EFFICIENT BAYESIAN INFERENCE FROM NOISY PAIRWISE COMPARISONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating generative models is challenging because standard metrics often fail to reflect human preferences. Human evaluations are more reliable but costly and noisy, as participants vary in expertise, attention, and diligence. Pairwise comparisons improve consistency, yet aggregating them into overall quality scores requires careful modeling. Bradley-Terry-based methods update item scores from comparisons, but existing approaches either ignore rater variability or lack convergence guarantees, limiting robustness and interpretability. We introduce BBQ, a Bayesian Bradley-Terry variant that explicitly models rater quality, downweighting or removing unreliable participants, and provides guaranteed monotonic likelihood convergence through an Expectation-Maximization algorithm. Empirical results show that BBQ achieves faster convergence, well-calibrated uncertainty estimates, and more robust, interpretable rankings compared to baseline Bradley-Terry models, even with noisy or crowdsourced raters. This framework enables more reliable and cost-effective human evaluation of generative models.

1 INTRODUCTION

Evaluating generative models is challenging, particularly for large language models (LLMs) and image generators, where standard metrics often fail to reflect human preferences. Metrics such as BLEU (Papineni et al., 2002) and perplexity (Jelinek, 1998) for LLMs, or PSNR (Gonzalez, 2009), MS-SSIM (Wang et al., 2003), and FID (Heusel et al., 2017) for image models, provide only limited insight into perceived quality (Mentzer et al., 2020; CLIC, 2025; Chiang et al., 2024). As a result, human evaluations remain indispensable for establishing meaningful rankings between models. Accordingly, leaderboards and comparative studies of large language models and Learned Image Compression (LIC) methods place strong emphasis on human preference data.

However, human evaluations are expensive and time-consuming, making it crucial to design protocols that are efficient while minimizing subjectivity and noise. In this context, pairwise comparisons, in which participants choose between two items rather than providing absolute scores, represent an effective and practical form of human evaluation (Zerman et al., 2018; Wang et al., 2023). They are generally easier for participants and produce more consistent judgments. Collecting all pairwise comparisons is infeasible because the number of pairs grows quadratically with the number of items. With only a limited set of comparisons, simple statistics such as win rates are not sufficient to derive overall quality scores, because a win rate depends on the quality of the items it is compared against.

To produce an overall ranking that can be displayed on leaderboards, pairwise comparisons must be aggregated effectively. This motivates the development of robust and efficient aggregation methods. The Bradley-Terry (BT) (Bradley & Terry, 1952) model addresses this problem by iteratively updating item scores based on comparison outcomes. It has become a standard approach for aggregating pairwise judgments across domains.

Despite its usefulness, the standard Bradley–Terry model has several limitations in practice. There are multiple algorithms for estimating BT parameters, including iterative scaling methods (Ford Jr, 1957), gradient descent, and the minorization maximization (MM) algorithm (Hunter, 2004). The MM algorithm guarantees convergence to the global maximum for the basic BT model, whereas gradient-based methods, which are closely related to the Elo rating system, are widely used in practice but provide no such guarantees (Hunter, 2004). For extended BT models that incorporate factors such as home field advantage (Agresti, 2010), multiple comparisons (Plackett, 1975; Luce et al., 1959), or ties (Rao & Kupper, 1967), even MM algorithms may converge only to local optima (Hunter, 2004). A further challenge is that human evaluations are inherently noisy: participants may rush, guess, or lose focus, leading to uneven reliability across raters. If such variability is ignored, unreliable raters can distort item scores and reduce ranking stability.

We propose a Bayesian Bradley–Terry variant that jointly models item quality and rater reliability. To the best of our knowledge, no prior work combines the BT framework with a Bayesian formulation that explicitly models rater quality and provides closed-form EM updates. Previous approaches that account for noisy raters typically rely on gradient-based optimization without convergence guarantees, whereas our method ensures stable convergence and yields interpretable parameters. The Bayesian framework addresses epistemic uncertainty, provides regularization, and ensures stable convergence of the estimation procedure. To capture variability in participant behavior, our method introduces rater-specific parameters that reflect how consistent or trustworthy each participant is, allowing the model to adjust the influence of individual comparisons. We derive an EM algorithm that efficiently estimates item and rater parameters via a latent-variable formulation, guaranteeing monotonic likelihood improvement. This approach allows partially reliable raters to contribute meaningful information while reducing the impact of inattentive or inconsistent participants. The Bayesian priors further regularize the estimates, preventing overfitting when many raters and parameters are involved, and leading to stable and generalizable score estimates. By modeling rater quality and adopting a Bayesian framework, our method also provides uncertainty estimates for item scores, facilitating more interpretable rankings and robust comparisons across studies.

We demonstrate the effectiveness of our approach on human evaluation datasets for generative models, showing faster convergence, improved robustness to noisy comparisons, and more consistent rankings compared to standard BT and naive aggregation methods. Beyond generative model evaluation, our framework can be applied to any scenario where noisy pairwise comparisons must be converted into reliable global rankings. Overall, our work contributes to more cost-efficient, interpretable, and reproducible human studies for evaluating AI-generated content.

2 Related Work

Human studies often require ranking items. Presenting participants with a choice between two items rather than a single item with a score increases sensitivity to subtle differences and reduces variability in responses (Zerman et al., 2018; Wang et al., 2023). Consequently, many leaderboards for machine learning models rely on pairwise comparisons. In such setups, participants indicate their preference between two alternatives, and these preferences are then aggregated to produce overall rankings. Notable examples include the Chatbot Arena (Chiang et al., 2024) and the CLIC image compression challenge (CLIC, 2025), which use pairwise comparisons and combine them using variants of the Bradley-Terry (BT) model.

The Bradley-Terry model (Bradley & Terry, 1952) was originally developed to rank competitors in sports. It provides a probabilistic framework to estimate the likelihood that one item is preferred over another. The model converts pairwise comparisons into a ranking, making it a cornerstone in studies across games, consumer preferences, and other applications. Several extensions have been proposed to broaden its applicability. For instance, the Plackett-Luce model generalizes the BT framework from pairwise comparisons to rankings over multiple items (Plackett, 1975; Luce et al., 1959), defining a probability distribution over permutations by multiplying successive BT probabilities (Luce et al., 1959). Other modifications address specific contexts, such as modeling home-field advantages (Agresti, 2010), incorporating ties (Rao & Kupper, 1967), or handling comparisons between groups instead of individuals (Huang et al., 2006).

Maximum likelihood estimation (MLE) is commonly used to infer the parameters of the basic BT model. Zermelo (1929) introduced an iterative approach to compute these estimates, which has become widely adopted. Later, Lange et al. (2000) demonstrated that this procedure is a particular case of minorization-maximization (MM) algorithms, which iteratively optimize surrogate functions to reach a local maximum of the likelihood. Hunter (2004) extended MM algorithms to generalized BT models and established conditions guaranteeing convergence to the MLE. For the classical BT model, the optimization is convex, ensuring convergence to the global optimum (Hunter, 2004). However, for extensions such as home-field advantage, multiple comparisons, or ties, MM algorithms may converge only to a local maximum (Hunter, 2004).

Bayesian formulations of the BT model have been explored to incorporate prior knowledge and regularization (Adams, 2005; Guiver & Snelson, 2009; Caron & Doucet, 2012). Caron & Doucet (2012) showed that MM algorithms can be interpreted as Expectation-Maximization (EM) procedures, enabling Bayesian inference via Gibbs sampling. This formulation provides tractable complete-data likelihoods and ensures convergence of the resulting Markov Chain Monte Carlo methods. The Bayesian perspective smooths posterior estimates, reducing susceptibility to local maxima, and allows for uncertainty quantification in the estimated item skills.

Annotator quality models represent a critical extension addressing the assumption that all comparisons are equally reliable. In crowdsourced evaluations, participant quality can vary widely, motivating approaches that explicitly account for annotator reliability (Chen et al., 2013). Chen et al. (2013) proposed a Bayesian model that jointly estimates both item quality and rater reliability using annotator-specific parameters. Their approach places Gaussian priors on item and rater parameters, and incorporates scaling factors in the likelihood to modulate individual annotator influence. They perform posterior inference using gradient-based optimization, which can be challenging in high-dimensional spaces due to potential local optima. Moreover, gradient-based methods have no guarantees of convergence. In contrast, our proposed EM-based method provides guaranteed monotonic likelihood improvement. The Elo-based rater model developed by Google Research implements a simplified version of this approach. It is widely used in the Challenge on Learned Image Compression (CLIC) (CLIC, 2025) and related research (Mentzer et al., 2020; Ballé et al., 2025).

To the best of our knowledge, no prior work combines the Bradley–Terry model with a Bayesian formulation that explicitly models rater quality while also providing closed-form EM updates. In contrast to Chen et al. (2013), who use Gaussian priors and gradient-based optimization without convergence guarantees, our approach leverages conjugate priors and an EM algorithm that ensures monotonic likelihood improvement. Compared to the Elo-based rater model widely used in CLIC (CLIC, 2025), which is a simplified heuristic relying on Elo-style updates, our method provides a principled Bayesian treatment with uncertainty estimates and interpretable rater-quality parameters.

3 METHODOLOGY

3.1 BRADLEY-TERRY MODEL WITH RATER QUALITY

The objective of converting a set of noisy pairwise comparisons into a reliable ranking of items is a fundamental problem in machine learning and statistics. Consider a set of K items that are repeatedly compared with one another in pairs by a set of R raters. The data, which we denote as D, consists of the outcomes of these comparisons. For two items i and j of this set, Bradley & Terry (1952) suggested the following model:

$$P(i \text{ beats } j) = \frac{\lambda_i}{\lambda_i + \lambda_j} \tag{1}$$

where $\lambda_k > 0$ is a parameter associated with item $k \in \{1, 2, \ldots, K\}$ that represents its skill rating. This model provides a clear and interpretable way to infer item scores from a set of observed wins and losses. However, it operates under the simplifying assumption that all comparisons are equally reliable. In the context of human evaluation, this assumption is often violated. Participants may exhibit varying levels of expertise, attention, or diligence, leading to unreliable and inconsistent judgments.

