

How to *Correctly* Report LLM-as-a-Judge Evaluations

Chungpa Lee¹ Thomas Zeng² Jongwon Jeong² Jy-yong Sohn¹ Kangwook Lee^{2,3}
¹Yonsei University ²University of Wisconsin–Madison ³KRAFTON

Abstract

Large language models (LLMs) are increasingly used as evaluators in lieu of humans. While scalable, their judgments are noisy due to imperfect specificity and sensitivity of LLMs, leading to biased accuracy estimates. Although bias-correction methods exist, they are underutilized in LLM research and typically assume exact knowledge of the model’s specificity and sensitivity. Furthermore, in general we only have estimates of these values and it is not well known how to properly construct confidence intervals using only estimates. This work presents a simple plug-in framework that corrects such bias and constructs confidence intervals reflecting uncertainty from both test and calibration dataset, enabling practical and statistically sound LLM-based evaluation. Additionally, to reduce uncertainty in the accuracy estimate, we introduce an adaptive algorithm that efficiently allocates calibration sample sizes.

1 Introduction

The use of Large language models (LLMs) as judges provides a cheap, scalable alternative to human evaluation for various tasks like grading factual accuracy, assessing code quality or detecting harmful content (Zheng et al., 2023; Liu et al., 2023; Wang et al., 2023; Li et al., 2025; Gu et al., 2025). However, directly using a point-estimate \hat{p} (e.g., the raw proportion of answers the LLM judges as ‘correct’) as a quality metric is statistically problematic (Angelopoulos et al., 2023; Boyreau et al., 2025; Fraser, 2024; Albinet, 2025). Because LLM judgments are inherently noisy, reporting these uncorrected results leads to biased evaluations (Wang et al., 2024; Koo et al., 2024; Huang et al., 2025).

The nature of this distortion becomes clear once we examine how LLM evaluators make errors. As shown in Fig. 1, an LLM may incorrectly judge an ‘incorrect’ answer as ‘correct’ or, conversely, mislabel a ‘correct’ answer as ‘incorrect’. Let q_0 and q_1 denote the probabilities that the LLM makes the right decision in each case. For instance, in the extreme case where $q_0 = 0$ and $q_1 = 1$, the LLM judges every answer as ‘correct’, causing the naive estimate \hat{p} to be identically 1 regardless of the true accuracy θ . This illustrates how misleading the raw judgment proportion can be.

In general, whenever the LLM is imperfect ($q_0 + q_1 < 2$), the expected value of \hat{p} deviates from the ground-truth accuracy θ :

$$\mathbb{E}[\hat{p}] = \theta + (2 - q_0 - q_1) \left(\frac{1 - q_0}{2 - q_0 - q_1} - \theta \right),$$

implying positive bias at low θ and negative bias at high θ (see Sec. 4 for details). This behavior is illustrated in Fig. 2a for an LLM with $q_0 = 0.7$ and $q_1 = 0.9$. $\mathbb{E}[\hat{p}]$ overestimates θ when $\theta < 0.75$ (blue line) and underestimates it when $\theta > 0.75$ (red line), a pattern caused by the two underlying

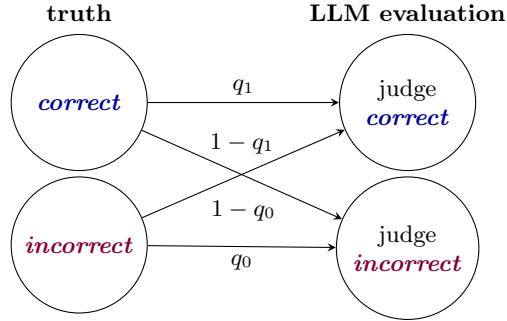


Figure 1: LLM judgment errors, where q_1 and q_0 are LLM’s sensitivity and specificity.

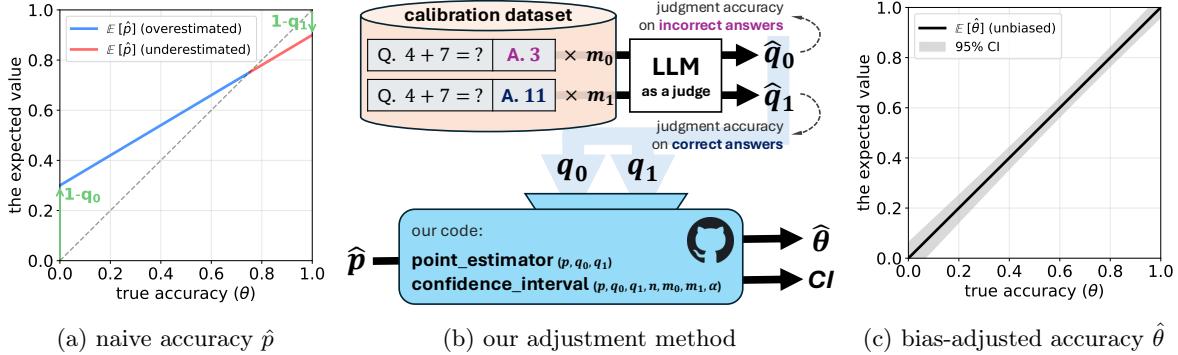


Figure 2: Bias and its adjustment in LLM-based judgment under imperfect LLM evaluators ($q_0 = 0.7$ and $q_1 = 0.9$). (a) When the true accuracy θ is low ($\theta < 0.75$), the expected value of the naive estimator $E[\hat{p}]$ overestimates θ , whereas when θ is high ($\theta > 0.75$), it underestimates θ . (b) By incorporating the judgment accuracies q_0 and q_1 , which can also be estimated from a calibration dataset with ground-truth labels, we obtain the bias-adjusted estimator $\hat{\theta}$ along with its confidence interval. (c) The resulting estimator $\hat{\theta}$ is unbiased when the true values of q_0 and q_1 are known or when a sufficiently large calibration dataset is available. A plug-in Python implementation of this procedure is provided in <https://github.com/UW-Madison-Lee-Lab/LLM-judge-reporting>.

judgment errors (green arrows). A high error probability of wrongly accepting an ‘incorrect’ answer (large $1 - q_0$) induces positive bias at low accuracies, whereas a high probability of wrongly rejecting a ‘correct’ answer (large $1 - q_1$) induces negative bias at high accuracies.

This issue is not merely theoretical. As LLM-based evaluation becomes more common, reported improvements may sometimes be influenced by judgment bias rather than true model gains. This suggests that some advances in the literature could stem from differences in evaluation procedures, motivating careful comparison across studies and the use of calibrated judges when interpreting past findings. Consequently, progress may have been overstated or understated depending on the bias direction, highlighting the need for a principled method for bias adjustment.

Importantly, this bias can be corrected. When q_0 and q_1 are known, a classical result from prevalence estimation (Rogan and Gladen, 1978) provides an exact adjustment. Even when they are unknown, they can be estimated from a calibration dataset with ground-truth labels, and the resulting estimates \hat{q}_0 and \hat{q}_1 can be substituted into the correction formula. Fig. 2c shows that this adjustment substantially reduces the distortion in \hat{p} , producing the bias-adjusted estimator $\hat{\theta}$.

Correcting the point estimate, however, is only part of the problem. LLM-as-a-Judge evaluation involves two sources of uncertainty: (i) randomness arising from the test dataset, which affects the estimated judgment score \hat{p} , and (ii) randomness from the calibration dataset, which affects \hat{q}_0 and \hat{q}_1 . A principled confidence interval must incorporate both components; yet prior discussion on LLM-as-a-Judge (Fraser, 2024; Albinet, 2025) has focused largely on bias, offering no practical method for constructing valid intervals.

This work provides a full statistical treatment of the LLM-as-a-Judge setting. We formalize the relationship between the observed judgment score \hat{p} and the true accuracy θ , derive a simple plug-in estimator that corrects the resulting bias, and construct a confidence interval that accounts for uncertainty arising from both the test and calibration datasets, as outlined in Fig. 2b. In addition, we introduce an adaptive allocation algorithm that efficiently distributes calibration sample sizes across the two true-label types to reduce the uncertainty of the final accuracy estimate.

2 Problem Setup: LLM-as-a-Judge

Consider the set of all possible test instances \mathcal{X} . For example, a single instance $x \in \mathcal{X}$ may consist of a question together with the corresponding answer produced by a given model¹. To evaluate these instances, we assume that there is a human-defined notion of *correctness*. This is specified by a ground-truth labeler $z : \mathcal{X} \rightarrow \{0, 1\}$, where $z(x) = 1$ indicates that humans judge the answer in instance x to be ‘*correct*’, and $z(x) = 0$ indicates it is judged ‘*incorrect*’. Applying this function to the random variable X induces a binary random variable $Z := z(X)$. Our goal is to measure the true accuracy of the model with respect to human judgment:

$$\theta := \Pr(Z = 1) = \mathbb{E}[Z]. \quad (1)$$

In practice, an LLM is often used to judge *correctness* instead of human annotators. Let $\hat{Z} := f_{\text{LLM}}(X) \in \{0, 1\}$ denote the LLM’s judgment, where $\hat{Z} = 1$ means the LLM marks the answer as ‘*correct*’, and $\hat{Z} = 0$ means it marks the answer as ‘*incorrect*’. For a test set consisting of n instances $\{x_1, \dots, x_n\}$, the LLM produces labels $\hat{z}_i := f_{\text{LLM}}(x_i) \in \{0, 1\}$ for $i \in \{1, \dots, n\}$. The accuracy reported in practice is the empirical fraction of instances labeled as ‘*correct*’ by the LLM judge:

$$\hat{p} := \frac{1}{n} \sum_{i=1}^n \hat{z}_i. \quad (2)$$

This quantity estimates the population-level probability $p := \Pr(\hat{Z} = 1)$, which represents the probability that the LLM judges a randomly drawn instance as ‘*correct*’.

