

# UNICBE: AN UNIFORMITY-DRIVEN COMPARING-BASED EVALUATION FRAMEWORK WITH UNIFIED MULTI-OBJECTIVE OPTIMIZATION

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Human preference plays a significant role in measuring large language models and guiding them to align with human values. Unfortunately, current comparing-based evaluation (CBE) methods typically focus on a single optimization objective, failing to effectively utilize scarce yet valuable preference signals. To address this, we delve into key factors that can enhance the accuracy, convergence, and scalability of CBE: suppressing sampling bias, balancing descending process of uncertainty, and mitigating updating uncertainty. Following the derived guidelines, we propose UNICBE, a unified uniformity-driven CBE framework which simultaneously optimize these core objectives by constructing and integrating three decoupled sampling probability matrices, each designed to ensure uniformity in specific aspects. We further ablate the optimal tuple sampling and preference aggregation strategies to achieve efficient CBE. On the AlpacaEval benchmark, UNICBE saves over 17% of evaluation budgets while achieving a Pearson correlation with ground truth exceeding 0.995, demonstrating excellent accuracy and convergence. In scenarios where new models are continuously introduced, UNICBE can even save over 50% of evaluation costs, highlighting its improved scalability.

## 1 INTRODUCTION

The ongoing evolution of large language models (LLMs) has made it increasingly important to assess their alignment with human preferences (Dubois et al., 2024; Zheng et al., 2023). The preference signals provided by humans are crucial for accurately assessing and guiding models toward safe and reliable AGI (Ji et al., 2023; Jiang et al., 2024). However, the rapid iteration of LLMs in training and application scenarios has created a substantial demand for evaluation, complicating the acquisition of sufficient labor-intensive human preferences (Chiang et al., 2024; Cui et al., 2023). Therefore, exploring the use of precious preference signals for efficient model alignment evaluation is of great significance and requires long-term research.

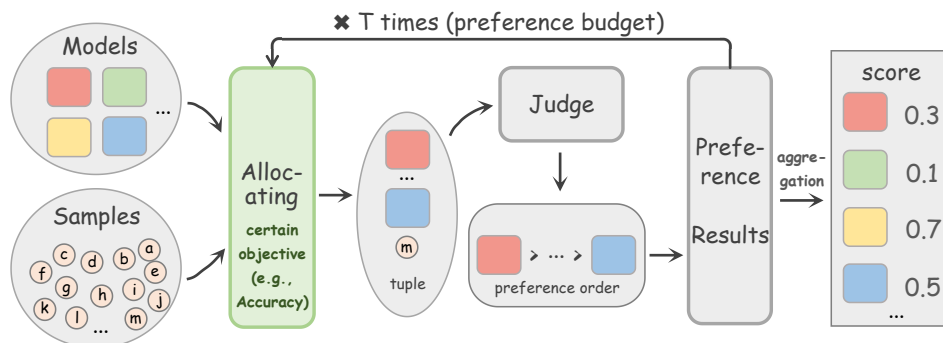


Figure 1: Flowchart of the process for comparing-based evaluation.

Current mainstream model evaluation paradigms include scoring-based evaluation (SBE) (Liu et al., 2023; Cai et al., 2023) and comparing-based evaluation (CBE) (Chiang et al., 2024; Dubois et al.,

2024). The former requires the judge to offer preference scores for individual responses, while the latter needs the judge to establish a preference order among multiple candidate model responses. By directly comparing the responses of different models, Zheng et al. (2023); Liu et al. (2024) confirm that CBE can more accurately assess model performance. However, the  $O(NM^2)$  evaluation overhead limits the practicality of CBE when there are  $M$  models to evaluate on  $N$  samples (Qin et al., 2024). To achieve efficient CBE, various methods have been explored Chiang et al. (2024); Zhou et al. (2024); Dubois et al. (2024). As shown in Figure 1, based on existing observational results, these methods iteratively allocate preference budget to the next (models, sample) tuple according to respective optimization objectives. Specific preference aggregation methods (e.g., ELO rating (Elo & Sloan, 1978)) are then applied to predict the model capability scores based on these preference results. Nevertheless, as shown in Table 1, the optimization objectives of these methods are often singular, failing to simultaneously achieve the accuracy, convergence, and scalability well. We will discuss this in detail in §2 and conduct experimental validation in §5.2.