To account for varying rater reliability, we introduce a rater-specific quality parameter $q_r \in [0, 1]$. Intuitively, with probability q_r, rater r makes an informed judgment following the Bradley-Terry model. With probability $1 - q_r$, the rater guesses randomly, as if flipping a fair coin between the two items. This leads to the following mixture model for the probability that rater r ranks item i above item j:

$$P(r \text{ ranks } i \text{ above } j) = q_r \left(\frac{\lambda_i}{\lambda_i + \lambda_j} \right) + (1 - q_r) \left(\frac{1}{2} \right) \tag{2}$$

The log-likelihood of the data is given by:

$$\begin{aligned}\log P(D \mid \lambda, q) = \sum_{r=1}^{R}\sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\Bigg[& w_{r,ij}\log\left(q_r\frac{\lambda_i}{\lambda_i+\lambda_j}+(1-q_r)\frac{1}{2}\right)\\ &+(n_{r,ij}-w_{r,ij})\log\left(q_r\frac{\lambda_j}{\lambda_i+\lambda_j}+(1-q_r)\frac{1}{2}\right)\Bigg],\end{aligned} \tag{3}$$

where $n_{r,ij}$ denotes the total number of comparisons between items i and j by rater r, and $w_{r,ij}$ is the number of times rater r ranks item i above item j. There is no closed-form maximum likelihood estimator for this likelihood function, so the optimal parameters λ and q cannot be derived analytically. While an iterative optimization method like gradient descent could be used, it offers no guarantee of convergence. On the other hand, the Expectation–Maximization (EM) algorithm avoids learning-rate tuning, and guarantees a monotonic increase of the observed-data likelihood and convergence to a stationary point (Dempster et al., 1977).

3.2 Thurstonian Interpretation

Following Caron & Doucet (2012), we interpret the Bradley-Terry model under a Thurstonian framework (Diaconis, 1988). In this perspective, a comparison between items i and j is conceptualized as a race, where each item has a random arrival time, Y_i and Y_j, respectively. These arrival times are assumed to follow exponential distributions:

$$Y_i \sim \mathcal{E}(\lambda_i), \quad Y_j \sim \mathcal{E}(\lambda_j), \tag{4}$$

and the item with the smaller arrival time is declared the winner. This leads directly to the standard Bradley-Terry probability:

$$P(i\,\text{beats}\,j) = P(Y_i < Y_j) = \frac{\lambda_i}{\lambda_i+\lambda_j}. \tag{5}$$

For the EM algorithm, we introduce latent variables to simplify the complete-data likelihood. First, we define an indicator variable

$$A_{r,ij}^{(c)} \sim \text{Bernoulli}(q_r), \tag{6}$$

which denotes whether the c-th comparison of items i and j by rater r follows the Bradley-Terry model. Using these indicators, we define the latent variable $Z_{r,ij}$ as the sum of the minimal arrival times across the $n_{r,ij}$ comparisons by rater r:

$$Z_{r,ij} = \sum_{c=1}^{n_{r,ij}} A_{r,ij}^{(c)}\ \min(Y_i^{(c)}, Y_j^{(c)}). \tag{7}$$

Conditioned on $m_{r,ij} = \sum_{c=1}^{n_{r,ij}} A_{r,ij}^{(c)}$, the variable $Z_{r,ij}$ follows a Gamma distribution,

$$Z_{r,ij} \mid m_{r,ij} \ \sim\ \Gamma\big(m_{r,ij}, \lambda_i+\lambda_j\big), \tag{8}$$

where $\Gamma(\alpha,\beta)$ denotes the Gamma distribution with shape parameter α and inverse scale β. This Gamma-distributed latent variable formulation allows for a tractable EM update while accounting for rater-specific quality.

3.3 Expectation-Maximization Updates

The EM algorithm is an iterative method for finding maximum a posteriori (MAP) estimates for our model parameters, λ and q, by treating the rater's quality and the unobserved arrival times from the Thurstonian interpretation as latent variables. The algorithm is guaranteed to converge to a stationary point of the posterior distribution.

First, we specify prior distributions for the parameters. The item skills λ_k are assigned a Gamma prior, $\lambda_k \sim \Gamma(a,b)$, which is a conjugate prior for the exponential distribution. Each rater's quality parameter, q_r, is given a Beta prior, $q_r \sim B(\alpha,\beta)$.

The core of the EM algorithm is the iterative maximization of the expected complete-data log-posterior, conditioned on the current parameter estimates (λ^*, q^*). The algorithm proceeds by alternating between two steps until convergence:

E-step: Expectation In the E-step, we compute the expected complete-data log-posterior, a function we denote as Q. This function represents the expected value of the log-posterior of all observed and latent variables, given our observed data and the current parameter estimates from the previous iteration. It is defined as:

$$Q(\lambda, q \mid \lambda^*, q^*) = \mathbb{E}_{A,Z|D,\lambda^*,q^*}\Big[\ell_c(\lambda, q; D, Z, A) + \log P(\lambda) + \log P(q)\Big]. \tag{9}$$

The complete-data log-likelihood, ℓ_c, is further broken down into three components.

$$\ell_c(\lambda, q; D, Z, A) = \log P(Z \mid D, A, \lambda, q) + \log P(A \mid D, \lambda, q) + \log P(D \mid \lambda, q). \tag{10}$$

M-step: Maximization This step updates the model parameters by maximizing the Q function. By leveraging the expected values from the E-step, the M-step transforms the original complex optimization problem into simpler, closed-form updates. The key quantity that is used in both updates is the posterior probability that a given comparison from rater r follows the Bradley-Terry model. This quantity, denoted as γ, represents the weight of a rater's judgment based on how much it aligns with the model's current predictions. It is given by:

$$\gamma_{r,ij}^{(t-1)} = \frac{q_r^{(t-1)}\, y_{ij}^{(t-1)}}{q_r^{(t-1)}\, y_{ij}^{(t-1)} + (1 - q_r^{(t-1)})\, \frac{1}{2}}, \tag{11}$$

where $y_{ij}^{(t-1)} = \frac{\lambda_i^{(t-1)}}{\lambda_i^{(t-1)} + \lambda_j^{(t-1)}}$ is the Bradley-Terry probability that item i beats item j. This posterior probability γ represents our confidence that a comparison was meaningful rather than random, given the current parameter estimates. Higher γ values indicate more trustworthy comparisons.

The new estimate for a rater's quality, q_r, is calculated as a weighted average. The numerator sums up the ”effective number of wins” for that rater, where each win is weighted by the posterior probability (γ) that it was a meaningful, non-random judgment. This is combined with the hyperparameters from the Beta prior to regularizing the estimate. The denominator normalizes this sum by the total number of comparisons and prior parameters. This update intuitively increases a rater's quality score if their judgments frequently align with the model's predictions. The update is given by:

$$q_r^{(t)} = \frac{\sum_{i=1}^{K} \sum_{j=i+1}^{K} \left[w_{r,ij}\, \gamma_{r,ij}^{(t-1)} + w_{r,ji}\, \gamma_{r,ji}^{(t-1)} \right] + (\alpha - 1)}{n_r + \alpha + \beta - 2}, \tag{12}$$

where n_r is the total number of comparisons by rater r.

The new estimate for an item's skill, λ_i, is a ratio that balances two key quantities. The numerator is a sum of the ”effective wins” for item i across all raters, where each win is again weighted by the rater's quality (γ). This term essentially represents the total positive evidence for item i. The denominator, on the other hand, accounts for the total comparisons item i was involved in, and acts as a normalizing factor. These terms are also regularized by the Gamma prior hyperparameters. The update is given by:

$$\lambda_i^{(t)} = \frac{\sum_{r=1}^{R} \left[\sum_{j=1, j\neq i}^{K} w_{r,ij}\, \gamma_{r,ij}^{(t-1)} \right] + (a - 1)}{\sum_{j=1, j\neq i}^{K} \left[\frac{\sum_{r=1}^{R} \left[w_{r,ij} \gamma_{r,ij}^{(t-1)} + w_{r,ji} \gamma_{r,ji}^{(t-1)} \right]}{\lambda_i^{(t-1)} + \lambda_j^{(t-1)}} \right] + b}. \tag{13}$$

The derivation of the Expectation Maximization algorithm is provided in Appendix B.

4 Experimental Results

To evaluate the performance of our **B**ayesian **B**radley-Terry model with rater **Q**uality (**BBQ**), we conduct a series of experiments comparing it against two baselines: Bayesian Bradley-Terry (Bayes-BT) (Caron & Doucet, 2012) and a gradient descent-based BT model that incorporates rater quality (Crowd-BT) (Chen et al., 2013). For Crowd-BT, we use the implementation provided by Google Research (Google, 2025), which was employed both by CLIC (2025) and Ballé et al. (2025). The hyperparameters used in our experiments can be found in Section C.

We evaluate the performance of several Bradley–Terry (BT) variants across datasets, including human preference benchmarks for large language models (HUMAINE (Team, 2025), MT-Bench (Zheng et al., 2023)) and image compression (WD (Ballé et al., 2025), HiFiC (Mentzer et al., 2020), ConHa (Aczel & Wattenhofer, 2024)), as well as a newly collected inhomogeneous rater quality (IHQ) dataset from the CLIC2024 image test set (CLIC, 2025). To study the effect of rater quality, we partition the IHQ dataset into two subsets: screened and unscreened. This setup enables analysis of how low-quality, high-quality, and mixed comparisons affect model performance. Details of the IHQ dataset creation are given in Section E, and a summary of all datasets, including comparisons, raters, and items, is provided in Section D and Table 2.