However, the LLM’s judgment \hat{Z} does not necessarily coincide with the human ground-truth label Z . That is, the LLM may incorrectly reject answers that are truly ‘*correct*’ or incorrectly accept answers that are truly ‘*incorrect*’. The accuracy of the LLM’s judgment in these two cases is captured by

$$q_1 := \Pr(\hat{Z} = 1 \mid Z = 1) \quad \text{and} \quad q_0 := \Pr(\hat{Z} = 0 \mid Z = 0),$$

which correspond to the *sensitivity* (true positive rate) and *specificity* (true negative rate), respectively (Forman, 2008; Lang and Reiczigel, 2014).

Because the LLM may misjudge both ‘*correct*’ and ‘*incorrect*’ answers, the naive estimator \hat{p} generally satisfies $\mathbb{E}[\hat{p}] \neq \theta$, where θ is the true accuracy defined in (1). Moreover, existing evaluations typically report only this point estimate and do not quantify the uncertainty of the judged accuracy, such as through a confidence interval. As a result, reported accuracy may appear precise even when it is statistically unreliable. Therefore, our goal is to obtain a bias-adjusted estimate of θ and to report its uncertainty using a statistically sound confidence interval.

3 Method to Correctly Report LLM-as-a-Judge Evaluations

In this section, we introduce a method for correcting the bias inherent in LLM-as-a-Judge evaluations and for quantifying the uncertainty of the resulting estimates. We first describe the bias-adjusted point estimator and then present a confidence interval that accounts for uncertainty from both the test and calibration datasets. The section concludes with an analysis of how sample sizes affect interval length and an adaptive allocation strategy to efficiently design the calibration dataset.

Mitigating Bias on Point Estimator We begin with the setting in which the LLM’s judgment accuracies q_0 and q_1 are known. In this case, the unbiased estimator of the true accuracy θ in (1) is

$$\hat{\theta} \mid q_0, q_1 = \frac{\hat{p} + q_0 - 1}{q_0 + q_1 - 1}, \quad (3)$$

¹The model producing the answer may be an LLM, but rule-based or statistical models are also possible.

where detailed derivations are deferred to Sec. 4.

In realistic settings, these accuracies are unknown and must be estimated from a calibration dataset with human-verified labels. Each calibration instance contains both the ground-truth label $z \in \{0, 1\}$ and the corresponding LLM prediction $\hat{z} \in \{0, 1\}$. Let m_0 and m_1 denote the number of calibration examples with $z = 0$ and $z = 1$, respectively. The accuracies of the LLM’s judgment on the two subsets are

$$\hat{q}_0 := \frac{1}{m_0} \sum_{z=0} \mathbf{1}\{\hat{z} = 0 | z = 0\}, \quad \hat{q}_1 := \frac{1}{m_1} \sum_{z=1} \mathbf{1}\{\hat{z} = 1 | z = 1\},$$

where $\mathbf{1}\{\cdot\}$ denotes the indicator function. Substituting these estimates into (3) gives the bias-adjusted estimator (Rogan and Gladen, 1978):

$$\hat{\theta} = \frac{\hat{p} + \hat{q}_0 - 1}{\hat{q}_0 + \hat{q}_1 - 1}. \quad (4)$$

Estimating Uncertainty through Confidence Interval To quantify uncertainty in $\hat{\theta}$, we derive a $(1 - \alpha)$ confidence interval for θ that incorporates variance contributions from both the test and calibration dataset (Lang and Reiczigel, 2014):

$$\tilde{\theta} + d\tilde{\theta} \pm z_\alpha \sqrt{\frac{\tilde{p}(1 - \tilde{p})/\tilde{n} + (1 - \tilde{\theta})^2 \cdot \tilde{q}_0(1 - \tilde{q}_0)/\tilde{m}_0 + \tilde{\theta}^2 \cdot \tilde{q}_1(1 - \tilde{q}_1)/\tilde{m}_1}{(\tilde{q}_0 + \tilde{q}_1 - 1)^2}}, \quad (5)$$

where values outside the interval $[0, 1]$ are truncated to 0 or 1. Here, z_α denotes the $(1 - \alpha/2)$ quantile of the standard normal distribution, e.g., $z_{0.05} = 1.96$, and the adjusted quantities are defined as

$$\begin{aligned} \tilde{n} &= n + z_\alpha^2, & \tilde{m}_0 &= m_0 + 2, & \tilde{m}_1 &= m_1 + 2, \\ \tilde{p} &= \frac{n \cdot \hat{p} + z_\alpha^2/2}{n + z_\alpha^2}, & \tilde{q}_0 &= \frac{m_0 \cdot \hat{q}_0 + 1}{m_0 + 2}, & \tilde{q}_1 &= \frac{m_1 \cdot \hat{q}_1 + 1}{m_1 + 2}, \end{aligned} \quad (6)$$

$$\tilde{\theta} = \frac{\tilde{p} + \tilde{q}_0 - 1}{\tilde{q}_0 + \tilde{q}_1 - 1}, \quad d\tilde{\theta} = 2z_\alpha^2 \left(-(1 - \tilde{\theta}) \cdot \frac{\tilde{q}_0(1 - \tilde{q}_0)}{\tilde{m}_0} + \tilde{\theta} \cdot \frac{\tilde{q}_1(1 - \tilde{q}_1)}{\tilde{m}_1} \right). \quad (7)$$

Impact of Sample Sizes on Confidence-Interval Length The confidence interval in (5) reflects uncertainty arising from both the test and calibration datasets through \tilde{n} , \tilde{m}_0 , and \tilde{m}_1 . As these sample sizes increase, the terms inside the square root decrease, leading to a shorter confidence interval for θ . In particular, because LLM-as-a-Judge evaluations can be run at scale with minimal cost, the test-set size n can often be made extremely large. In the limit $n \rightarrow \infty$, test-set uncertainty vanishes entirely, and the interval length is determined solely by the calibration sample sizes m_0 and m_1 . This observation enables practitioners to specify a desired interval length and then determine the minimal calibration budget required to achieve it.

Fig. 3 illustrates how the confidence-interval length decreases as the calibration dataset grows. The dashed curves correspond to cases where the calibration dataset has symmetric true label counts ($m_0 = m_1$), using $\hat{q}_0 = 0.7$, $\hat{q}_1 = 0.9$, and four test-set accuracies $\hat{p} \in \{0.3, 0.5, 0.7, 0.9\}$. For example, the red dashed

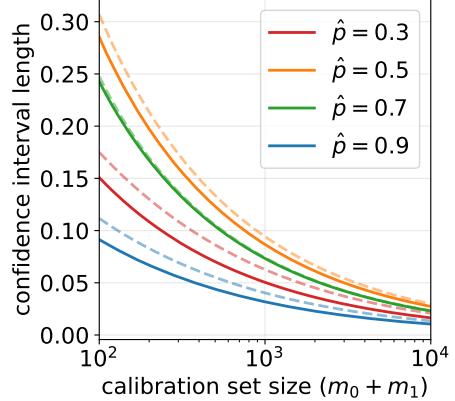


Figure 3: Confidence-interval length across the calibration size for different values of $\hat{p} \in \{0.3, 0.5, 0.7, 0.9\}$, when $\hat{q}_0 = 0.7$, $\hat{q}_1 = 0.9$, and $n \rightarrow \infty$. Dashed lines correspond to using a calibration dataset with symmetric label counts ($m_0 = m_1$), while solid lines correspond to using calibration samples allocated using the adaptive rule introduced in Algorithm 1.

Algorithm 1 Adaptive allocation of calibration samples

Require: Total calibration dataset size m , pilot calibration sample size m_{pilot} with $2m_{\text{pilot}} \leq m$, and the estimate \hat{p} from the test dataset

Ensure: Allocated calibration sample sizes (m_0, m_1)

1: **Pilot calibration.**

2: Collect m_{pilot} calibration examples with true label $z = 0$ and m_{pilot} examples with $z = 1$.

3: Let \tilde{q}_0 and \tilde{q}_1 be the empirical accuracies of the LLM judge on these two subsets:

$$\tilde{q}_0 \leftarrow \frac{\sum_{z=0} \mathbf{1}\{\hat{z} = 0 \mid z = 0\} + 1}{m_{\text{pilot}} + 2}, \quad \tilde{q}_1 \leftarrow \frac{\sum_{z=1} \mathbf{1}\{\hat{z} = 1 \mid z = 1\} + 1}{m_{\text{pilot}} + 2}.$$

4: Compute the estimated error–ratio: $\hat{\kappa} \leftarrow (1 - \tilde{q}_0)/(1 - \tilde{q}_1)$.