Table 1: Optimization Objectives of widely applied CBE Methods. The number of '+' indicates the degree of optimization for the objective, which is discussed in §2 and measured in Table 2.

| Methods     | Qin et al. (2024)<br>RANDOM | Chiang et al. (2024)<br>ARENA | Dubois et al. (2024)<br>ALPACAEVAL | Ours<br>UNICBE |
|-------------|-----------------------------|-------------------------------|------------------------------------|----------------|
| Accuracy    | +                           | -                             | -                                  | ++             |
| Convergence | -                           | +                             | -                                  | ++             |
| Scalability | -                           | -                             | ++                                 | ++             |

To develop a method that can accurately assess model performance, quickly converge evaluation results, and ensure good scalability when new models are introduced, we theoretically analyze and summarize the following guidelines:

- Improving the **accuracy** of evaluation results relies on completely **uniform** sampling of tuple combinations, so as to mitigate sampling bias.
- Accelerating the **convergence** process involves ensuring the **uniformity** of the win rate uncertainty matrix during its descending process to reduce observation variance.
- Enhancing **scalability** requires sufficient budgets being allocated to new added models to ensure the **uniform** allocation among models, which helps reduce updating uncertainty.

Based on these insights, we propose UNICBE, a unified uniformity-driven framework that can achieve CBE with better accuracy, convergence and scalability. In each iteration of the evaluation process, we first establish sampling probability matrices under different optimization objectives respectively based on real-time preference results. Afterwards, we integrate these matrices to obtain a global sampling probability matrix. Furthermore, we explore various tuple sampling strategies and preference aggregation methods to achieve optimal evaluation results.

To comprehensively validate the effectiveness and generalizability of UNICBE, we conduct multiple experiments involving various types of judges (LLMs and humans), different benchmarks, varied model sets to be evaluated, diverse scenarios (static and dynamic), and multiple evaluation metrics. The main results indicate that, compared to the random sampling baseline, UNICBE saves over 17% of evaluation budgets when achieving the same assessment accuracy (with a Pearson coefficient exceeding 0.995 with the ground truth), demonstrating significantly better convergence and accuracy than other baselines. Furthermore, in scenarios where new models are continuously introduced, UNICBE can even save over 50% of evaluation costs compared to random sampling, showcasing excellent scalability.

## 2 RELATED WORK

Comparative preference signals have long been used for model training (Ouyang et al., 2022; Touvron et al., 2023) and evaluation (Chiang et al., 2024; Yuan et al., 2024). Centered around comparing-based evaluation, we will discuss existing budget allocation strategies and preference aggregation methods below.

**Budget Allocation** Many efforts have been made to explore preference budget allocation approaches. The most naive allocation method is to randomly select (models, sample) tuple for judging

each time until the preset preference budget is reached (Qin et al., 2024). This method ensures a relatively uniform sampling across different tuple combinations in expectation, thereby guaranteeing the accuracy of evaluation results according to our derivation in §3.2. Arena (Chiang et al., 2024) aims to sample model pairs proportionally to the variance gradient of win rate at each step, seeking to accelerating the convergence of evaluation by reducing the uncertainty of the observed win rate matrix in a greedy manner. AlpacaEval (Dubois et al., 2024) measures model performance by comparing the models under evaluation with a fixed reference model. Different models under evaluation are expected to be assessed on the same set. Thus, when new models are introduced, preference budget is prioritized for them to stabilize the estimation of their capabilities, thereby achieving good scalability. Despite these methods performing well in their intended objectives, they cannot achieve a balance among accuracy, convergence, and scalability. This makes it imperative to explore better preference budget allocation strategy that can effectively reconcile all these attributes.

**Preference Aggregation** Due to the possibility that the same group of models may exhibit different ranking relationships across different samples, it is essential to estimate the global model capability scores to better fit these non-transitive preference results. Dubois et al. (2024); Zheng et al. (2023) directly use the average pair-wise win rate of each model as a measure of its capability. Feng et al. (2024); Wu & Aji (2023) apply the classical Elo rating system (Elo & Sloan, 1978) (see the Appendix B.1 for detailed introduction) by treating the evaluation process as a sequence of model battles in order to derive model scores. Fageot et al. (2024); Chiang et al. (2024) employ the Bradley-Terry model (Bradley & Terry, 1952) (see Appendix B.2 for detailed introduction) to estimate model scores by maximizing the likelihood of the comparison results between models. We will systematically compare the effectiveness of these preference aggregation methods in §5.3.