Ground truth rankings are not available for real datasets. We first validated the models on simulated datasets, where all methods recover the same ordering as the number of samples increases. For crowd-sourced datasets, we approximate the ground truth by the ranking achieved on the whole dataset. We validate that this provides a reasonable approximation of ordering for real-world datasets by examining the top-1 item in each ranking. For all datasets except IHQ-unscreened, all three models recover the same top-ranked item. For the IHQ-unscreened dataset, Crowd-BT and Bayes-BT fail to identify the reference image as the highest-quality item, whereas BBQ succeeds.

A reliable aggregation method should reproduce the same ordering if the study is repeated. We approximate this stability using bootstrapping, which provides an estimate of the variability in the rankings. Details on the bootstrapping are described in Section F. Note that the WD and HiFiC studies employed active selection of comparison pairs. HiFiC used a binary search strategy, while WD selected pairs based on maximum information gain. For this reason, the bootstrapping results on these two datasets should be interpreted with caution.

We evaluate performance using two metrics. Top-1 agreement is the fraction of bootstrapped samples that identify the same best item as the full dataset. Kendall's Tau (τ) measures ordinal correlation between rankings (Kendall, 1938), with higher values indicating stable results. Top-1 agreement is most relevant when the best model matters, while Kendall's Tau assesses overall ranking stability.

4.1 Performance Across Datasets

We compare models across datasets to assess their performance on ranking accuracy and stability. Top-1 agreement and Kendall's Tau are summarized in Table 1. Datasets with more comparisons per model, such as MT-Bench, WD, and HiFiC, are generally easier. Interestingly, HUMAINE deviates from this trend, highlighting that factors beyond the total number of comparisons, such as rater consistency and diversity, can influence model performance.

Overall, BBQ demonstrates superior stability and robustness across datasets. It identifies the top-performing item most frequently, being the shared best on three datasets. It recovers the overall ranking most accurately on five out of eight datasets, ranking second on the remaining three.

The three datasets where BBQ ranks second in Kendall's Tau consist of high quality, homogeneous rater sets. On MT-Bench, all models perfectly recover the top item. The WD dataset, collected by Ballé et al. (2025), includes only five raters, likely the paper's authors, suggesting exceptionally careful evaluation. In IHQ-screened, raters were explicitly filtered for quality. In such settings where rater quality is consistently high, explicitly modeling rater reliability offers little advantage, and the benefits of BBQ over simpler models are reduced.

Experiments on the IHQ dataset, considering both screened and unscreened rater subsets, reveal a clear pattern. BBQ substantially outperforms Crowd-BT and Bayes-BT when low-quality raters are present, as in CLIC-all and CLIC-unscreened. All three models achieve their best Top-1 accuracy when restricted to the screened subset, highlighting the importance of rater selection. While Crowd-BT explicitly models rater quality, its performance drops noticeably on the full dataset, likely because crowdsourced raters provide fewer than 40 comparisons each, which increases susceptibility to noisy annotations. In contrast, BBQ maintains strong performance even without screening, with only a minor decrease in Top-1 accuracy, demonstrating robustness to low-quality raters.

Nonetheless, achieving uniformly high rater quality is challenging. Large-scale crowdsourcing introduces variability, screening procedures are costly, and subjective factors may affect even diligent annotators. In this context, BBQ provides a principled way to leverage partially reliable raters while reducing the impact of noisy contributions.

Table 1: Performance of different BT-based aggregation methods across several datasets. Top: Top-1 agreement [%], Bottom: Kendall's Tau. BBQ most frequently identifies the top-performing item across all datasets, recovers the best overall ranking for more than half of the datasets, and ranks second on the remaining datasets.

	HUMAINE	MT-Bench	WD	HiFiC	ConHa	IHQ		
						all	scr.	unscr.
Top-1 Agreement [%]								
Crowd-BT	85.60	**100.00**	99.29	98.63	66.59	85.31	98.57	33.15
Bayes-BT	97.30	**100.00**	**100.00**	**100.00**	57.30	75.07	98.90	24.32
BBQ (ours)	**97.50**	**100.00**	**100.00**	**100.00**	**77.12**	**99.32**	**99.80**	**61.92**
Kendall's Tau								
Crowd-BT	0.9385	**0.9743**	0.9279	0.9366	0.9180	0.9245	**0.9171**	0.8482
Bayes-BT	0.9443	0.9569	**0.9459**	0.9293	0.9110	0.9240	0.9116	0.8507
BBQ (ours)	**0.9463**	0.9675	0.9359	**0.9525**	**0.9265**	**0.9270**	0.9132	**0.8563**

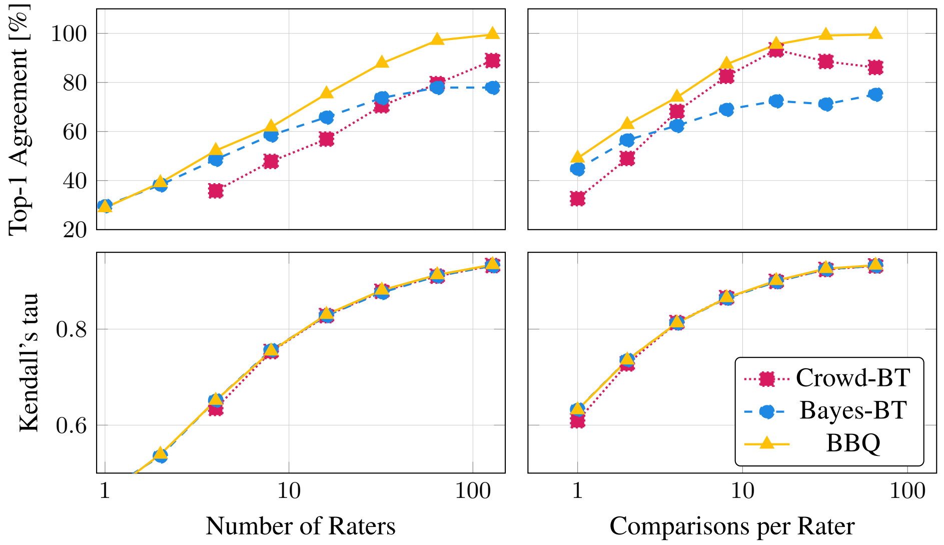


Figure 1: Scaling behavior of Bradley–Terry variants (Crowd-BT (Caron & Doucet, 2012), Bayes-BT (Chen et al., 2013), BBQ (ours)) on the IHQ-all dataset. **Left:** Performance vs. number of raters. **Right:** Performance vs. number of comparisons per rater. Both Top-1 agreement and Kendall's τ improve noticeably with more raters or comparisons. While Top-1 agreement differentiates between models, Kendall's τ remains similar across models. Crowd-BT fails to converge with very few raters, highlighting the EM algorithm's advantage. Crowd-BT and BBQ perform similarly under sparse data, but BBQ outperforms Bayes-BT as the number of raters or comparisons grows.

4.2 Scaling with Raters and Comparisons

As observed in Table 1, datasets with more comparisons per model generally yield better performance. The number of comparisons can be increased in two ways: by adding more raters, or by increasing the number of comparisons each rater performs.

Figure 1 illustrates the impact of both factors on the three BT variants. The left column shows performance as a function of the number of raters (with the number of comparisons per rater fixed at the maximum), while the right column shows performance as a function of the number of comparisons per rater (with the number of raters fixed at the maximum). A clear difference in performance can be observed in Top-1 accuracy, while Kendall's τ remains similar across models. Crowd-BT fails to converge with only one or two raters, highlighting the advantage of using the EM algorithm, which provides convergence guarantees compared to gradient descent.

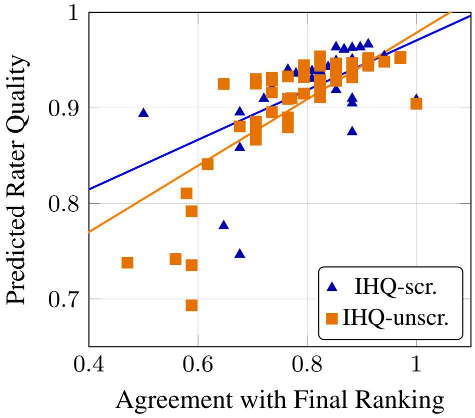


Figure 2: Scatter plot of rater agreement with the final ranking (x-axis) versus the predicted rater quality (y-axis) for the IHQ datasets. Each point corresponds to an individual rater. Triangles denote the filtered dataset, and squares denote the unfiltered dataset.

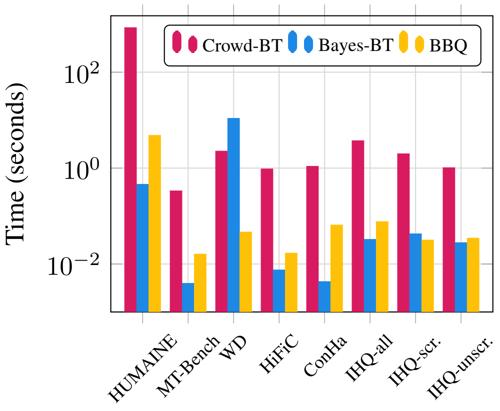


Figure 3: Average computation time in seconds (log-scale) for three models measured on a single bootstrapped sample across eight datasets. While Crowd-BT can require substantial computation time on some datasets, BBQ consistently remains fast across all datasets.

Crowd-BT and BBQ perform similarly when the number of raters or comparisons per rater is small. BBQ requires both multiple raters and multiple comparisons per rater to effectively distinguish between rater qualities. As the number of raters or comparisons per rater increases, BBQ increasingly outperforms Bayes-BT. In contrast, Bayes-BT tends to underperform overall but can surpass Crowd-BT when limited data per rater or a few number of raters are available, since Crowd-BT cannot reliably estimate rater quality in such sparse settings.

4.3 Rater Quality

Figure 2 shows the relationship between agreement with the final ranking and predicted rater quality. We observe a clear positive correlation: raters who agree more closely with the consensus ranking are assigned higher quality by the model. For the filtered dataset, the Pearson correlation is $r = 0.551$ across 50 raters. For the unfiltered dataset, the correlation is stronger, with $r = 0.724$ across 62 raters.

