.055825
(bootstrap_items first_element)	MAE	0.34272	0.418018	-0.075298
(bootstrap_items all)	MAE	0.304942	0.236123	0.068819
(all_items sample(5))	Wins(MAE)	0.464451	0.268947	0.195504
(all_items sample(1))	Wins(MAE)	0.447476	0.413566	0.033910
(bootstrap_items sample(5))	Wins(MAE)	0.439536	0.271841	0.167695
(bootstrap_items sample(1))	Wins(MAE)	0.474745	0.40652	0.068225
(bootstrap_items first_element)	Wins(MAE)	0.441751	0.404092	0.037659
(bootstrap_items all)	Wins(MAE)	0.480294	0.271587	0.208707
(all_items sample(5))	Wins(MAE) (vectorized)	0.394146	0.29071	0.103436
(bootstrap_items sample(5))	Wins(MAE) (vectorized)	0.378982	0.293645	0.085337
(bootstrap_items all)	Wins(MAE) (vectorized)	0.246202	0.293862	-0.047660
(all_items sample(5))	MSE	0.512868	0.308322	0.204546
(all_items sample(1))	MSE	0.460799	0.455346	0.005453
(bootstrap_items sample(5))	MSE	0.484069	0.308322	0.175747
(bootstrap_items sample(1))	MSE	0.478558	0.449968	0.028590
(bootstrap_items first_element)	MSE	0.310163	0.446163	-0.136000
(bootstrap_items all)	MSE	0.480889	0.308322	0.172567
(all_items sample(5))	Spearman Rho	0.478749	0.291773	0.186976
(all_items sample(1))	Spearman Rho	0.495227	0.455432	0.039795
(bootstrap_items sample(5))	Spearman Rho	0.497833	0.292022	0.205811
(bootstrap_items sample(1))	Spearman Rho	0.503192	0.44878	0.054412
(bootstrap_items first_element)	Spearman Rho	0.405459	0.440566	-0.035107
(bootstrap_items all)	Spearman Rho	0.415004	0.291148	0.123856
(all_items sample(5))	EMD Agg	0.4512	0.492407	-0.041207
(all_items sample(1))	EMD Agg	0.4608	0.479153	-0.018353
(bootstrap_items sample(5))	EMD Agg	0.3984	0.506423	-0.108023
(bootstrap_items sample(1))	EMD Agg	0.436	0.504794	-0.068794
(bootstrap_items first_element)	EMD Agg	0.5262	0.485954	0.040246
(bootstrap_items all)	EMD Agg	0.349	0.488988	-0.139988
(all_items sample(5))	EMD Agg (vectorized)	0.4067	0.478072	-0.071372
(bootstrap_items sample(5))	EMD Agg (vectorized)	0.4193	0.478072	-0.058772
(bootstrap_items all)	EMD Agg (vectorized)	0.3572	0.478072	-0.120872
(all_items sample(5))	Mean Agg	0.3623	0.358396	0.003904
(bootstrap_items sample(5))	Mean Agg	0.3967	0.358396	0.038304
(bootstrap_items all)	Mean Agg	0.2297	0.358396	-0.128696
(all_items sample(5))	COS (vectorized)	0.427134	0.431173	-0.004039
(all_items sample(1))	COS (vectorized)	0.464628	0.477367	-0.012739
(bootstrap_items sample(5))	COS (vectorized)	0.430813	0.433462	-0.002649
(bootstrap_items sample(1))	COS (vectorized)	0.473743	0.464818	0.008925
(bootstrap_items first_element)	COS	0.349589	0.465382	-0.115793
(bootstrap_items all)	COS (vectorized)	0.441221	0.440026	0.001195