5: **Compute adaptive allocation.**

6: Using the approximation in Proposition 2, solve for the provisional allocation:

$$m_1^* \leftarrow \text{round} \left(\frac{m}{1 + (1/\hat{p} - 1)\sqrt{\hat{\kappa}}} \right).$$

7: Enforce pilot size, by setting

$$m_1 \leftarrow \min \{ \max\{m_1^*, m_{\text{pilot}}\}, m - m_{\text{pilot}} \}, \quad m_1 \leftarrow m - m_0.$$

return (m_0, m_1)

curve ($\hat{p} = 0.3$) shows that achieving an interval shorter than 0.1 requires roughly $m_0 = m_1 \approx 200$ calibration examples. Because the calibration dataset is collected independently and its label composition is fully under our control, we can purposely choose asymmetric calibration sizes ($m_0 \neq m_1$). This flexibility is important: the two calibration label types generally contribute asymmetrically to the overall uncertainty, and allocating more samples to the higher-variance side can substantially reduce the dominant source of uncertainty.

Adaptive Allocation to Reduce Confidence-Interval Length Motivated by this observation, we introduce an adaptive allocation procedure in Algorithm 1. The algorithm begins by collecting a small pilot calibration sample (e.g., $m_{\text{pilot}} = 10$ for each label type) to obtain preliminary estimates of $(\tilde{q}_0, \tilde{q}_1)$. These estimates allow us to compute the empirical error ratio $(1 - \tilde{q}_0)/(1 - \tilde{q}_1)$ between the two label types. Using this ratio, together with the naive accuracy \hat{p} from the test set, the algorithm computes the approximate optimal ratio of (m_0, m_1) that minimizes the confidence–interval length in (5). The solid curves in Fig. 3, illustrating the results of adaptive allocation, show that this strategy achieves shorter intervals under a fixed calibration budget compared to the symmetric allocation represented by the dashed lines. The optimality of this allocation rule is established in the following section.

A Python implementation that computes the bias-adjusted estimator $\hat{\theta}$ in (4) and the confidence interval in (5), and applies the adaptive allocation procedure in Algorithm 1 to obtain optimal calibration sizes (m_0, m_1) , is available at <https://github.com/UW-Madison-Lee-Lab/LLM-judge-reporting>.

4 Derivations and Theoretical Guarantees

We derive the point estimator and confidence interval introduced in Sec. 3, showing how they mitigate bias and account for uncertainty from both the test and calibration datasets. We further characterize the theoretical properties of the resulting estimators and establish the optimality of the proposed allocation algorithm.

4.1 Mitigating Bias on Point Estimator

We first analyze how the adjusted estimator $\hat{\theta}$ in (4) mitigates the bias in estimating θ by comparing it with the naive estimator \hat{p} in (2).

The naive estimator \hat{p} is generally biased in estimating the true parameter θ . By the law of total probability, we have

$$\begin{aligned} p &= \Pr(\hat{Z} = 1 \mid Z = 1) \cdot \Pr(Z = 1) + (1 - \Pr(\hat{Z} = 0 \mid Z = 0)) \cdot (1 - \Pr(Z = 1)) \\ &= (q_0 + q_1 - 1) \cdot \theta + (1 - q_0), \end{aligned}$$

which shows that $\mathbb{E}[\hat{p}] = p = \theta$ for all θ only when the LLM judge is perfectly accurate (i.e., $q_0 = q_1 = 1$).

If $q_0 + q_1 < 2$, the expression can be rewritten as

$$\mathbb{E}[\hat{p}] = p = \theta + (2 - q_0 - q_1) \cdot \left(\frac{1 - q_0}{2 - q_0 - q_1} - \theta \right),$$

which makes the sign of the bias explicit: $\mathbb{E}[\hat{p}] > \theta$ whenever $\theta < \frac{1 - q_0}{2 - q_0 - q_1}$ and $\mathbb{E}[\hat{p}] < \theta$ whenever $\theta > \frac{1 - q_0}{2 - q_0 - q_1}$. Thus, \hat{p} induces a bias in estimating θ .

To mitigate this bias, we derive an adjusted estimator. Assuming $q_0 + q_1 > 1$, the above relation can be inverted to express θ as

$$\theta = \frac{p + q_0 - 1}{q_0 + q_1 - 1}. \quad (8)$$

Replacing p , q_0 , and q_1 with their empirical estimates gives the bias-adjusted estimator $\hat{\theta}$ in (4) (Rogan and Gladen, 1978; Lang and Reiczigel, 2014). When q_0 and q_1 are known, substituting their true values into (4) gives an unbiased estimator of θ , which exactly corrects for the LLM's misclassification bias. In practice, however, these parameters are unknown and can be estimated from a calibration dataset. Thus, we use the empirical estimates \hat{q}_0 and \hat{q}_1 defined in Sec. 3, which inevitably introduce some estimation bias. Nevertheless, the following proposition shows that the adjusted estimator $\hat{\theta}$ in (4), constructed with these empirical estimates, still effectively mitigates bias compared to the naive estimator as the size of the calibration dataset increases. All proofs are provided in Appendix A.

Proposition 1. *Suppose that $m := m_0 = m_1$ and $q := q_0 = q_1$, where $0.5 < q \leq 1$. For sufficiently large $m \gtrsim q/(2q - 1)^2$, the absolute bias of $\hat{\theta}$ in (1) is always smaller than that of \hat{p} in (2) for all θ .*

This proposition implies that even when q_0 and q_1 are estimated rather than known, the adjusted estimator $\hat{\theta}$ consistently achieves lower absolute bias than the naive estimator \hat{p} , provided that the calibration dataset is sufficiently large. Moreover, the required sample size depends on the reliability of the LLM judge: when the LLM is highly accurate ($q \approx 1$), only a small calibration dataset is needed for effective bias correction, whereas when the LLM behaves almost randomly ($q \approx 0.5$), a much larger dataset is required to ensure comparable bias reduction.

4.2 Estimating Uncertainty through Confidence Interval

We now quantify the uncertainty of the adjusted estimator $\hat{\theta}$ in (4), where two distinct sources of randomness must be considered: (i) the test dataset used to estimate p , and (ii) the calibration dataset used to estimate q_0 and q_1 .

Asymptotic variance of the point estimator To construct a confidence interval for θ , we first compute the asymptotic variance of the adjusted estimator $\hat{\theta}$. By applying the delta method (Dorfman, 1938; Ver Hoef, 2012), the asymptotic variance of $\hat{\theta}$ is

$$\text{Var}(\hat{\theta}) = \frac{\hat{p}(1 - \hat{p})/n + (1 - \hat{\theta})^2 \cdot \hat{q}_0(1 - \hat{q}_0)/m_0 + \hat{\theta}^2 \cdot \hat{q}_1(1 - \hat{q}_1)/m_1}{(\hat{q}_0 + \hat{q}_1 - 1)^2},$$

where we also use the variance formula for the binomial estimates \hat{p} , \hat{q}_0 , and \hat{q}_1 . A detailed derivation is provided in Sec. A.

Estimating confidence interval Based on this variance, we construct a $(1 - \alpha)$ confidence interval for θ following the “add two successes and two failures” adjusted Wald interval approach (de Laplace, 1820; Agresti and Coull, 1998; Brown et al., 2001; Lang and Reiczigel, 2014). To improve coverage accuracy, we replace \hat{p} , \hat{q}_0 , and \hat{q}_1 with their adjusted versions \tilde{p} , \tilde{q}_0 , and \tilde{q}_1 , as defined in (6). These adjustments can be interpreted as adding one (or $z_\alpha^2/2$) success and one (or $z_\alpha^2/2$) failure to each estimate, such that the interval remains reliable even for small sample sizes (Agresti and Caffo, 2000). Substituting these adjusted estimates yields the Wald-type confidence interval shown in (5).

Furthermore, the adjustment introduces a minor shift in the interval center (i.e., $d\hat{\theta}$ in (7)) due to the presence of \tilde{q}_0 and \tilde{q}_1 in the denominator of $\hat{\theta}$. Although this also slightly affects the interval length, the effect is negligible and thus ignored in the final approximation. Detailed derivations are provided in Lang and Reiczigel (2014).

The optimal allocation of the calibration dataset Lastly, we investigate the optimal allocation of the calibration dataset that minimizes the confidence interval length. For simplicity, consider the case where the LLM’s prediction accuracies \tilde{q}_0 and \tilde{q}_1 are both close to one. Here, \tilde{q}_0 represents the accuracy where the LLM predicts ‘incorrect’ responses as ‘incorrect’, while \tilde{q}_1 denotes the accuracy of predicting ‘correct’ responses as ‘correct’. Consequently, $(1 - \tilde{q}_0)$ and $(1 - \tilde{q}_1)$ correspond to the respective error probabilities of these two conditions.

Proposition 2. *Suppose that \tilde{q}_0 and \tilde{q}_1 are close to 1, and let $\kappa := (1 - \tilde{q}_0)/(1 - \tilde{q}_1)$. Then the minimum length of the confidence interval defined in (5) is achieved when $\tilde{m}_0 \approx (1/\tilde{p} - 1)\sqrt{\kappa} \cdot \tilde{m}_1$.*

This result provides a practical guideline for allocating calibration samples between the two response types, increasing either ‘incorrect’ responses (m_0) or ‘correct’ responses (m_1) in the calibration dataset. When the LLM is less accurate at identifying ‘incorrect’ outputs (that is, when the error ratio κ is large), more calibration data should be assigned to estimating \tilde{q}_0 in order to reduce the variance of $\hat{\theta}$. Conversely, if the LLM performs well overall but the proportion \tilde{p} of ‘correct’ predictions in the test dataset is small, the factor $(1/\tilde{p} - 1)$ amplifies the relative importance of \tilde{m}_0 , suggesting that additional samples should again be allocated to calibration dataset that have the true ‘incorrect’ response to maintain an efficient confidence interval.