### 3 PRELIMINARY

In this section, we start by symbolically introducing the working process of CBE. Afterwards, we introduce the key objectives for achieving efficient CBE: accuracy, convergence, and scalability, and analyze the factors that influence them. We mainly discuss the pair-wise evaluation scenario (where the judge provides preference between two models per time) for its wide applications (Tashu & Horváth, 2018; Qin et al., 2024). Actually, list-wise preferences can be easily converted into pair-wise ones, as demonstrated in §5.4, so the discussions below are general for CBE.

#### 3.1 PROCESS OF CBE

Generally, a CBE method  $f$  can be divided into three parts: budget allocation strategy  $f^{ba}$ , tuple sampling strategy  $f^{ts}$ , and preference aggregation strategy  $f^{pa}$ . Given benchmark  $\mathcal{D} : s_{1:N}$  and models under evaluation  $\mathcal{M} : m_{1:M}$ , we iterate the following steps: *step 1.* applying  $f^{ba}$  to attain sampling matrix  $P^l$  at iteration  $l$ , where  $P_{i,j,k}^l$  denotes the probability to select tuple  $(m_i, m_j, s_k)$  for judging; *step 2.* applying  $f^{ts}$  to sample certain tuple  $(m^{l1}, m^{l2}, s^l)$  based on  $P^l$ ; *step 3.* attaining preference result  $r^l$  from the judge, where  $r^l \in [0, 1]$  denotes the degree  $m^{l1}$  wins over  $m^{l2}$  (0.5 means tie). We stop this iterative process when the preset preference budget  $T$  is achieved and then apply  $f^{pa}$  on preference results  $\{(m^{l1}, m^{l2}, s^l, r^l)\}_{l=1}^T$  to attain estimated model scores  $u_{1:M}$ .

#### 3.2 ACCURACY

Theoretically, if we have a budget of  $\hat{T} = \frac{NM(M-1)}{2}$ , we can explore all tuples to obtain the ground truth estimation for the model scores  $\hat{u}_{1:M}$ . However, typically  $T$  is much smaller than  $\hat{T}$  in reality considering the preciousness of preference signals. Previous studies (Vabalas et al., 2019; Kossen et al., 2021) have discussed the risks of introducing sampling bias in incomplete sampling scenarios, which we believe could similarly lead to potential risks in CBE. Considering that the content of each sample is  $(m^{l1}, m^{l2}, s^l)$ , we think the sample bias exists across both samples and models.

**Bias across Samples.** Since different models may excel at answering different types of queries, the model scores can vary depending on the sampled data:

$$u_t = f^{pa}(\{(m_i, m_j, s_k, r_{i,j,k})\}_{i \in 1:M, j \in i+1:M})_t = \hat{u}_t + \eta_{m_t, \cdot, s_k} \quad \text{for } \forall t, k \quad (1)$$