As expected, the unscreened dataset contains several raters with both lower agreement and lower predicted quality. Some raters in the IHQ-unscreened dataset even fall below the level of random guessing (50% agreement), systematically disagreeing with the majority. This highlights the importance of modeling rater quality when aggregating pairwise comparison data. BBQ successfully identifies such raters and assigns them lower quality scores, thereby reducing their influence on the final ranking and mitigating the noise they introduce.

4.4 Computational Efficiency

Figure 3 reports the average computation time in seconds required by each method to process a single bootstrapped sample across different datasets. These timings provide a practical perspective on the feasibility of the methods in real-world evaluation scenarios. BBQ consistently converges within a few seconds on all datasets. Crowd-BT requires more time than BBQ across the board, with particularly long runtimes on datasets with many comparisons, such as HUMAINE, where convergence takes around 15 minutes. Bayes-BT converges slowly on the WD dataset, though on other datasets it is slightly faster than BBQ. The efficiency of BBQ stems from the closed-form EM updates derived in our method, which enable rapid convergence even on large datasets.

It is also important to note that Crowd-BT was highly optimized and implemented in C (Kernighan & Ritchie, 1988), whereas BBQ was implemented using plain NumPy without specific optimizations. This suggests that BBQ could be made even faster with a compiled or vectorized implementation.

Consequently, BBQ is not only more robust and stable but also highly practical for large-scale human preference studies or applications that require repeated bootstrapping. The combination of accuracy, stability, and speed makes BBQ a compelling choice for real-world deployments.

4.5 Confidence Intervals

We evaluate how Crowd-BT, Bayes-BT, and BBQ estimate uncertainty in item rankings. In a controlled simulation, raters provide pairwise comparisons of equally skilled items (H_0). Each model estimates skills and constructs uncertainty intervals to test for significant differences (Section G). The resulting Type I error rates are shown in Figure 6. At the 99% confidence level, the expected error rate is 1%. Crowd-BT and BBQ align closely, yielding slightly lower rates, while Bayes-BT is more conservative at $\sim$0.1%. Thus, Crowd-BT and BBQ provide well-calibrated uncertainty estimates, whereas Bayes-BT underestimates false positives.

5 Limitations

Although BBQ effectively models rater quality, its advantages diminish in settings where all raters are uniformly reliable. In such homogeneous datasets, modeling variability provides little additional benefit. Ensuring consistently high-quality raters, however, often requires substantial cost and effort, which may not be feasible in large-scale studies. This tension highlights that BBQ is most useful in realistic crowdsourced settings, but less so when evaluations are carefully curated.

Another limitation concerns the type of data the framework can handle. BBQ is currently restricted to pairwise comparisons, whereas many human evaluation studies use ratings, rankings, or multi-way inputs. Extending the model to handle these forms of feedback would broaden its applicability.

From a methodological perspective, our experiments rely on a limited number of bootstrapped comparisons. While this provides a practical measure of stability, larger-scale studies would be needed to further validate robustness. Additionally, the model assumes independence across comparisons and does not account for contextual or order effects, which may influence human judgments in practice.

Finally, although BBQ scales well computationally, extremely large numbers of items or raters could still pose challenges without optimized or compiled implementations.

6 Conclusion

We introduced BBQ, a Bayesian Bradley-Terry model that jointly estimates item quality and rater reliability. Our core contribution is the derivation of an Expectation-Maximization (EM) algorithm that simultaneously estimates item skills and individual rater quality, effectively down-weighting or removing the influence of unreliable participants. By explicitly modeling rater quality, the method produces more stable and accurate rankings. The EM algorithm ensures rapid convergence and monotonic likelihood improvement, addressing limitations of gradient-based approaches.

Across diverse datasets, BBQ consistently achieved high Top-1 agreement and superior Kendall's Tau, demonstrating both robustness and reliability. The model excels in scenarios with noisy or crowdsourced raters, maintaining accuracy even when raters contribute few comparisons. BBQ is particularly effective in large-scale settings where many raters are non-experts or vary widely in attentiveness and diligence. Our results highlight that incorporating rater quality is especially crucial when evaluation quality is heterogeneous or partially unreliable. Additionally, the Bayesian framework provides principled uncertainty estimates for item scores, enabling interpretable comparisons across studies. We further demonstrate that the model's error bars are well-calibrated and can be used to assess whether differences between items are statistically significant. Predicted rater quality aligns with agreement to final rankings, validating the model's ability to identify reliable evaluators.

Overall, BBQ advances human evaluation methodology by offering a practical, interpretable, and generalizable approach to aggregate noisy pairwise comparisons. This work contributes a significant step toward more cost-effective, interpretable, and reproducible human studies for evaluating AI-generated content.

Reproducibility Statement

The code used in our experiments is provided in the supplementary material, along with a detailed README that explains how to set up the environment and run all bootstrapping experiments. The supplementary material also includes all datasets used in our study, including the publicly available datasets on which we ran our experiments. In addition, our newly collected dataset is included, making it fully accessible for replication and further research. For the camera-ready version, we will make both the code and all datasets publicly available. We also provide comprehensive descriptions of the proposed model (Section 3), experimental setup (Section 4), hyperparameters (Section C), and data collection procedures (Section E).

References

Till Aczel and Roger Wattenhofer. Conditional hallucinations for image compression. *arXiv preprint arXiv:2410.19493*, 2024.

Eldridge S Adams. Bayesian analysis of linear dominance hierarchies. *Animal Behaviour*, 69(5): 1191–1201, 2005.

Alan Agresti. *Analysis of ordinal categorical data*. John Wiley & Sons, 2010.

Jona Ballé, Luca Versari, Emilien Dupont, Hyunjik Kim, and Matthias Bauer. Good, cheap, and fast: Overfitted image compression with wasserstein distortion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23259–23268, 2025.

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

Francois Caron and Arnaud Doucet. Efficient bayesian inference for generalized bradley–terry models. *Journal of Computational and Graphical Statistics*, 21(1):174–196, 2012.

Xi Chen, Paul N Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 193–202, 2013.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*, 2024.

CLIC. Clic 2024: Challenge on learned image compression. `https://archive.compression.cc/2024/`, 2025. Accessed: 2025-09-12.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1): 1–22, 1977.

Persi Diaconis. Group representations in probability and statistics. *Lecture notes-monograph series*, 11:i–192, 1988.

Lester R Ford Jr. Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly*, 64(8P2):28–33, 1957.

Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009.

Google. Google, 2025. URL `https://github.com/google-research/google-research`. Accessed: 2025-09-12.

John Guiver and Edward Snelson. Bayesian inference for plackett-luce ranking models. In *proceedings of the 26th annual international conference on machine learning*, pp. 377–384, 2009.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Tzu-Kuo Huang, Ruby C Weng, Chih-Jen Lin, and Greg Ridgeway. Generalized bradley-terry models and multi-class probability estimates. *Journal of Machine Learning Research*, 7(1), 2006.

David R Hunter. Mm algorithms for generalized bradley-terry models. *The annals of statistics*, 32 (1):384–406, 2004.

Frederick Jelinek. *Statistical methods for speech recognition*. MIT press, 1998.

Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language*. Prentice Hall, 1988.

Kenneth Lange, David R Hunter, and Ilsoon Yang. Optimization transfer using surrogate objective functions. *Journal of computational and graphical statistics*, 9(1):1–20, 2000.

Mabyduck Ltd. Mabyduck, 2025. URL `https://www.mabyduck.com/`. Accessed: 2025-09-12.

R Duncan Luce et al. *Individual choice behavior*, volume 4. Wiley New York, 1959.

Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in neural information processing systems*, 33:11913–11924, 2020.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

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

Pejaver V Rao and Lawrence L Kupper. Ties in paired-comparison experiments: A generalization of the bradley-terry model. *Journal of the American Statistical Association*, 62(317):194–204, 1967.

Prolific AI Team. Humaine: Human-ai interaction evaluation dataset, 2025. URL `https://huggingface.co/datasets/ProlificAI/humaine-evaluation-dataset`.

Kexin Wang, Yunlong Zhao, Qianqian Dong, Tom Ko, and Mingxuan Wang. Mospc: Mos prediction based on pairwise comparison. *arXiv preprint arXiv:2306.10493*, 2023.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The thrity-seventh asilomar conference on signals, systems & computers, 2003*, volume 2, pp. 1398–1402. Ieee, 2003.

Emin Zerman, Vedad Hulusic, Giuseppe Valenzise, Rafał K Mantiuk, and Frédéric Dufaux. The relation between mos and pairwise comparisons and the importance of cross-content comparisons. *Electronic Imaging*, 30:1–6, 2018.

Ernst Zermelo. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436–460, 1929.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

A Usage of LLMs

During the preparation of this paper, we utilized large language models (LLMs) as supportive tools. ChatGPT, Claude, Gemini, and Grammarly assisted with spellchecking, refining phrasing, and condensing text to enhance clarity and readability. Additionally, ChatGPT, Claude, and Cursor were used for code analysis, completion, and the generation of visualizations to aid our development workflow. These tools were employed solely as auxiliary aids, while all core research ideas, experimental design, and interpretation of results were developed independently by the authors.

B Bayesian BT with rater quality derivation

We consider R raters comparing K items. Each rater r has quality q_r, meaning that with probability q_r they follow the Bradley–Terry model, and with probability $1 - q_r$ they choose randomly. The following is a standard EM (Dempster et al., 1977) derivation that incorporates these rater-specific reliabilities into the Bayesian estimation framework.