To use this allocation rule in practice, we employ the adaptive procedure described in Algorithm 1. The algorithm begins by collecting a small pilot calibration sample to estimate $(\tilde{q}_0, \tilde{q}_1)$ and, in turn, the error-ratio $\hat{\kappa}$. Using the estimate \hat{p} from the test dataset and the value of $\hat{\kappa}$, the calibration sizes (m_0, m_1) are then determined according to the approximate optimal ratio implied by Proposition 2. This procedure ensures that the calibration budget is distributed in a data-driven manner, adapting automatically to the relative difficulty of estimating the two judge accuracy parameters.

5 Empirical Validation

To validate the theoretical results established above, we empirically evaluate the proposed estimator, confidence interval, and calibration allocation strategy through Monte Carlo simulation.

Experimental Setup We evaluate the method under the following parameter configuration. The LLM judge has accuracy parameters $(q_0, q_1) = (0.7, 0.9)$, and the true accuracy varies over $\theta \in \{0, 0.05, 0.10, \dots, 1\}$, resulting in 21 distinct settings. For each setting of (q_0, q_1, θ) , we generate a test dataset of size $n = 1000$ and a calibration dataset of total size $m_0 + m_1 = 500$, using an equal allocation $m_0 = m_1$ unless stated otherwise. We then compute the naive estimator \hat{p} in (2) together with its confidence interval, and further compute the bias-adjusted estimator $\hat{\theta}$ in (4) and the confidence interval in (5). Each configuration is replicated 10,000 times to evaluate estimator behavior, interval coverage, and average interval length. Additional simulations exploring alternative parameter choices are provided in Appendix C.

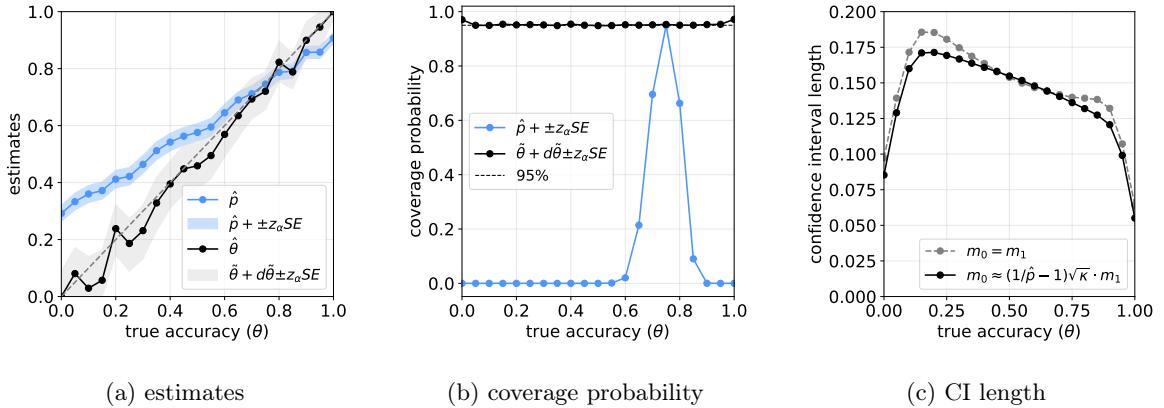


Figure 4: Monte Carlo simulation for estimating θ under an imperfect LLM judge with $(q_0, q_1) = (0.7, 0.9)$. We evaluate estimators across 21 values of $\theta \in \{0, 0.05, 0.10, \dots, 1\}$, each visualized as a single point. Fig. 4a reports the results from a single run, while Fig. 4b and Fig. 4c summarize averages computed over 10,000 Monte Carlo replications. All experiments use a test dataset of size $n = 1000$ and a calibration dataset of size $m_0 + m_1 = 500$, and we use an equal allocation $m_0 = m_1$ for Fig. 4a and Fig. 4c. (a) The naive estimator \hat{p} in (2) generally exhibits bias, while the unbiased estimator $\hat{\theta}$ in (4) closely recovers the true accuracy θ across all values. Shaded regions represent the 95% confidence intervals. (b) Across all θ , the coverage probability of the confidence interval in (5) remains consistently close to the nominal 95% level. (c) Given a fixed calibration budget of $m_0 + m_1 = 500$, we compare two allocation strategies: an equal split ($m_0 = m_1 = 250$) and the allocation proportional to $m_0 \propto (1/\hat{p} - 1)\sqrt{\kappa} \cdot m_1$ by using Algorithm 1. The proposed allocation gives consistently shorter confidence intervals.

Bias Reduction in Point Estimation Fig. 4a compares the naive estimator \hat{p} and the bias-adjusted estimator $\hat{\theta}$ based on a single simulation run. As shown in Sec. 4, \hat{p} exhibits bias, particularly overestimating the true accuracy when the underlying θ is small. In contrast, $\hat{\theta}$ closely aligns with the ground-truth accuracy across all values of θ , demonstrating the bias correction achieved by (4).

Coverage of the Confidence Interval Fig. 4b reports the empirical coverage probability of the confidence interval in (5). Across all values of θ , the coverage remains consistently close to the nominal 95% level, whereas the confidence interval constructed from the naive estimator \hat{p} achieves nearly zero coverage except at a few values of θ . These results confirm that the proposed confidence interval provides reliable uncertainty quantification.

Efficiency of Optimal Calibration Allocation To examine the benefits of optimal calibration allocation, we compare two strategies under a fixed calibration budget of $m_0 + m_1 = 500$: (i) an equal allocation ($m_0 = m_1 = 250$), and (ii) the allocation produced by Algorithm 1, which approximates the optimal ratio derived in Proposition 2. Fig. 4c shows that the adaptive allocation consistently gives shorter confidence intervals than the equal-split baseline.

The improvement becomes more pronounced as \hat{p} moves away from the middle point. For instance, when $\kappa = 1$ and $\tilde{p} = 0.5$, Proposition 2 implies that the optimal allocation satisfies $\tilde{m}_0 \approx \tilde{m}_1$, since in this case $(1/\hat{p} - 1)\sqrt{\kappa} = 1$. As \hat{p} departs from 0.5, however, asymmetric allocations become optimal. This pattern appears clearly in Fig. 4c. In this experiment, we have $\kappa = 3$, and the optimal allocation becomes nearly symmetric ($\tilde{m}_0 \approx \tilde{m}_1$) when $\hat{p} \approx 0.633$, which corresponds to a true accuracy of $\theta \approx 0.557$ via (8). When θ lies moderately far from this value, the adaptive allocation produced by Algorithm 1 results in a shorter interval.

6 Conclusion

In LLM-as-a-judge, noisy evaluations cause a simple point estimate to be biased. The adjustment introduced in this work reduces the bias from imperfect judgments, and the accompanying confidence interval reflects uncertainty from both the evaluation and calibration datasets. To narrow these intervals, focusing on the calibration design, such as the allocation between response types, is preferable to simply increasing the total number of judgments. We hope this work contributes to more reliable and transparent reporting practices in LLM-based evaluation.

Acknowledgements

This work was supported by the National Science Foundation (NSF) Award DMS-2023239, the NSF CAREER Award CCF-2339978, an Amazon Research Award, and a grant from FuriosaAI. In addition, it was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Ministry of Science and ICT (MSIT) (RS-2024-00345351, RS-2024-00408003), by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by MSIT (RS-2023-00259934, RS-2025-02283048), and by the Brain Korea 21 program funded by the Korean Ministry of Education and the NRF (BK21 FOUR, No.5199990913980).

References

- Agresti, A. and Coull, B. A. (2000). Simple and effective confidence intervals for proportions and differences of proportions result from adding two successes and two failures. *The American Statistician*, 54(4):280–288.
- Agresti, A. and Coull, B. A. (1998). Approximate is better than “exact” for interval estimation of binomial proportions. *The American Statistician*, 52(2):119–126.
- Albinet, F. (2025). Why llms can actually judge other llms (and it’s not cheating). <https://franck-albi-net.pla.sh/post/llm-as-a-judge>.
- Angelopoulos, A. N., Bates, S., Fannjiang, C., Jordan, M. I., and Zrnic, T. (2023). Prediction-powered inference. *Science*, 382(6671):669–674.
- Boyeau, P., Angelopoulos, A. N., Li, T., Yosef, N., Malik, J., and Jordan, M. I. (2025). Autoeval done right: Using synthetic data for model evaluation. In *International Conference on Machine Learning*.
- Brown, L. D., Cai, T. T., and DasGupta, A. (2001). Interval Estimation for a Binomial Proportion. *Statistical Science*, 16(2):101 – 133.
- de Laplace, P. S. (1820). *Théorie analytique des probabilités*, volume 7. Courcier.
- Dorfman, R. (1938). A note on the δ -method for finding variance formulae. *Biometric Bulletin*.
- Forman, G. (2008). Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206.
- Fraser, C. (2024). Estimating how many there are of something when you can’t see them all perfectly. <https://colin-fraser.net/posts/2024-11-12-estimating-how-many-there-are-of-something-using-a-classifier>.
- Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, S., Zhang, K., Wang, Y., Gao, W., Ni, L., and Guo, J. (2025). A survey on llm-as-a-judge. arXiv. <https://arxiv.org/abs/2411.15594>.