Table 11: Full results on the toxicity data for $\varepsilon_B = 0.05$

Sampling method	Metric	Estimated p	True p	diff
(all_items sample(5))	MAE	0.339637	0.013932	0.325705
(all_items sample(1))	MAE	0.495166	0.275734	0.219432
(bootstrap_items sample(5))	MAE	0.345923	0.013927	0.331996
(bootstrap_items sample(1))	MAE	0.474302	0.290926	0.183376
(bootstrap_items first_element)	MAE	0.471536	0.286101	0.185435
(bootstrap_items all)	MAE	0.147494	0.013921	0.133573
(all_items sample(5))	Wins(MAE)	0.369515	0.034306	0.335209
(all_items sample(1))	Wins(MAE)	0.473703	0.235071	0.238632
(bootstrap_items sample(5))	Wins(MAE)	0.368552	0.03339	0.335162
(bootstrap_items sample(1))	Wins(MAE)	0.461407	0.238133	0.223274
(bootstrap_items first_element)	Wins(MAE)	0.483026	0.242806	0.240220
(bootstrap_items all)	Wins(MAE)	0.164385	0.033239	0.131146
(all_items sample(5))	Wins(MAE) (vectorized)	0.394516	0.061719	0.332797
(bootstrap_items sample(5))	Wins(MAE) (vectorized)	0.398901	0.057616	0.341285
(bootstrap_items all)	Wins(MAE) (vectorized)	0.300014	0.065973	0.234041
(all_items sample(5))	MSE	0.357053	0.058006	0.299047
(all_items sample(1))	MSE	0.475893	0.359268	0.116625
(bootstrap_items sample(5))	MSE	0.365982	0.058006	0.307976
(bootstrap_items sample(1))	MSE	0.490422	0.386699	0.103723
(bootstrap_items first_element)	MSE	0.417344	0.371051	0.046293
(bootstrap_items all)	MSE	0.150775	0.058006	0.092769
(all_items sample(5))	Spearman Rho	0.229324	0.054669	0.174655
(all_items sample(1))	Spearman Rho	0.467556	0.345706	0.121850
(bootstrap_items sample(5))	Spearman Rho	0.238225	0.054456	0.183769
(bootstrap_items sample(1))	Spearman Rho	0.456305	0.374117	0.082188
(bootstrap_items first_element)	Spearman Rho	0.374624	0.351033	0.023591
(bootstrap_items all)	Spearman Rho	0.017202	0.054292	-0.037090
(all_items sample(5))	EMD Agg	0.4446	0.438747	0.005853
(all_items sample(1))	EMD Agg	0.4008	0.49448	-0.093680
(bootstrap_items sample(5))	EMD Agg	0.4805	0.441111	0.039389
(bootstrap_items sample(1))	EMD Agg	0.3422	0.476702	-0.134502
(bootstrap_items first_element)	EMD Agg	0.228	0.469601	-0.241601
(bootstrap_items all)	EMD Agg	0.4349	0.440939	-0.006039
(all_items sample(5))	EMD Agg (vectorized)	0.2881	0.388563	-0.100463
(bootstrap_items sample(5))	EMD Agg (vectorized)	0.3477	0.388563	-0.040863
(bootstrap_items all)	EMD Agg (vectorized)	0.3013	0.388563	-0.087263
(all_items sample(5))	Mean Agg	0.3243	0.18099	0.143310
(bootstrap_items sample(5))	Mean Agg	0.3476	0.18099	0.166610
(bootstrap_items all)	Mean Agg	0.4127	0.18099	0.231710
(all_items sample(5))	COS (vectorized)	0.396048	0.319219	0.076829
(all_items sample(1))	COS (vectorized)	0.436377	0.407283	0.029094
(bootstrap_items sample(5))	COS (vectorized)	0.411232	0.317353	0.093879
(bootstrap_items sample(1))	COS (vectorized)	0.471161	0.44151	0.029651
(bootstrap_items first_element)	COS	0.351366	0.41475	-0.063384
(bootstrap_items all)	COS (vectorized)	0.23454	0.320495	-0.085955