B.1 Model definition

Notation:

- $\lambda = (\lambda_1, \ldots, \lambda_K)$: skill parameters for the K items.
- $q = (q_1, \ldots, q_R)$: quality parameters for the R raters.
- $Y_i \sim \mathcal{E}(\lambda_i)$: latent arrival time associated with item i.
- $A_{r,ij}^{(c)} \sim \text{Bernoulli}(q_r)$: indicator that the c-th comparison of pair (i, j) by rater r follows the Bradley-Terry model.
- $n_{r,ij}$: total number of comparisons of (i, j) by rater r.
- $n_{ij} = \sum_{r=1}^{R} n_{r,ij}$: total number of comparisons of (i, j).
- $w_{r,ij}$: number of times rater r ranked i above j.
- $w_{ij} = \sum_{r=1}^{R} w_{r,ij}$: total number of times i was ranked above j.
- $m_{r,ij} = \sum_{c=1}^{n_{r,ij}} A_{r,ij}^{(c)}$: number of Bradley-Terry-model comparisons of (i, j) by rater r.
- $m_{ij} = \sum_{r=1}^{R} m_{r,ij}$: total number of Bradley-Terry-model comparisons of (i, j).
- $v_{r,ij} = \sum_{c=1}^{w_{r,ij}} A_{r,ij}^{(c)}$: number of Bradley-Terry-model wins of i over j by rater r.
- $v_{ij} = \sum_{r=1}^{R} v_{r,ij}$: total number of Bradley-Terry-model wins of i over j.
- $Z_{r,ij} = \sum_{c=1}^{n_{r,ij}} A_{r,ij}^{(c)} \min(Y_i^{(c)}, Y_j^{(c)})$: sum of minimal arrival times, with

$$Z_{r,ij} \;\middle|\; m_{r,ij} \;\sim\; \Gamma(m_{r,ij}, \lambda_i + \lambda_j)\,.$$

Bradley-Terry probability:

$$P(i \,\text{beats}\, j) = \frac{\lambda_i}{\lambda_i + \lambda_j}.$$

Mixture with rater quality:

$$P(r \text{ ranks } i \text{ above } j) = q_r \frac{\lambda_i}{\lambda_i + \lambda_j} + (1 - q_r)\frac{1}{2}.$$

Latent exponential view:

$$Y_i \sim \mathcal{E}(\lambda_i), \quad Y_j \sim \mathcal{E}(\lambda_j), \quad P(i \,\text{beats}\, j) = P(Y_i < Y_j).$$

B.2 Complete-Data Log-Likelihood

We need to compute:

$$
\begin{aligned}
\ell_c(\lambda, q; D, Z, A) &= \log P(D, Z, A \mid \lambda, q) \\
&= \log P(Z \mid D, A, \lambda, q) + \log P(A \mid D, \lambda, q) + \log P(D \mid \lambda, q)
\end{aligned}
$$

Log-Likelihood of $P(Z \mid A, \lambda)$:

$$
\begin{aligned}
\log P(Z \mid D, A, \lambda, q) = \sum_{i=1}^{K} \sum_{j=i+1}^{K} \Big[& m_{ij} \log(\lambda_i + \lambda_j) - (\lambda_i + \lambda_j) z_{ij} \\
& + (m_{ij} - 1) \log z_{ij} - \log \Gamma(m_{ij}) \Big]
\end{aligned}
$$

Log-Likelihood of $P(A \mid q)$:

$$
\log P(A \mid D, \lambda, q) = \sum_{r=1}^{R} \sum_{i=1}^{K} \sum_{j=i+1}^{K} \Big[m_{r,ij} \log q_r + (n_{r,ij} - m_{r,ij}) \log(1 - q_r) \Big]
$$

Log-Likelihood of $P(D \mid \lambda, A)$:

$$
\begin{aligned}
\log P(D \mid \lambda, q) &= \sum_{r=1}^{R} \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} \left[v_{r,ij} \log \frac{\lambda_i}{\lambda_i + \lambda_j} + (w_{r,ij} - v_{r,ij}) \log \frac{1}{2} \right] \\
&= \sum_{r=1}^{R} \sum_{i=1}^{K} [v_{ir} \log \lambda_i] \\
&\quad - \sum_{r=1}^{R} \sum_{i=1}^{K} \sum_{j=i+1}^{K} \left[(v_{r,ij} + v_{r,ji}) \log (\lambda_i + \lambda_j) + (w_{r,ij} + w_{r,ji} - v_{r,ij} - v_{r,ji}) \log 2 \right] \\
&= \sum_{i=1}^{K} [v_i \log \lambda_i] \\
&\quad - \sum_{i=1}^{K} \sum_{j=i+1}^{K} \left[m_{ij} \log (\lambda_i + \lambda_j) + (n_{ij} - m_{ij}) \log 2 \right]
\end{aligned}
$$

Complete-Data Log-Likelihood:

$$\begin{aligned}
\ell_c(\lambda, q; D, Z, A) &= \log P(Z \mid D, A, \lambda, q) + \log P(A \mid D, \lambda, q) + \log P(D \mid \lambda, q) \\
&= \sum_{i=1}^{K} \sum_{j=i+1}^{K} \Big[m_{ij} \log(\lambda_i + \lambda_j) - (\lambda_i + \lambda_j) z_{ij} + (m_{ij} - 1) \log z_{ij} - \log \Gamma(m_{ij}) \Big] \\
&\quad + \sum_{r=1}^{R} \sum_{i=1}^{K} \sum_{j=i+1}^{K} \Big[m_{r,ij} \log q_r + (n_{r,ij} - m_{r,ij}) \log(1 - q_r) \Big] \\
&\quad + \sum_{i=1}^{K} [v_i \log \lambda_i] \\
&\quad - \sum_{i=1}^{K} \sum_{j=i+1}^{K} [m_{ij} \log(\lambda_i + \lambda_j) + (n_{ij} - m_{ij}) \log 2] \\
&= \sum_{i=1}^{K} \sum_{j=i+1}^{K} \Big[(n_{ij} - m_{ij}) \log 2 - (\lambda_i + \lambda_j) z_{ij} + (m_{ij} - 1) \log z_{ij} - \log \Gamma(m_{ij}) \Big] \\
&\quad + \sum_{r=1}^{R} \sum_{i=1}^{K} \sum_{j=i+1}^{K} \Big[m_{r,ij} \log q_r + (n_{r,ij} - m_{r,ij}) \log(1 - q_r) \Big] \\
&\quad + \sum_{i=1}^{K} [v_i \log \lambda_i]
\end{aligned}$$

B.3 Expectation step

We introduce conjugate priors: Gamma distribution $\lambda_i \sim \Gamma(a, b)$ for each item i, and Beta $q_r \sim B(\alpha, \beta)$ for each rater r.

The Q-function is the expectation of the complete-data log-posterior:

$$Q(\lambda, q \mid \lambda^*, q^*) = \mathbb{E}_{A,Z \mid D, \lambda^*, q^*} \big[\ell_c(\lambda, q; D, Z, A) + \log P(\lambda) + \log P(q) \big].$$

Posterior probability of a Bradley-Terry-consistent annotation:

$$\begin{aligned}
P\big(A_{r,ij}^{(k)} = 1 \mid i \succ j\big) &= \frac{P(i \succ j \mid A = 1)\, P(A = 1)}{P(i \succ j \mid A = 1)\, P(A = 1) \;+\; P(i \succ j \mid A = 0)\, P(A = 0)} \\
&= \frac{\big(\frac{\lambda_i^*}{\lambda_i^* + \lambda_j^*}\big)\, q_r^*}{\big(\frac{\lambda_i^*}{\lambda_i^* + \lambda_j^*}\big)\, q_r^* \;+\; \frac{1}{2}\, (1 - q_r^*)} \;=\; \gamma_{r,ij}^*.
\end{aligned}$$

Expected sufficient statistics:

$$\begin{aligned}
\mathbb{E}[m_{r,ij}] &= w_{r,ij} \gamma_{r,ij}^* + w_{r,ji} \gamma_{r,ji}^* \\
\mathbb{E}[m_{ij}] &= \sum_{r=1}^{R} (w_{r,ij} \gamma_{r,ij}^* + w_{r,ji} \gamma_{r,ji}^*) \\
\mathbb{E}[v_{r,ij}] &= w_{r,ij}\, \gamma_{r,ij}^* \\
\mathbb{E}[v_{ir}] &= \sum_{j, j \neq i}^{K} w_{r,ij}\, \gamma_{r,ij}^* \\
\mathbb{E}[v_i] &= \sum_{r=1}^{R} \sum_{j, j \neq i}^{K} w_{r,ij}\, \gamma_{r,ij}^* \\
\mathbb{E}[z_{ij}] &= \frac{\sum_{r=1}^{R} (w_{r,ij} \gamma_{r,ij}^* + w_{r,ji} \gamma_{r,ji}^*)}{\lambda_i^* + \lambda_j^*}.
\end{aligned}$$

Expected complete-data log-posterior:

$$
\begin{aligned}
\mathbb{E}[\ell_c] = \sum_{i=1}^{K}\sum_{j=i+1}^{K} & \left[-(\lambda_i+\lambda_j)\frac{\sum_{r=1}^{R}(w_{r,ij}\gamma^*_{r,ij}+w_{r,ji}\gamma^*_{r,ji})}{\lambda^*_i+\lambda^*_j} \right] \\
& + \sum_{r=1}^{R}\sum_{i=1}^{K}\sum_{j=i+1}^{K} \Big[(w_{r,ij}\gamma^*_{r,ij}+w_{r,ji}\gamma^*_{r,ji})\log q_r \\
& \qquad\qquad + (n_{r,ij}-(w_{r,ij}\gamma^*_{r,ij}+w_{r,ji}\gamma^*_{r,ji}))\log(1-q_r) \Big] \\
& + \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K} w_{r,ij}\gamma^*_{r,ij}\log\lambda_i \\
& + \text{const.}
\end{aligned}
$$

Priors contribute:

$$
\begin{aligned}
\mathbb{E}[\log P(\lambda)] &= \sum_{i=1}^{K}\left[(a-1)\log\lambda_i - b\lambda_i\right], \\
\mathbb{E}[\log P(q)] &= \sum_{r=1}^{R}\left[(\alpha-1)\log q_r + (\beta-1)\log(1-q_r)\right].
\end{aligned}
$$