- Huang, H., Bu, X., Zhou, H., Qu, Y., Liu, J., Yang, M., Xu, B., and Zhao, T. (2025). An empirical study of LLM-as-a-judge for LLM evaluation: Fine-tuned judge model is not a general substitute for GPT-4. In *Findings of the Association for Computational Linguistics*.
- Koo, R., Lee, M., Raheja, V., Park, J. I., Kim, Z. M., and Kang, D. (2024). Benchmarking cognitive biases in large language models as evaluators. In *Findings of the Association for Computational Linguistics*.
- Lang, Z. and Reiczigel, J. (2014). Confidence limits for prevalence of disease adjusted for estimated sensitivity and specificity. *Preventive Veterinary Medicine*, 113(1):13–22.
- Li, D., Jiang, B., Huang, L., Beigi, A., Zhao, C., Tan, Z., Bhattacharjee, A., Jiang, Y., Chen, C., Wu, T., Shu, K., Cheng, L., and Liu, H. (2025). From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In *Empirical Methods in Natural Language Processing*.
- Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. (2023). G-eval: NLG evaluation using gpt-4 with better human alignment. In *Empirical Methods in Natural Language Processing*.
- Rogan, W. J. and Gladen, B. (1978). Estimating prevalence from the results of a screening test. *American Journal of Epidemiology*, 107(1):71–76.
- Ver Hoef, J. M. (2012). Who invented the delta method? *The American Statistician*, 66(2):124–127.
- Wang, J., Liang, Y., Meng, F., Sun, Z., Shi, H., Li, Z., Xu, J., Qu, J., and Zhou, J. (2023). Is ChatGPT a good NLG evaluator? a preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*.
- Wang, P., Li, L., Chen, L., Cai, Z., Zhu, D., Lin, B., Cao, Y., Kong, L., Liu, Q., Liu, T., and Sui, Z. (2024). Large language models are not fair evaluators. In *Annual Meeting of the Association for Computational Linguistics*.
- Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., Zhang, H., Gonzalez, J. E., and Stoica, I. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*.

A Proofs

This section provides the proofs deferred from the main paper. We first derive the variances of the estimators \hat{p} in (2) and $\hat{\theta}$ in (4), followed by proofs of the propositions stated in the main text.

A.1 Deriving the Variance of Estimators

Because p follows a binomial distribution, the variance of \hat{p} is

$$\text{Var}(\hat{p}) = \hat{p}(1 - \hat{p})/n.$$

Similarly, we have $\text{Var}(\hat{q}_0) = \hat{q}_0(1 - \hat{q}_0)/m_0$ and $\text{Var}(\hat{q}_1) = \hat{q}_1(1 - \hat{q}_1)/m_1$.

We now derive the asymptotic variance of $\hat{\theta}$ using the delta method (Dorfman, 1938; Ver Hoef, 2012) for $\hat{\theta} = \frac{\hat{p} + \hat{q}_0 - 1}{\hat{q}_0 + \hat{q}_1 - 1}$. The first order derivatives with respect to \hat{p} , \hat{q}_0 , and \hat{q}_1 are

$$\frac{\partial \hat{\theta}}{\partial \hat{p}} = \frac{1}{\hat{q}_0 + \hat{q}_1 - 1}, \quad \frac{\partial \hat{\theta}}{\partial \hat{q}_0} = \frac{1 - \hat{\theta}}{\hat{q}_0 + \hat{q}_1 - 1}, \quad \frac{\partial \hat{\theta}}{\partial \hat{q}_1} = \frac{-\hat{\theta}}{\hat{q}_0 + \hat{q}_1 - 1}.$$

Assuming independence between the test dataset and the calibration dataset, the delta method gives

$$\text{Var}(\hat{\theta}) = \frac{\hat{p}(1 - \hat{p})/n + (1 - \hat{\theta})^2 \cdot \hat{q}_0(1 - \hat{q}_0)/m_0 + \hat{\theta}^2 \cdot \hat{q}_1(1 - \hat{q}_1)/m_1}{(\hat{q}_0 + \hat{q}_1 - 1)^2}.$$

A.2 Proofs of Propositions

Proposition A.1. Suppose that $m := m_0 = m_1$ and $q := q_0 = q_1$, where $0.5 < q \leq 1$. For sufficiently large $m \gtrsim q/(2q - 1)^2$, the absolute bias of $\hat{\theta}$ in (1) is always smaller than that of \hat{p} in (2) for all θ .

Proof. First, note that the bias of \hat{p} in (2) is

$$\begin{aligned} \mathbb{E}[\hat{p}] - \theta &= (q_0 + q_1 - 1)\theta + (1 - q_0) - \theta \\ &= -(2\theta - 1)(q - 1). \end{aligned}$$

Next, consider the bias of $\hat{\theta}$ in (1). By the second-order delta method, we have

$$\begin{aligned} \mathbb{E}[\hat{\theta}] &\approx \frac{p + q_0 - 1}{q_0 + q_1 - 1} + \frac{1}{2} \left(-\frac{2(q_1 - p)}{(q_0 + q_1 - 1)^3} \cdot \frac{q_0(1 - q_0)}{m_0} + \frac{2(p + q_0 - 1)}{(q_0 + q_1 - 1)^3} \cdot \frac{q_1(1 - q_1)}{m_1} \right) \\ &= \theta - \frac{(q_1 - p)}{(q_0 + q_1 - 1)^3} \cdot \frac{q_0(1 - q_0)}{m_0} + \frac{(p + q_0 - 1)}{(q_0 + q_1 - 1)^3} \cdot \frac{q_1(1 - q_1)}{m_1}, \end{aligned}$$

which implies

$$\begin{aligned} \mathbb{E}[\hat{\theta}] - \theta &\approx \frac{-(1 - \theta)q_0(1 - q_0)/m_0 + \theta q_1(1 - q_1)/m_1}{(q_0 + q_1 - 1)^2} \\ &= \frac{1}{m} \cdot \frac{q}{(2q - 1)^2} (2\theta - 1)(1 - q). \end{aligned}$$

Hence, for sufficiently large m satisfying $m \gtrsim q/(2q - 1)^2$, we conclude the following for all θ :

$$|\mathbb{E}[\hat{\theta}] - \theta| \approx \left| \frac{1}{m} \cdot \frac{q}{(2q - 1)^2} \right| \cdot |(2\theta - 1)(1 - q)| < |(2\theta - 1)(1 - q)| = |\mathbb{E}[\hat{p}] - \theta|.$$

□

Proposition A.2. Suppose that \tilde{q}_0 and \tilde{q}_1 are close to 1, and let $\kappa := (1 - \tilde{q}_0)/(1 - \tilde{q}_1)$. Then the minimum length of the confidence interval defined in (5) is achieved when $\tilde{m}_0 \approx (1/\tilde{p} - 1)\sqrt{\kappa} \cdot \tilde{m}_1$.

Proof. The length of the confidence interval in (5) is given by

$$\begin{aligned} 2z_\alpha &\sqrt{\frac{\tilde{p}(1 - \tilde{p})/\tilde{n} + (1 - \tilde{\theta})^2 \cdot \tilde{q}_0(1 - \tilde{q}_0)/\tilde{m}_0 + \tilde{\theta}^2 \cdot \tilde{q}_1(1 - \tilde{q}_1)/\tilde{m}_1}{(\tilde{q}_0 + \tilde{q}_1 - 1)^2}} \\ &\propto \sqrt{(1 - \tilde{\theta})^2 \cdot \tilde{q}_0(1 - \tilde{q}_0)/\tilde{m}_0 + \tilde{\theta}^2 \cdot \tilde{q}_1(1 - \tilde{q}_1)/\tilde{m}_1} \\ &\propto \sqrt{(\tilde{q}_1 - \tilde{p})^2 \cdot \tilde{q}_0(1 - \tilde{q}_0)/\tilde{m}_0 + (\tilde{p} + \tilde{q}_0 - 1)^2 \cdot \tilde{q}_1(1 - \tilde{q}_1)/\tilde{m}_1}. \end{aligned}$$

By the arithmetic–geometric mean inequality, the minimum condition is satisfied when

$$\frac{\tilde{m}_0}{\tilde{m}_1} = \frac{|\tilde{q}_1 - \tilde{p}|}{|\tilde{p} + \tilde{q}_0 - 1|} \sqrt{\frac{\tilde{q}_0(1 - \tilde{q}_0)}{\tilde{q}_1(1 - \tilde{q}_1)}} \approx \frac{|1 - \tilde{p}|}{|\tilde{p}|} \sqrt{\frac{1 - \tilde{q}_0}{1 - \tilde{q}_1}} = (1/\tilde{p} - 1)\sqrt{\kappa},$$

where the approximation holds under the assumption that \tilde{q}_0 and \tilde{q}_1 are close to 1. \square

B Code

All code used for this paper, including a plug-in Python implementation of the introduced method for LLM-as-a-judge evaluation, is available in the public GitHub repository (<https://github.com/UW-Madison-Lee-Lab/LLM-judge-reporting>). To make this appendix self-contained, we provide below the key functions that compute the bias-adjusted estimator and its confidence interval, corresponding to the method described in Sec. 3.

```
from math import sqrt
from scipy.stats import norm

def clip(x, low=0.0, high=1.0):
    return max(low, min(high, x))

def point_estimator(p, q0, q1):
    """Compute the adjusted point estimate."""
    th = (p+q0-1)/(q0+q1-1)
    return clip(th)

def confidence_interval(p, q0, q1, n, m0, m1, alpha=0.05):
    """Compute the adjusted (1 - alpha) confidence interval."""
    z = norm.ppf(1-alpha/2)
    p, q0, q1 = (n*p+z**2/2)/(n+z**2), (m0*q0+1)/(m0+2), (m1*q1+1)/(m1+2)
    n, m0, m1 = n+z**2, m0+2, m1+2
    th = (p+q0-1)/(q0+q1-1)
    dth = 2*z**2*(-(1-th)*q0*(1-q0)/m0+th*q1*(1-q1)/m1)
    se = sqrt(p*(1-p)/n+(1-th)**2*q0*(1-q0)/m0+th**2*q1*(1-q1)/m1)/(q0+q1-1)
    return clip(th+dth-z*se), clip(th+dth+z*se)
```

Figure 5: Python code implementation of the adjustment method described in Sec. 3 that computes the bias-adjusted point estimate and the $(1 - \alpha)$ confidence interval for the true accuracy θ . The inputs p , q_0 , and q_1 are empirical estimates from the test and calibration datasets.