162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215

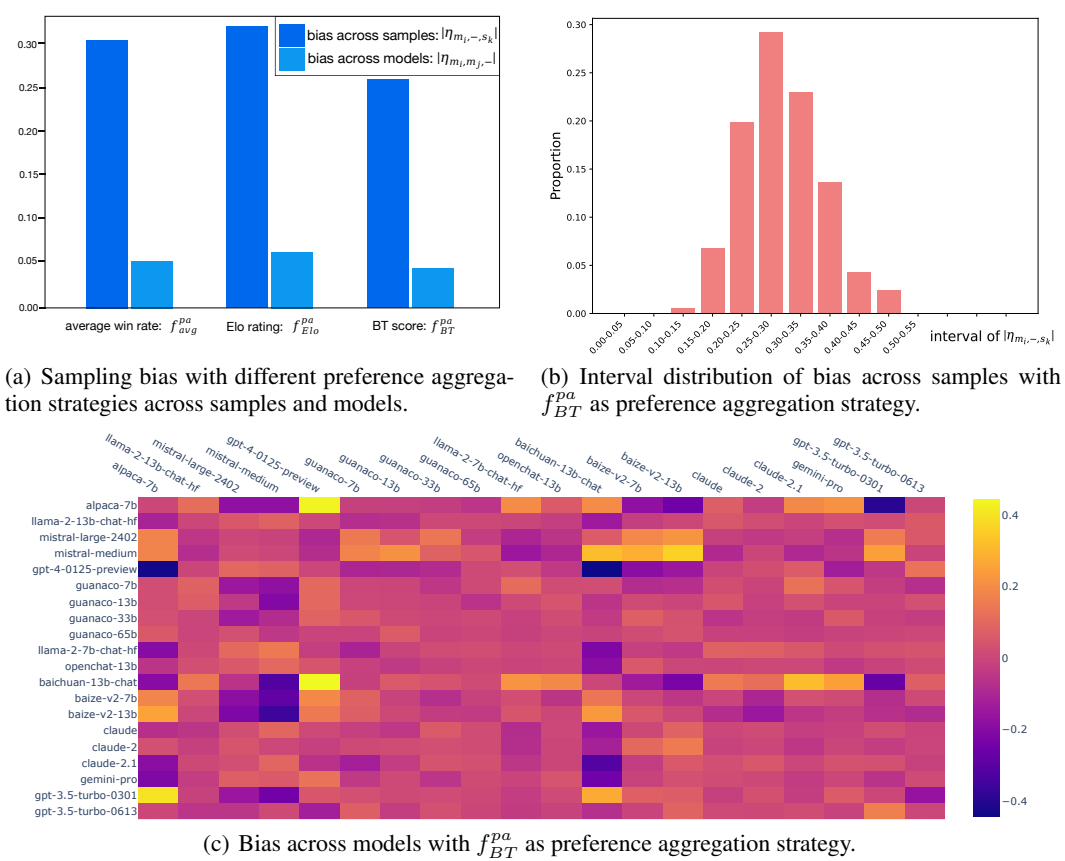


Figure 2: Analyses of potential sampling bias risks in CBE.

where  $\eta_{m_t, -, s_k}$  represents the bias between the observed model score  $u_t$  of  $m_t$  and the ground truth  $\hat{u}_t$  when solely assessing on sample  $s_k$ . To verify this, we conduct experiments on the AlpacaEval benchmark (Dubois et al., 2024) using GPT-4o (OpenAI, 2024) as the judge across randomly selected 20 LLMs (listed in Figure 2(c)). We first traversed all model pairs for samples  $s_{1:N}$  to obtain corresponding  $N$  sets of preference results and then calculate the respective  $|\eta_{m_i, -, s_k}|$  for  $i \in 1 : M$  and  $k \in 1 : N$  according to equation 1 (model scores are normalized to an average of 1). We calculate the average value of  $|\eta_{m_i, -, s_k}|$  across models and samples using different preference aggregation strategies  $f^{pa}$  discussed in §2. As shown in Figure 2(a), with all kinds of  $f^{pa}$ , the average difference between the model scores estimated on single sample and the ground truth values exceeds 0.25, indicating a significant bias across samples. We further analyze the proportion of samples with different biases using  $f_{BT}^{pa}$  in Figure 2(b) and find that they overall follow a Gaussian distribution, showing the wide existence of sample bias in CBE.

**Bias across Models.** Just as humans may perform differently when facing different opponents, models may also have varying scores when competing against different models:

$$u_i = f^{pa}(\{(m_i, m_j, s_k, r_{i,j,k})\}_{k \in 1:N})_i = \hat{u}_i + \eta_{m_i, m_j, -} \quad \text{for } \forall i, j \quad (2)$$

We validate this from two perspectives: (1) We calculate the average  $|\eta_{m_i, m_j, -}|$  according to equation 2 like the process above and show the results in Figure 2(a). Overall, although the bias across models is significantly lower than the bias across samples, it still exists at a scale around 0.05. We further visualize the pair-wise model score bias in Figure 2(c) to validate its wide existence. (2) We obtain over 1.7 million pairwise preference results across 129 LLMs collected by Chatbot Arena<sup>1</sup>. After excluding pairs with fewer than 50 comparisons, we calculate the pairwise win rates and find non-transitivity in 81 model triplets (win rate:  $A > B$ ,  $B > C$ ,  $C > A$ ), which also verifies the existence of bias across models.