Final Q-function:

$$
\begin{aligned}
Q(\lambda, q \mid \lambda^*, q^*) = \sum_{i=1}^{K}\sum_{j=i+1}^{K} & \left[-(\lambda_i+\lambda_j)\frac{\sum_{r=1}^{R}(w_{r,ij}\gamma^*_{r,ij}+w_{r,ji}\gamma^*_{r,ji})}{\lambda^*_i+\lambda^*_j} \right] \\
& + \sum_{r=1}^{R}\sum_{i=1}^{K}\sum_{j=i+1}^{K} \Big[(w_{r,ij}\gamma^*_{r,ij}+w_{r,ji}\gamma^*_{r,ji})\log q_r \\
& \qquad\qquad + (n_{r,ij}-(w_{r,ij}\gamma^*_{r,ij}+w_{r,ji}\gamma^*_{r,ji}))\log(1-q_r) \Big] \\
& + \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K} w_{r,ij}\gamma^*_{r,ij}\log\lambda_i \\
& + \sum_{i=1}^{K}\left[(a-1)\log\lambda_i - b\lambda_i\right] \\
& + \sum_{r=1}^{R}\left[(\alpha-1)\log q_r + (\beta-1)\log(1-q_r)\right] \\
& + \text{const.}
\end{aligned}
$$

where

$$
\gamma^*_{r,ij} = \frac{q^*_r\,\dfrac{\lambda^*_i}{\lambda^*_i+\lambda^*_j}}{q^*_r\,\dfrac{\lambda^*_i}{\lambda^*_i+\lambda^*_j} + (1-q^*_r)\,\frac{1}{2}}.
$$

B.4 M-STEP

The M-step maximizes the Q-function w.r.t. the parameters (λ, q), holding the expectations computed in the E-step fixed.

Update for q_r The update for each rater quality q_r is obtained by maximizing Q with respect to q_r (including the Beta prior).

$$
\begin{aligned}
Q(q \mid \lambda^*, q^*) = \sum_{r=1}^{R} \sum_{i=1}^{K} \sum_{j=i+1}^{K} \Big[& (w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji}) \log q_r \\
& + (n_{r,ij} - (w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji})) \log(1 - q_r) \Big] \\
& + (\alpha - 1) \log q_r + (\beta - 1) \log(1 - q_r) + \text{const.} \\
\frac{\partial Q}{\partial q_r} = \sum_{i=1}^{K} \sum_{j=i+1}^{K} \Big[& \frac{w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji}}{q_r} - \frac{n_{r,ij} - (w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji})}{1 - q_r} \Big] \\
& + \frac{\alpha - 1}{q_r} - \frac{\beta - 1}{1 - q_r} \\
0 \overset{!}{=} & \frac{\sum_{i=1}^{K} \sum_{j=i+1}^{K} (w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji}) + (\alpha - 1)}{q_r} \\
& - \frac{\sum_{i=1}^{K} \sum_{j=i+1}^{K} (n_{r,ij} - (w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji})) + (\beta - 1)}{1 - q_r} \\
\implies \quad & (1 - q_r) \left(\sum_{i=1}^{K} \sum_{j=i+1}^{K} (w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji}) + (\alpha - 1) \right) \\
\implies \quad & q_r \left(\sum_{i=1}^{K} \sum_{j=i+1}^{K} (n_{r,ij} - (w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji})) + (\beta - 1) \right) \\
\implies \quad & q_r = \frac{\sum_{i=1}^{K} \sum_{j=i+1}^{K} (w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji}) + (\alpha - 1)}{\sum_{i=1}^{K} \sum_{j=i+1}^{K} (w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji} + n_{r,ij} - (w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji})) + (\beta + \alpha - 2)} \\
\implies \quad & q_r^{(t)} = \frac{\sum_{i=1}^{K} \sum_{j=i+1}^{K} (w_{r,ij}\gamma^{(t-1)}_{r,ij} + w_{r,ji}\gamma^{(t-1)}_{r,ji}) + (\alpha - 1)}{n_r + \beta + \alpha - 2}
\end{aligned}
$$

where the last line gives the explicit update at iteration t.

Update for λ_i The update for each item skill λ_i is obtained by maximizing Q with respect to λ_i (including the Gamma prior).

$$
\begin{aligned}
Q(\lambda \mid \lambda^*, q^*) = \sum_{i=1}^{K} \sum_{j=i+1}^{K} & -(\lambda_i + \lambda_j) \frac{\sum_{r=1}^{R} (w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji})}{\lambda^*_i + \lambda^*_j} \\
& + \sum_{r=1}^{R} \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} w_{r,ij}\gamma^*_{r,ij} \log \lambda_i \\
& + \sum_{i=1}^{K} [(a - 1) \log \lambda_i - b\lambda_i] + \text{const.}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial Q}{\partial \lambda_i} &= -\sum_{j\neq i}^{K} \frac{\sum_{r=1}^{R}(w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji})}{\lambda^*_i + \lambda^*_j} + \sum_{r=1}^{R} \frac{\sum_{j\neq i}^{K} w_{r,ij}\,\gamma^*_{r,ij}}{\lambda_i} + \frac{a-1}{\lambda_i} - b \\
0 &\overset{!}{=} \frac{\sum_{r=1}^{R}\sum_{j\neq i}^{K} w_{r,ij}\,\gamma^*_{r,ij} + (a-1)}{\lambda_i} - \sum_{j\neq i}^{K} \frac{\sum_{r=1}^{R}(w_{r,ij}\gamma^*_{r,ij} + w_{r,ji}\gamma^*_{r,ji})}{\lambda^*_i + \lambda^*_j} - b \\
\implies \quad \lambda_i^{(t)} &= \frac{\sum_{r=1}^{R}\sum_{j\neq i}^{K} w_{r,ij}\,\gamma^{(t-1)}_{r,ij} + (a-1)}{\sum_{j\neq i}^{K} \dfrac{\sum_{r=1}^{R}(w_{r,ij}\,\gamma^{(t-1)}_{r,ij} + w_{r,ji}\,\gamma^{(t-1)}_{r,ji})}{\lambda_i^{(t-1)} + \lambda_j^{(t-1)}} + b} \\
\gamma^{(t-1)}_{r,ij} &= \frac{q_r^{(t-1)}\,\dfrac{\lambda_i^{(t-1)}}{\lambda_i^{(t-1)} + \lambda_j^{(t-1)}}}{q_r^{(t-1)}\,\dfrac{\lambda_i^{(t-1)}}{\lambda_i^{(t-1)} + \lambda_j^{(t-1)}} + (1 - q_r^{(t-1)})\,\frac{1}{2}}
\end{aligned}
$$

where the last line gives the explicit update at iteration t.

C HYPERPARAMETERS

For Crowd-BT, we use the implementation of Google (2025) with the default hyperparameters.

The only hyperparameters in the Bayes-BT and BBQ models are the prior distribution parameters and the stopping thresholds. For both Bayes-BT and BBQ, we consider the model converged when no ELO score changes by more than 1 between two iterations. The ELO score can be calculated from the skill parameter λ as:

$$
\text{ELO} = \log(\text{skill}) \cdot \text{ELO\_SCALE\_FACTOR}. \tag{14}
$$

where we set the ELO_SCALE_FACTOR to 400.

We chose a gamma prior with shape 5 and rate 0.1 for the skill parameters in both Bayes-BT and BBQ. For BBQ, the beta prior on the rater quality has $\alpha = 10$ and $\beta = 2$.

D DATASETS

We evaluate our models on a diverse set of human preference datasets covering both language and image domains. Table 2 provides an overview.

For natural language, we use the HUMAINE dataset (Team, 2025), a large-scale benchmark with over 100k comparisons, and MT-Bench (Zheng et al., 2023), which provides model comparison judgments from crowd workers on multi-turn dialogues.

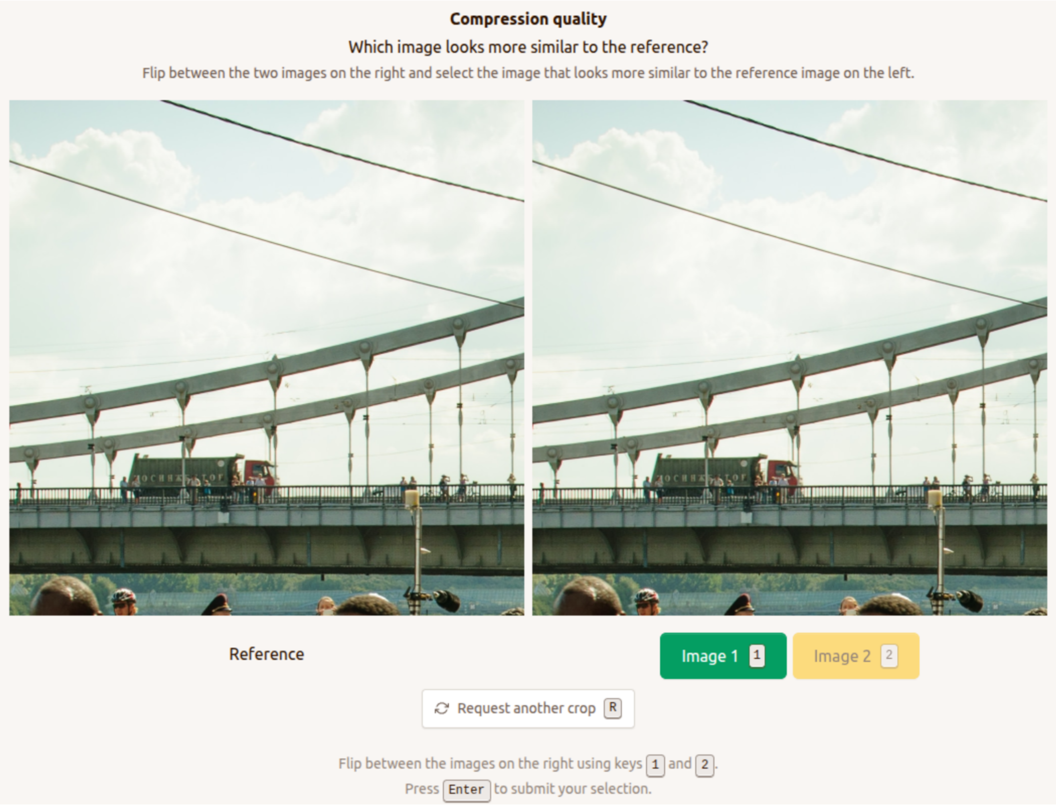
For image compression, we consider three datasets: WD (Ballé et al., 2025), a dense expert-labeled dataset with thousands of comparisons per rater; HiFiC (Mentzer et al., 2020), which evaluates learned image codecs; and ConHa (Aczel & Wattenhofer, 2024), which focuses on conditional generative models. These datasets vary substantially in scale, number of raters, and rater expertise, providing a broad testbed for robustness.