C Additional Results on Monte Carlo Simulation

To complement the main simulation results presented in Fig. 4, we report an extensive set of Monte Carlo experiments conducted across multiple configurations of the test dataset size $n \in \{200, 1000\}$, the calibration sizes $m_0 + m_1 \in \{200, 500\}$, and the judge reliability parameters $(q_0, q_1) \in \{(0.9, 0.9), (0.7, 0.7), (0.9, 0.7), (0.7, 0.9)\}$. The remaining aspects of the simulation design follow the same setup as in the main text.

Across all combinations of $(n, m_0 + m_1, q_0, q_1)$, the qualitative findings observed in the main simulation persist. Bias correction consistently improves estimation accuracy, empirical coverage attains the nominal level, and optimized calibration allocation yields shorter confidence intervals.

Below we present the complete collection of results. Each figure corresponds to one configuration (n, m_0, m_1, q_0, q_1) and includes three subplots.

C.1 Results for $n = 200$ and $m_0 + m_1 = 200$

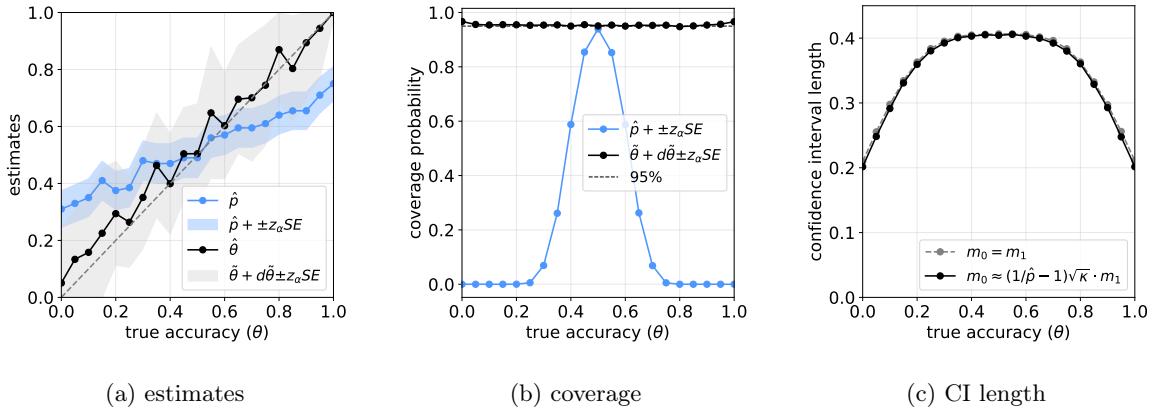


Figure 6: Monte Carlo results for $(n, m_0 + m_1, q_0, q_1) = (200, 200, 0.7, 0.7)$.

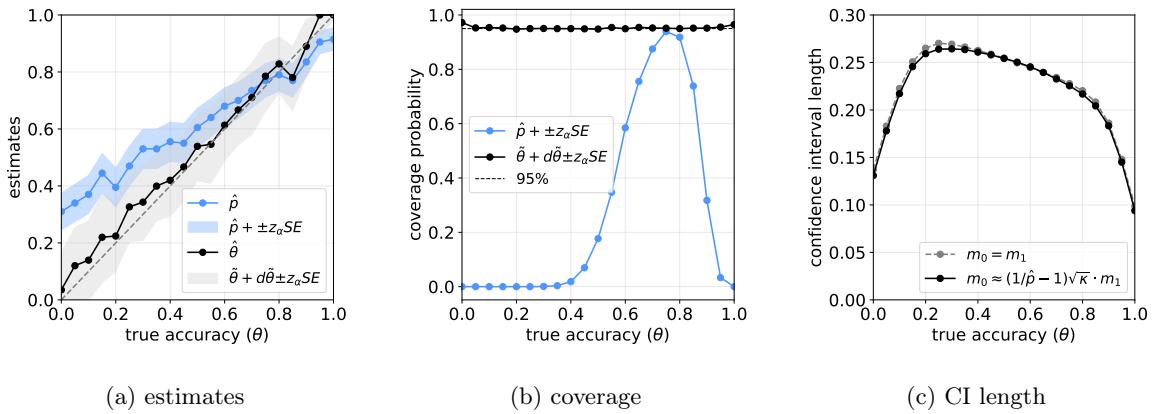


Figure 7: Monte Carlo results for $(n, m_0 + m_1, q_0, q_1) = (200, 200, 0.7, 0.9)$.

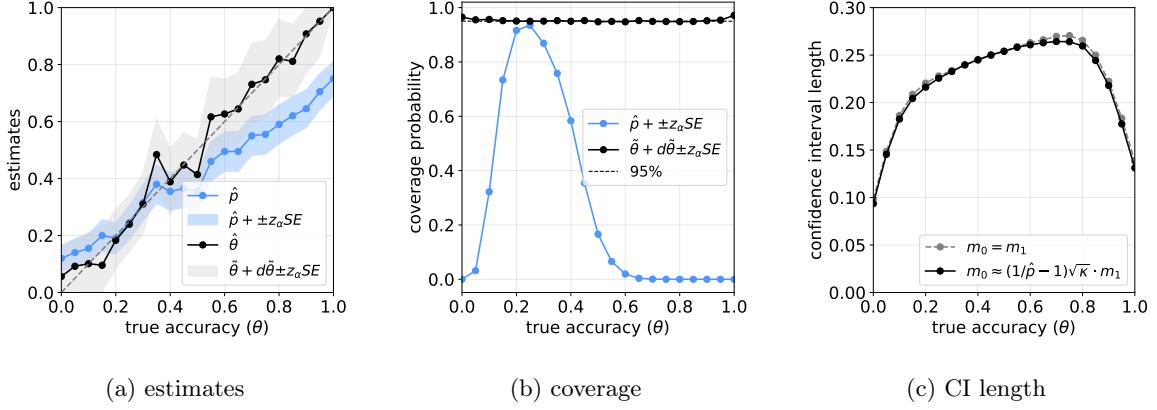


Figure 8: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (200, 200, 0.9, 0.7)$.

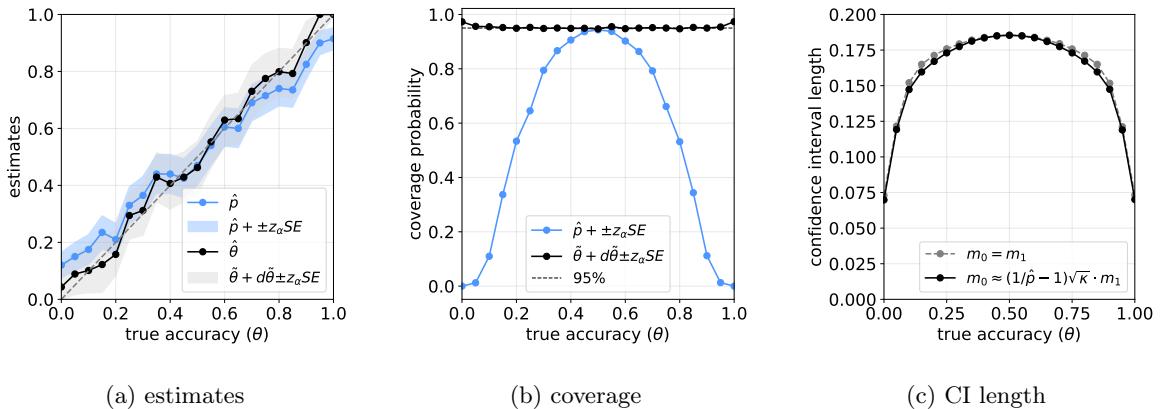


Figure 9: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (200, 200, 0.9, 0.9)$.

C.2 Results for $n = 200$ and $m_0 + m_1 = 500$

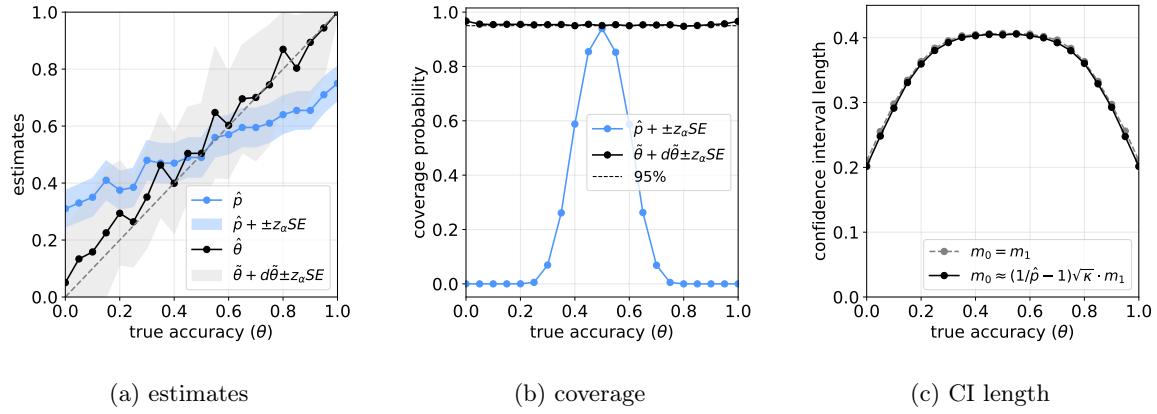


Figure 10: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (200, 500, 0.7, 0.7)$.