<sup>1</sup>[https://storage.googleapis.com/arena\\_external\\_data/public/clean\\_battle\\_20240814\\_public.json](https://storage.googleapis.com/arena_external_data/public/clean_battle_20240814_public.json)

**Uniform Allocation Brings the Least Bias.** Based on the discussions above, we analyze the budget allocation strategy that can introduce the least bias. Considering the presence of sampling bias, the estimation error of  $u_i$  with  $T$  evaluation budget can be expressed as follows:

$$u_i - \hat{u}_i = \sum_{l=1}^T \mathbf{1}_{m^{l1}=m_i} \times \eta_{m^{l1}, m^{l2}, s^l} \quad (3)$$

Considering that  $u = \hat{u}$  when all the tuples are traversed, we have the following equation:

$$0 = u_i - \hat{u}_i = \sum_{j=1}^M \sum_{k=1}^N \eta_{i,j,k} \quad \text{for } \forall i \quad (4)$$

The goal of obtaining the minimum estimation error for  $u_i$  is transformed into sampling  $T$  numbers (equation 3) from  $MN$  numbers that sum to zero (equation 4), such that the absolute value of the sum of these  $T$  numbers is minimized. We have provided a detailed proof in Appendix A that the best strategy is completely uniform sampling. ***This denotes that the score estimation error can be minimized when the preference budgets are uniformly distributed across models and samples to bring the least sampling bias.***

### 3.3 CONVERGENCE

During the evaluation process, as new preference results are continuously observed, the estimated values of the models win rate matrix and model scores also change constantly. To accelerate the convergence process, we analyze the uncertainty of the win rate matrix as follows. Defining that:

$$X_{i,j}^l = \frac{1}{P_{i,j}^l} r^l \mathbf{1}_{m^{l1}=m_i \ \& \ m^{l2}=m_j} + \frac{1}{P_{j,i}^l} (1 - r^l) \mathbf{1}_{m^{l1}=m_j \ \& \ m^{l2}=m_i} \quad (5)$$

The unbiased estimated win rate matrix  $\Phi$  at iteration  $L$  can be calculated as follows:

$$\Phi^L = \frac{1}{L} \sum_{l=1}^L X^l \quad (6)$$

We further estimate the variance matrix  $\Theta$  as:

$$\Theta^L = \frac{1}{L} \sum_{l=1}^L (X^l - \Phi^L) \circ (X^l - \Phi^L) \quad (7)$$

Denoting if the model pair  $(m_i, m_j)$  has been compared on sample  $s_k$  after  $l$  iterations as  $C_{i,j,k}^l$ , the uncertainty (standard deviation) of each element in the win rate matrix is as follows:

$$\epsilon_{i,j}^l = \sqrt{\frac{\Theta_{i,j}^l}{\sum_{k=1}^N C_{i,j,k}^l}} \quad (8)$$

Allocating the next preference budget on  $(m_i, m_j)$  can reduce the uncertainty of their win rate by:

$$\sqrt{\frac{\Theta_{i,j}^l}{\sum_{k=1}^N C_{i,j,k}^l}} - \sqrt{\frac{\Theta_{i,j}^l}{\sum_{k=1}^N C_{i,j,k}^l + 1}} \quad (9)$$

Considering that our core objective is to conduct accurate capability assessments for all models and estimate their ranking relationship, ***we should globally ensure the uniformity of the win rate uncertainty matrix during its descending process to achieve smooth evaluation convergence.***

### 3.4 SCALABILITY

Due to the continuous emergence of new LLMs, the demand for scalability in evaluation method is becoming increasingly prominent (Chern et al., 2024). Considering that we have evaluated  $m_{1:M}$  with  $T$  budgets, when model  $m_{M+1}$  is introduced for assessment, a well-scalable CBE method

should be able to quickly calibrate the capability estimates of  $m_{1:M+1}$  with minimal additional preference budget. In this scenario, at the beginning stage when  $m_{M+1}$  is introduced,  $\text{avg}(C_{M+1,-,-})$  is much smaller than  $\text{avg}(C_{\neq M+1,-,-})$ . According to equation 8, the uncertainty at this point mainly arises from  $\epsilon_{M+1}$ , which is also intuitively easy to understand. **Therefore, the key to improving scalability lies in allocating sufficient evaluation budgets to the newly added models to ensure the uniform allocation among models, reducing the updating uncertainty.**

## 4 UNICBE

The discussions above reveal guidelines for strengthening scalability, accuracy, and convergence in CBE. Based on this, we propose UNICBE, a unified uniformity-driven framework that can simultaneously enhance these objectives well.