Finally, we introduce the inhomogeneous rater quality (**IHQ**) dataset, obtained from a user study on the CLIC2024 (CLIC, 2025) data conducted via the *Mabyduck* platform. It contains two-alternative-forced-choice (2AFC) judgments across 28 generative image compression models. To study the impact of rater quality, we provide two subsets: (i) **screened**, where raters passed pre-screening checks for attention and display quality, and (ii) **unscreened**, where all raters are included. The screened subset represents higher-quality raters, whereas the unscreened subset better reflects the noisy conditions typical of large-scale human evaluations.

This collection of datasets allows us to study both large-scale, relatively clean benchmarks and smaller, noisier settings where modeling rater reliability is critical.

Table 2: Summary of datasets used in our experiments. The IHQ dataset was collected by us on the Mabyduck platform and includes both 2AFC and 3AFC settings. For this dataset, we provide screened and unscreened subsets: screened subsets include only raters who passed pre-screening tests for attention and display quality, ensuring higher reliability, while unscreened subsets include all raters.

dataset	# comparisons	# raters	# models	comp/rater	comp/model
HUMAINE (Team, 2025)	105,220	1,977	27	53.2	3897.0
MT-Bench (Zheng et al., 2023)	3,355	65	6	51.6	559.2
WD (Ballé et al., 2025)	16,659	5	30	3331.8	555.3
HiFiC (Mentzer et al., 2020)	5,220	20	9	261.0	580.0
ConHa (Aczel & Wattenhofer, 2024)	1,531	40	8	38.3	191.4
IHQ-all	4,074	112	28	36.4	145.5
IHQ-screened	2,012	50	28	40.2	71.9
IHQ-unscreened	2,062	62	28	33.3	73.6



Compression quality

Which image looks more similar to the reference?

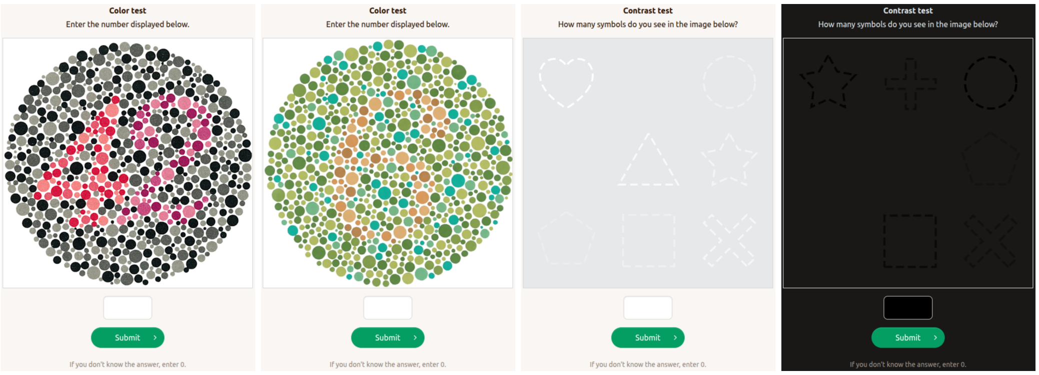
Flip between the two images on the right and select the image that looks more similar to the reference image on the left.

Flip between the images on the right using keys 1 and 2.
Press Enter to submit your selection.

Figure 4: Screenshot of the Mabyduck user study platform used for collecting pairwise comparisons. A reference image is shown on the left, and the rater selects between two compressed images on the right.

E USER STUDY PLATFORM

All pairwise comparisons on the CLIC2024 (CLIC, 2025) dataset were collected using the Mabyduck platform (Ltd., 2025). A screenshot of the platform can be seen in Figure 4. The task asks: “Which image looks more similar to the reference image?” Ties are not allowed, and all pairs were selected uniformly at random.



Color test

Enter the number displayed below.

If you don't know the answer, enter 0.

Color test

Enter the number displayed below.

If you don't know the answer, enter 0.

Contrast test

How many symbols do you see in the image below?

If you don't know the answer, enter 0.

Contrast test

How many symbols do you see in the image below?

If you don't know the answer, enter 0.

Figure 5: Four example images from the pre-screening process for raters. The first two are color blindness tests, where raters must identify the number displayed in each pattern. The last two are shape detection tests designed to evaluate sensitivity to low-contrast objects: one light gray shape on a white background and one dark gray shape on a black background.

To study rater quality, we split the IHQ dataset into screened and unscreened subsets. Screened raters passed pre-tests for color vision, display contrast, and sensitivity to subtle differences, and completed training comparisons choosing the higher-quality image to reinforce evaluation criteria.

Figure 5 shows four example images from the pre-screening process. The first two are standard color blindness tests, where raters must correctly identify the number shown in each pattern. The final two images are shape detection tests designed to evaluate the raters' sensitivity to low-contrast objects: one features a light gray shape on a white background, and the other a dark gray shape on a black background. These pre-screening tests help ensure that only raters with adequate visual capabilities contribute to the screened subset.

F Bootstrapping Details

Since both BBQ and Crowd-BT explicitly model rater quality, we perform bootstrapping over raters rather than individual comparisons. This approach preserves each rater's comparison distribution and ensures a fair assessment of stability. For all datasets, we perform 10,000 bootstrap resamples of raters, except for the HUMAINE dataset, where a single bootstrap iteration for Crowd-BT takes approximately 15 minutes, as discussed in Section 4.4. On the HUMAINE dataset, we perform 1,000 resamples to reduce computation time.

G Uncertainty Estimation Details

In addition to ranking items, estimating uncertainty is important to assess whether observed differences are statistically significant.

The Bayesian Bradley-Terry (Bayes-BT) and Bayesian Bradley-Terry with Quality (BBQ) models quantify uncertainty via the posterior distribution over item skills. Each skill has a Gamma prior, which is updated using the observed pairwise comparisons. Credible intervals derived from the posterior are then converted to the Elo scale to facilitate comparison across items.

The classical Crowd-BT model, being non-Bayesian, estimates uncertainty using a frequentist approximation. Specifically, a second-order Taylor expansion around the maximum likelihood estimate is employed. The Hessian of the log-likelihood is inverted to obtain the covariance matrix, whose diagonal entries correspond to the variances of individual items. These variances are converted to 99% confidence intervals via:

$$p_{99} = \sqrt{\text{diag(covariance)}} \times k_{\text{Erfc0\_01}} \times \sqrt{2} \approx \sqrt{\text{diag(covariance)}} \times 3.29,$$

where the constants scale the standard deviation to match the 99% confidence level. Narrower likelihood peaks yield smaller intervals, reflecting higher certainty, while flatter peaks produce larger intervals, indicating greater uncertainty.

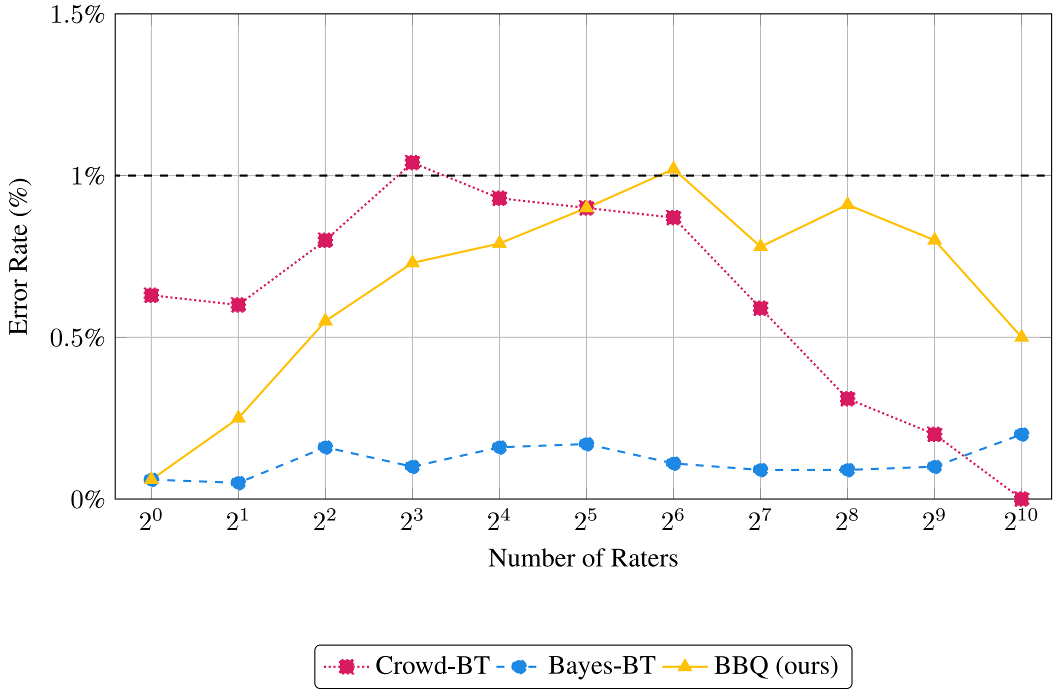


Figure 6: Type I error rate as a function of the number of raters. The dashed line indicates the expected error rate of 1%.