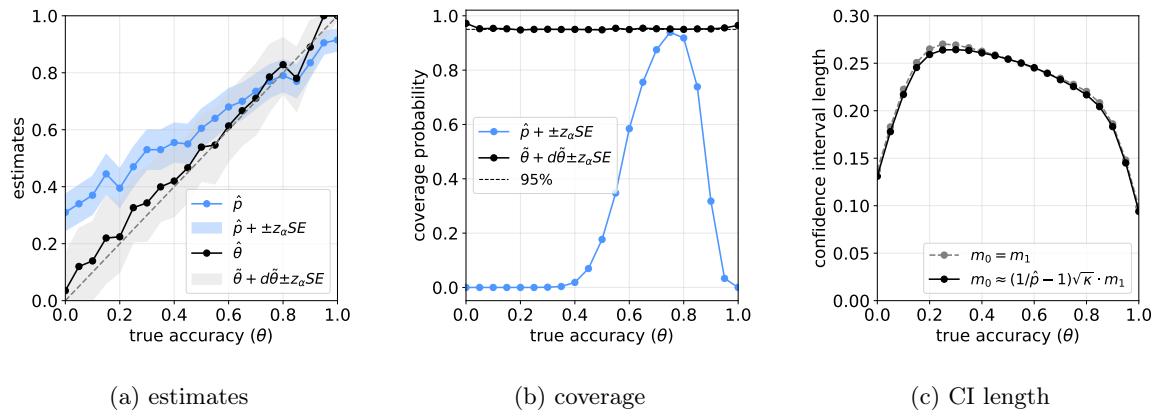


Figure 11: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (200, 500, 0.7, 0.9)$.

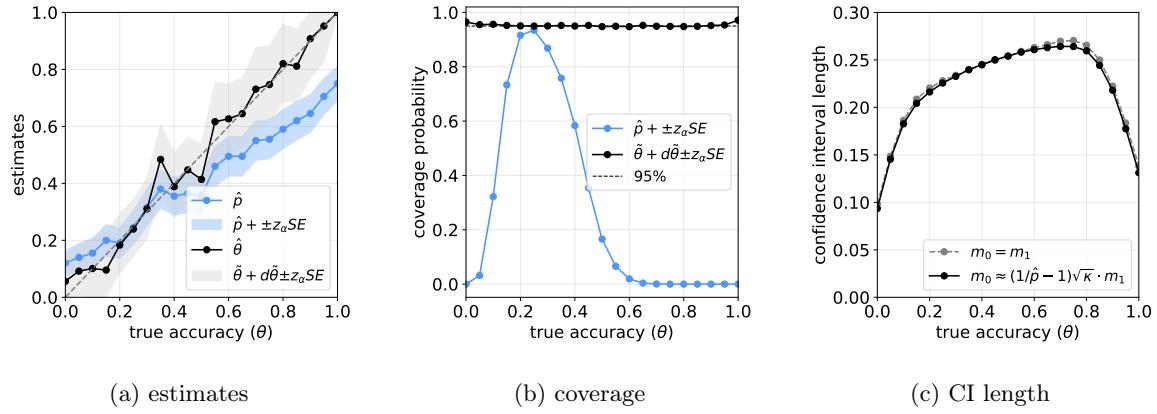


Figure 12: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (200, 500, 0.9, 0.7)$.

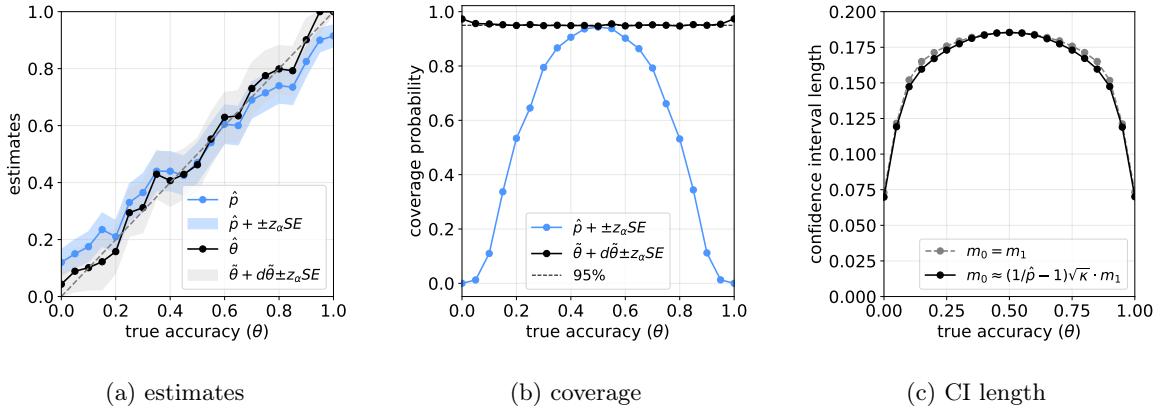


Figure 13: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (200, 500, 0.9, 0.9)$.

C.3 Results for $n = 1000$ and $m_0 + m_1 = 200$

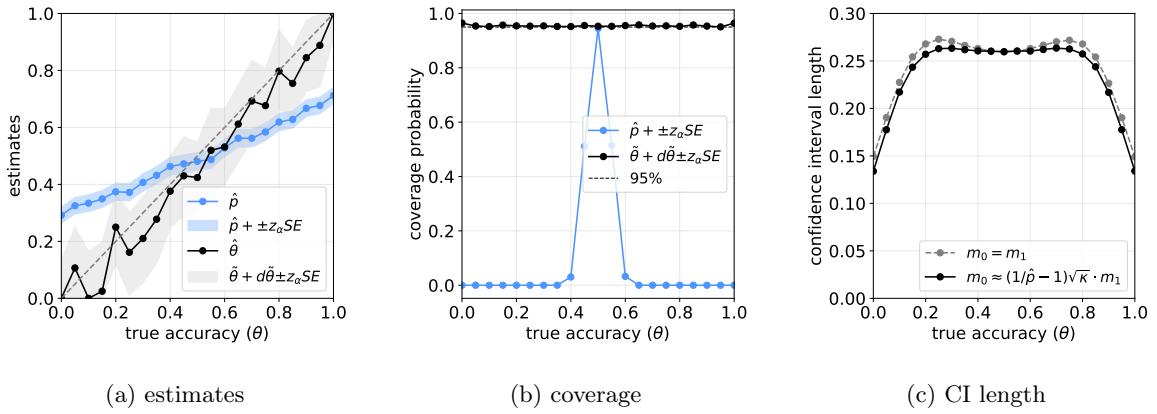


Figure 14: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (1000, 200, 0.7, 0.7)$.

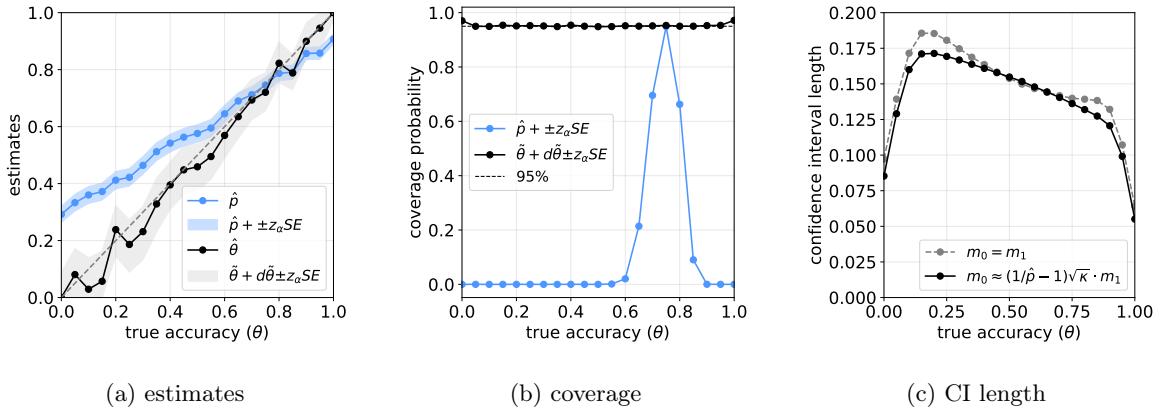


Figure 15: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (1000, 200, 0.7, 0.9)$.