Table 12: Full results on the toxicity data for $\varepsilon_B = 0.1$

Sampling method	Metric	Estimated p	True p	diff
(all_items sample(5))	MAE	0	0	0.000000
(all_items sample(1))	MAE	0.034235	0.000458	0.033777
(bootstrap_items sample(5))	MAE	0	0	0.000000
(bootstrap_items sample(1))	MAE	0.029396	0.000589	0.028807
(bootstrap_items first_element)	MAE	0.077095	0.000665	0.076430
(bootstrap_items all)	MAE	0	0	0.000000
(all_items sample(5))	Wins(MAE)	1.50E-05	0	0.000015
(all_items sample(1))	Wins(MAE)	0.010578	0.000207	0.010371
(bootstrap_items sample(5))	Wins(MAE)	6.30E-05	0	0.000063
(bootstrap_items sample(1))	Wins(MAE)	0.009861	0.000277	0.009584
(bootstrap_items first_element)	Wins(MAE)	0.029022	0.000549	0.028473
(bootstrap_items all)	Wins(MAE)	0	0	0.000000
(all_items sample(5))	Wins(MAE) (vectorized)	0	0	0.000000
(bootstrap_items sample(5))	Wins(MAE) (vectorized)	0	0	0.000000
(bootstrap_items all)	Wins(MAE) (vectorized)	0	0	0.000000
(all_items sample(5))	MSE	0	0	0.000000
(all_items sample(1))	MSE	0.114515	0.007642	0.106873
(bootstrap_items sample(5))	MSE	9.70E-05	0	0.000097
(bootstrap_items sample(1))	MSE	0.102692	0.007427	0.095265
(bootstrap_items first_element)	MSE	0.187849	0.007277	0.180572
(bootstrap_items all)	MSE	0	0	0.000000
(all_items sample(5))	Spearman Rho	0	0	0.000000
(all_items sample(1))	Spearman Rho	0.072887	0.029437	0.043450
(bootstrap_items sample(5))	Spearman Rho	0	0	0.000000
(bootstrap_items sample(1))	Spearman Rho	0.069153	0.028034	0.041119
(bootstrap_items first_element)	Spearman Rho	0.075912	0.033092	0.042820
(bootstrap_items all)	Spearman Rho	0	0	0.000000
(all_items sample(5))	EMD Agg	0.0647	0.010304	0.054396
(all_items sample(1))	EMD Agg	0.3256	0.146758	0.178842
(bootstrap_items sample(5))	EMD Agg	0.1355	0.011847	0.123653
(bootstrap_items sample(1))	EMD Agg	0.3838	0.14794	0.235860
(bootstrap_items first_element)	EMD Agg	0.1294	0.147411	-0.018011
(bootstrap_items all)	EMD Agg	0.3004	0.011109	0.289291
(all_items sample(5))	EMD Agg (vectorized)	0.2419	0.003602	0.238298
(bootstrap_items sample(5))	EMD Agg (vectorized)	0.1899	0.003602	0.186298
(bootstrap_items all)	EMD Agg (vectorized)	0.1645	0.003602	0.160898
(all_items sample(5))	Mean Agg	0.1394	0.001862	0.137538
(bootstrap_items sample(5))	Mean Agg	0.1537	0.001862	0.151838
(bootstrap_items all)	Mean Agg	0.0462	0.001862	0.044338
(all_items sample(5))	COS (vectorized)	0.12329	0.002952	0.120338
(all_items sample(1))	COS (vectorized)	0.306477	0.106948	0.199529
(bootstrap_items sample(5))	COS (vectorized)	0.166159	0.003684	0.162475
(bootstrap_items sample(1))	COS (vectorized)	0.29751	0.092561	0.204949
(bootstrap_items first_element)	COS	0.412021	0.100953	0.311068
(bootstrap_items all)	COS (vectorized)	0.125169	0.003615	0.121554

Table 13: Full results on the toxicity data for $\varepsilon_B = 0.3$. A result of “0” means that the p-value was less than the simulator’s minimum level of precision (10^{-5}).