To evaluate how well these models estimate uncertainty, we conduct a controlled simulation experiment. We generate trials with two items of equal skill (50–50 win probability) and simulate pairwise comparison data using coin flips. For each trial, multiple raters (users) are simulated, and models are applied to estimate Elo scores and their associated uncertainty.

Formally, the null hypothesis H_0 states that the two items are equally strong. For each trial, we compute the 99% confidence (or credible) interval for each item's skill estimate. A model is said to incorrectly reject H_0 if the confidence intervals do not overlap, indicating a statistically significant difference between items when none exists. The primary metric of interest is the frequency with which each model incorrectly concludes that the items are different, i.e., the observed Type I error rate.

We perform 10,000 trials, using 50 comparisons per rater, while varying the number of raters to assess how uncertainty estimates scale with the amount of data. This setup allows us to estimate the empirical Type I error rate for each model and compare it against the theoretical expectation of 1% at the 99% confidence level. As shown in Figure 6, Crowd-BT and BBQ exhibit Type I error rates close to the expected 1%, indicating that their uncertainty estimates are well-calibrated. In contrast, Bayes-BT is overly conservative, consistently producing lower error rates than expected, which suggests its credible intervals are not as well-calibrated for significance testing.

H Visual Examples

Visual examples of the binary comparisons used in our evaluation are shown in Figures 7 and 8. For each pair, we also display the predicted win probability (in percentage) of the method that generated the corresponding image. These examples are randomly sampled from the IHQ all dataset and are not cherry-picked. Note that the predicted probability depends solely on the ELO score of the generative method and is therefore independent of the specific image content.

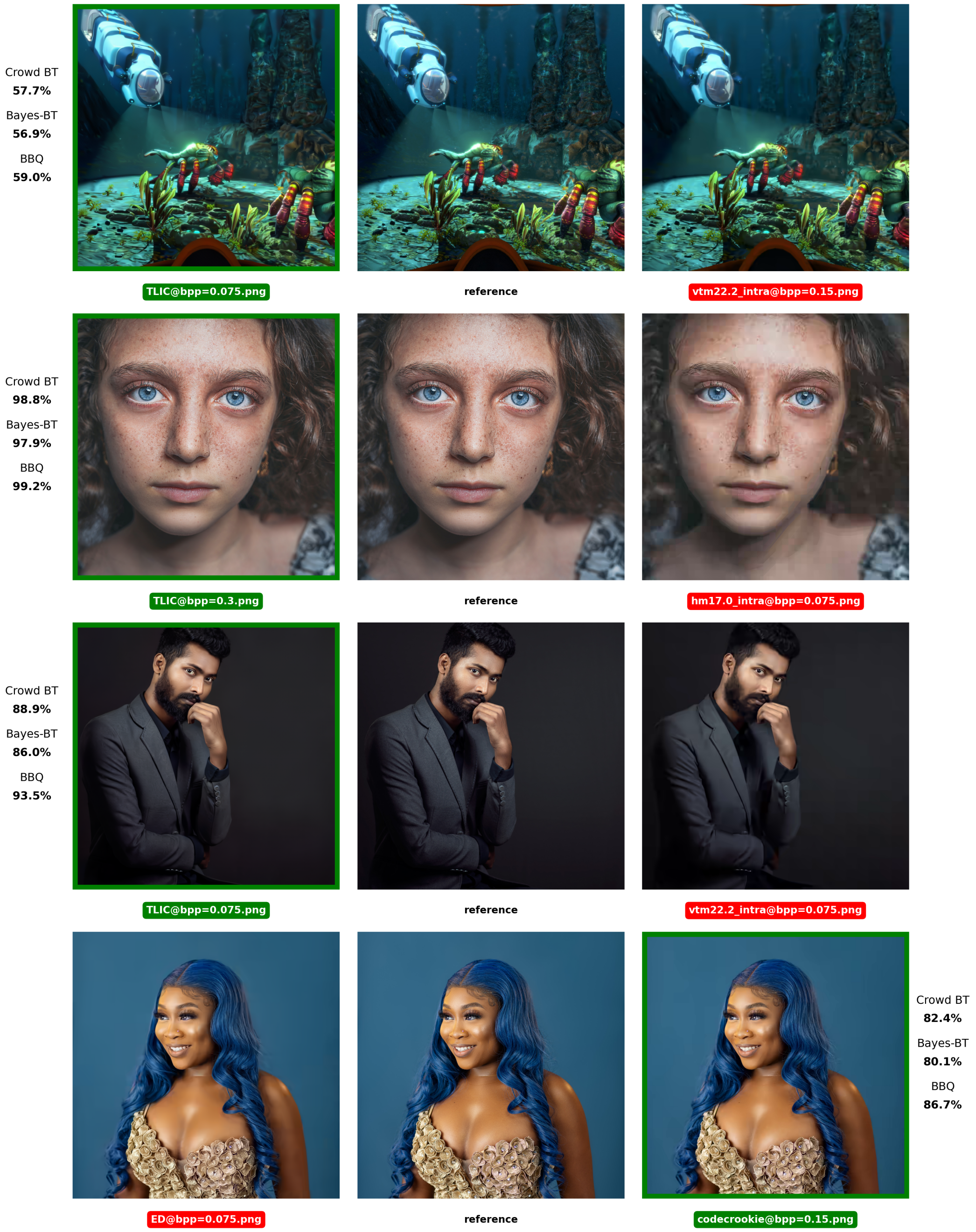


Figure 7: Example result from our binary comparison evaluation. The human-selected image is highlighted in green, and the non-selected image is shown in red. A table beside the selected image reports the predicted win probability of the corresponding generative method over its competitor. Note that these probabilities depend only on the ELO scores of the methods, not on the specific image content.

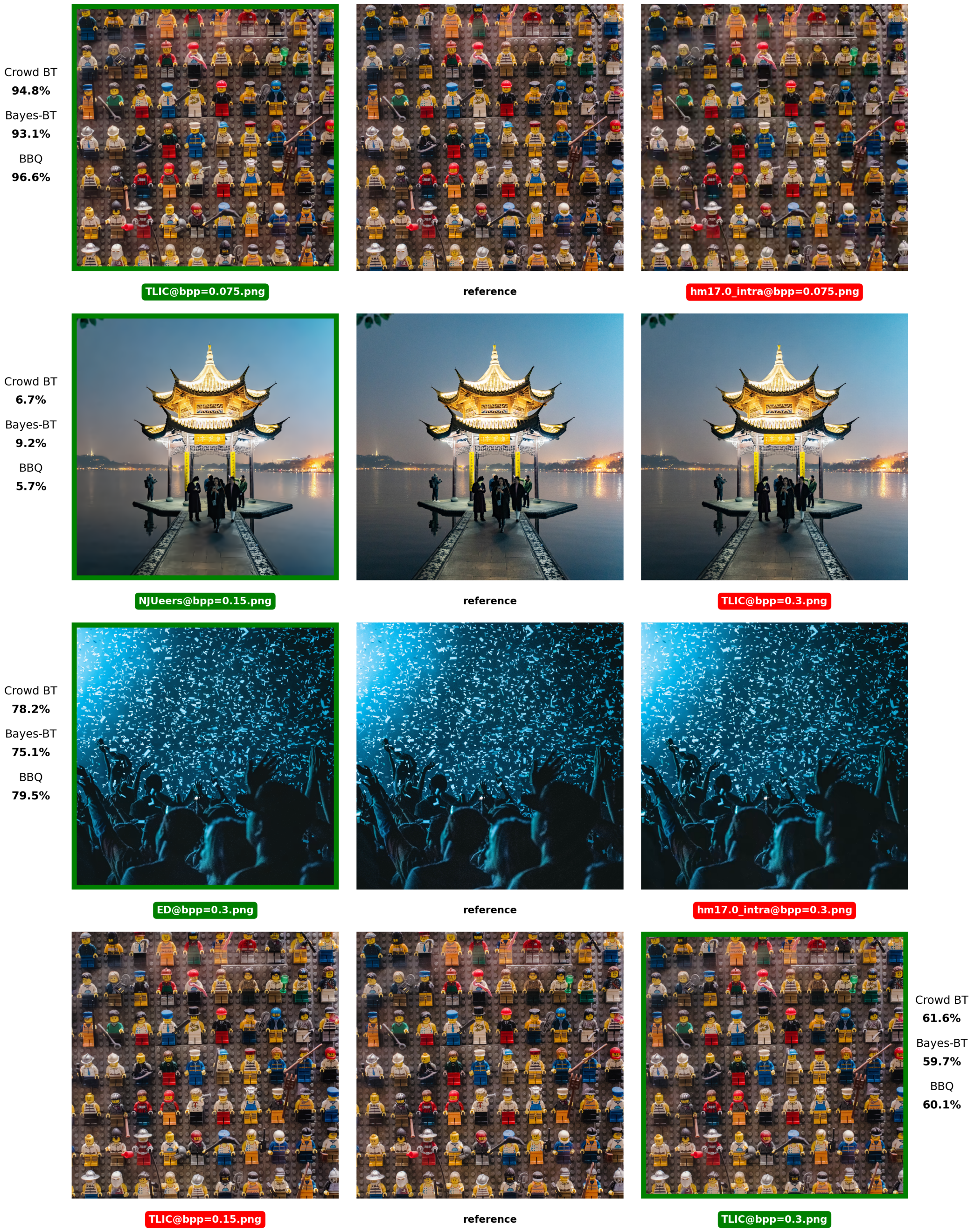


Figure 8: Example result from our binary comparison evaluation. The human-selected image is highlighted in green, and the non-selected image is shown in red. A table beside the selected image reports the predicted win probability of the corresponding generative method over its competitor. Note that these probabilities depend only on the ELO scores of the methods, not on the specific image content.