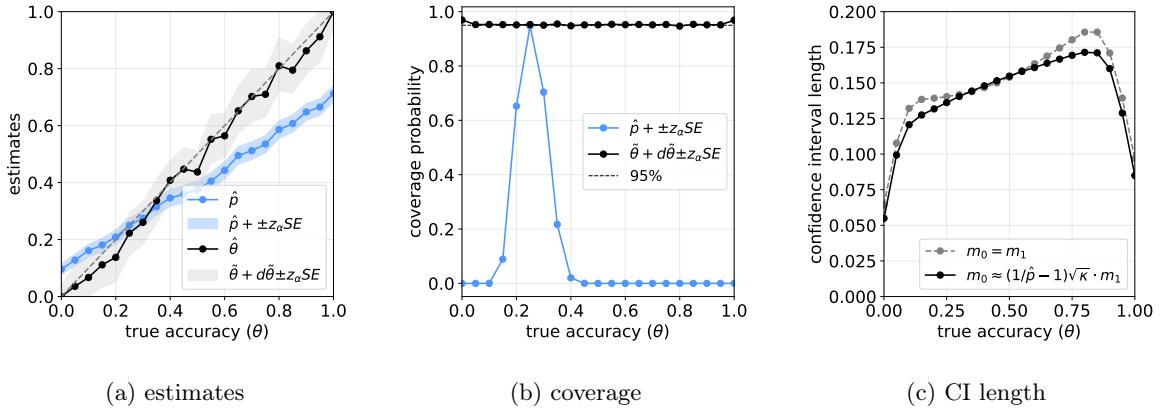


Figure 16: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (1000, 200, 0.9, 0.7)$.

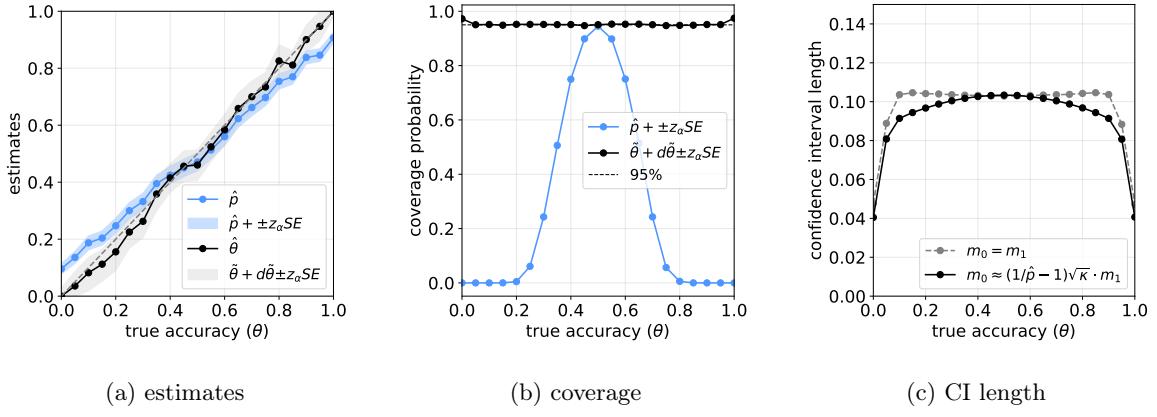


Figure 17: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (1000, 200, 0.9, 0.9)$.

C.4 Results for $n = 1000$ and $m_0 + m_1 = 500$

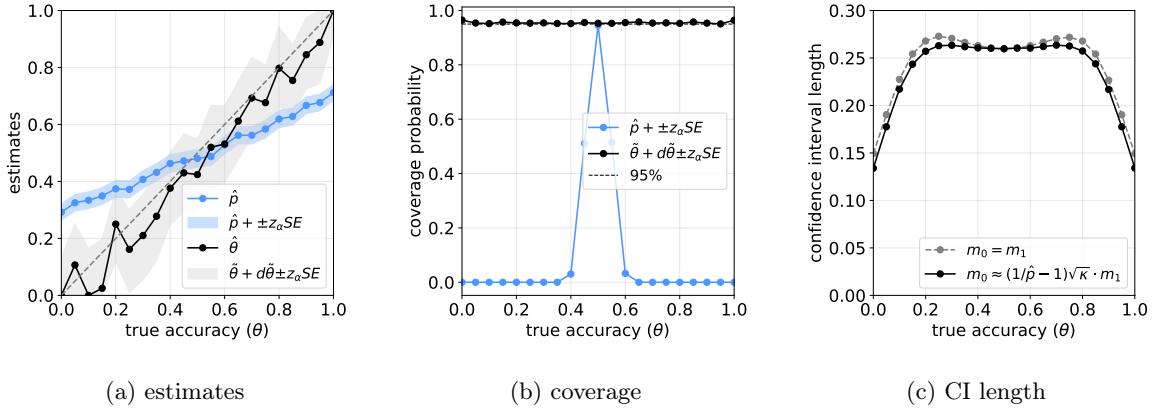


Figure 18: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (1000, 500, 0.7, 0.7)$.

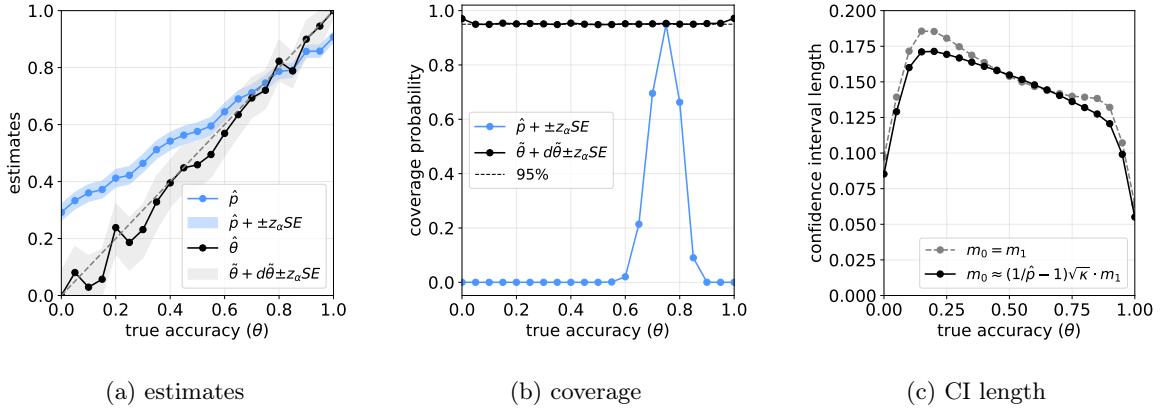


Figure 19: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (1000, 500, 0.7, 0.9)$.

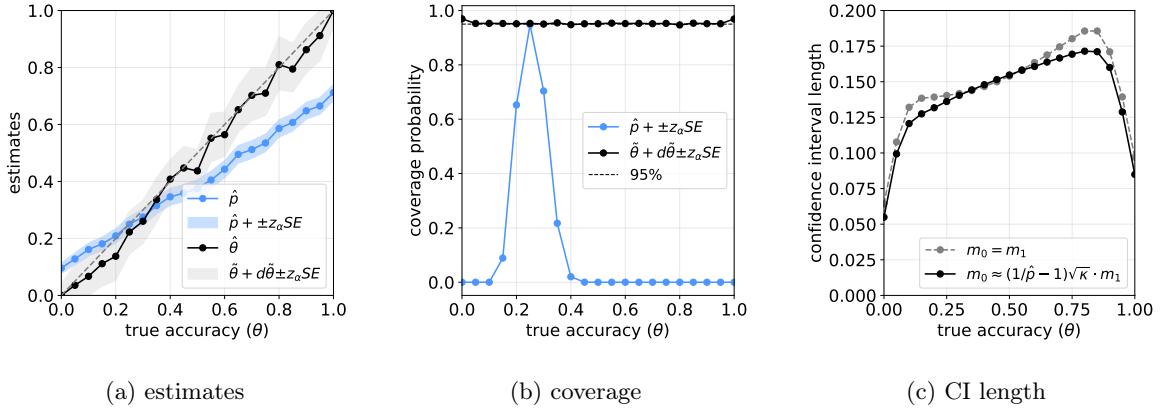


Figure 20: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (1000, 500, 0.9, 0.7)$.

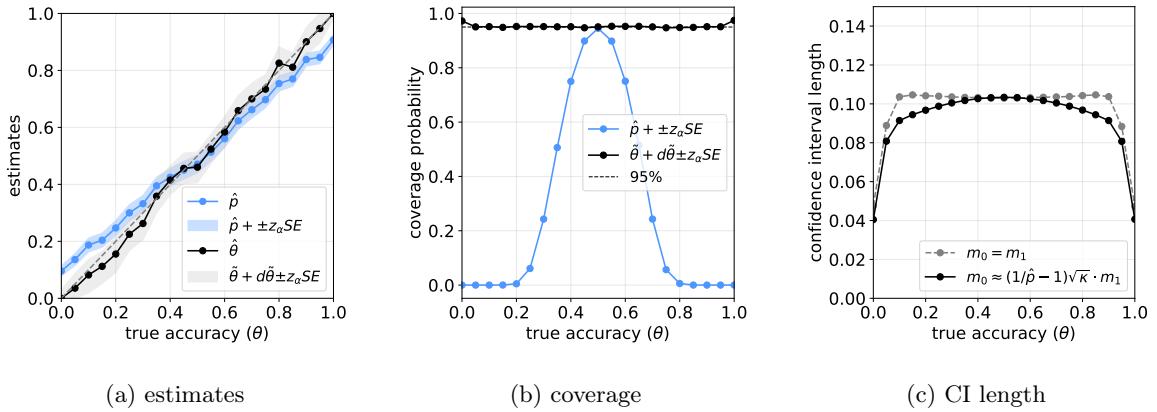


Figure 21: Monte Carlo results for $(n, m_0+m_1, q_0, q_1) = (1000, 500, 0.9, 0.9)$.