Sampling method	Metric	Estimated p	True p	diff
(all_items sample(5))	MAE	0	0	0.000000
(all_items sample(1))	MAE	0	0	0.000000
(bootstrap_items sample(5))	MAE	0	0	0.000000
(bootstrap_items sample(1))	MAE	0	0	0.000000
(bootstrap_items first_element)	MAE	0	0	0.000000
(bootstrap_items all)	MAE	0	0	0.000000
(all_items sample(5))	Wins(MAE)	0	0	0.000000
(all_items sample(1))	Wins(MAE)	0	0	0.000000
(bootstrap_items sample(5))	Wins(MAE)	0	0	0.000000
(bootstrap_items sample(1))	Wins(MAE)	0	0	0.000000
(bootstrap_items first_element)	Wins(MAE)	0	0	0.000000
(bootstrap_items all)	Wins(MAE)	0	0	0.000000
(all_items sample(5))	Wins(MAE) (vectorized)	0	0	0.000000
(bootstrap_items sample(5))	Wins(MAE) (vectorized)	0	0	0.000000
(bootstrap_items all)	Wins(MAE) (vectorized)	0	0	0.000000
(all_items sample(5))	MSE	0	0	0.000000
(all_items sample(1))	MSE	0	0	0.000000
(bootstrap_items sample(5))	MSE	0	0	0.000000
(bootstrap_items sample(1))	MSE	0	0	0.000000
(bootstrap_items first_element)	MSE	2.30E-05	0	0.000023
(bootstrap_items all)	MSE	0	0	0.000000
(all_items sample(5))	Spearman Rho	0	0	0.000000
(all_items sample(1))	Spearman Rho	3.30E-05	2.60E-05	0.000007
(bootstrap_items sample(5))	Spearman Rho	0	0	0.000000
(bootstrap_items sample(1))	Spearman Rho	8.40E-05	1.50E-05	0.000069
(bootstrap_items first_element)	Spearman Rho	0.002207	1.20E-05	0.002195
(bootstrap_items all)	Spearman Rho	0	0	0.000000
(all_items sample(5))	EMD Agg	0.0055	0	0.005500
(all_items sample(1))	EMD Agg	0.3862	1.80E-05	0.386182
(bootstrap_items sample(5))	EMD Agg	0.0301	0	0.030100
(bootstrap_items sample(1))	EMD Agg	0.3843	0.001	0.383300
(bootstrap_items first_element)	EMD Agg	0.0522	0.000999	0.051201
(bootstrap_items all)	EMD Agg	0.0008	0	0.000800
(all_items sample(5))	EMD Agg (vectorized)	0.3511	0.000816	0.350284
(bootstrap_items sample(5))	EMD Agg (vectorized)	0.3889	0.000816	0.388084
(bootstrap_items all)	EMD Agg (vectorized)	0.2746	0.000816	0.273784
(all_items sample(5))	Mean Agg	0.0104	1.00E-06	0.010399
(bootstrap_items sample(5))	Mean Agg	0.0255	1.00E-06	0.025499
(bootstrap_items all)	Mean Agg	0	1.00E-06	-0.000001
(all_items sample(5))	COS (vectorized)	0	0	0.000000
(all_items sample(1))	COS (vectorized)	0.006554	0.00048	0.006074
(bootstrap_items sample(5))	COS (vectorized)	0	0	0.000000
(bootstrap_items sample(1))	COS (vectorized)	0.010388	0.000377	0.010011
(bootstrap_items first_element)	COS	0.09628	0.000329	0.095951
(bootstrap_items all)	COS (vectorized)	5.10E-05	0	0.000051

Table 14: Full results on the toxicity data for $\varepsilon_B = 0.7$. A result of "0" means that the p-value was less than the simulator's minimum level of precision (10^{-5}).