### 4.1 BUDGET ALLOCATION

To ensure the uniformity of tuple combination sampling for minimizing the introduction of sampling bias according to §3.2, we construct  $P^{acc-l}$  at iteration  $l$  as follows:

$$P_{i,j,k}^{acc-l} = \alpha^{-\sum_{k=1}^N C_{i,j,k}^l} \times \alpha^{-\sum_{i=1}^M C_{i,j,k}^l} \times \alpha^{-\sum_{j=1}^M C_{i,j,k}^l} \quad (10)$$

where  $\sum_{k=1}^N C_{i,j,k}^l$  denotes the times model pair  $(m_i, m_j)$  has been compared,  $\sum_{i=1}^M C_{i,j,k}^l$  and  $\sum_{j=1}^M C_{i,j,k}^l$  denote the times model  $m_i$  and  $m_j$  has been tested on  $s_k$  respectively. If certain model-model combination or model-sample combination have been sampled multiple times, equation 10 will reduce the probability of such combinations being selected again, thereby achieving sufficient uniformity to minimize the introduction of bias between models and samples, respectively.

To accelerate the convergence of evaluation results, we construct  $P^{con-l}$  according to §3.3 as follows:

$$P_{i,j,k}^{con-l} = \epsilon_{i,j}^l \quad (11)$$

Sampling specific model pair helps reduce the uncertainty of their win rate estimation according to equation 9. By sampling proportionally to the win rate uncertainty matrix, we can uniformly decrease the uncertainty for each model pair, thereby facilitating convergence.

We construct  $P^{sca-l}$  to allocate more preference budget to the newly introduced model so as to improving the scalability according to §3.4 as follows:

$$P_{i,j,k}^{sca-l} = \alpha^{-\sum_{k=1}^N \sum_{i=1}^M C_{i,j,k}^l} \times \alpha^{-\sum_{k=1}^N \sum_{j=1}^M C_{i,j,k}^l} \quad (12)$$

Finally, we integrate the matrices mentioned above to obtain  $P^l$ , ensuring that sampling according to  $P^l$  can simultaneously balance the accuracy, convergence, and scalability of evaluation results:

$$P^l = \frac{P^{acc-l} \circ P^{con-l} \circ P^{sca-l}}{\sum (P^{acc-l} \circ P^{con-l} \circ P^{sca-l})} \quad (13)$$

### 4.2 TUPLE SAMPLING

After obtaining  $P^l$ , we need to sample tuples for judging based on it. Two tuple sampling strategies are considered:

- **probabilistic sampling**  $f_p^{ts}$  means sampling tuple directly according to  $P^l$ .
- **greedy sampling**  $f_g^{ts}$  means selecting the tuple with the maximum probability in  $P^l$ .

The default tuple sampling strategy of UNICBE is  $f_g^{ts}$ , which can avoid the suboptimal achievement of objectives due to uncertainty in the sampling process.

### 4.3 PREFERENCE AGGREGATION

As discussed in §2, mainstream preference aggregation strategies include averaging win rate  $f_{avg}^{pa}$ , Elo rating system  $f_{Elo}^{pa}$  and Bradley-Terry model  $f_{BT}^{pa}$ . In our preliminary experiment (Figure 2(c)) we observe that  $f_{BT}^{pa}$  can better alleviate sampling bias, for which we choose it as our default setting.

## 5 EXPERIMENTS

Centered around UNICBE, we will empirically compare its performance with baselines and validate its scalability in §5.2, explore the optimal variants in §5.3, and demonstrate its generalizability under different settings in §5.4.

### 5.1 EXPERIMENTAL SETTINGS

**Benchmarks.** We choose AlpacaEval (Dubois et al., 2024) and MT-Bench (Zheng et al., 2023) benchmarks for our experiments. For AlpacaEval, we use its default version which includes 805 high-quality human annotated instructions and corresponding responses from multiple LLMs. We randomly choose 20 LLMs (listed in Figure 2(c)) for experiments, with GPT-4o and GPT-3.5-turbo as judges (see Appendix D for the prompt). For MT-Bench, we use the released responses from the all 6 LLMs and corresponding human preferences for experiments.

**Baselines.** We choose widely applied<sup>2</sup> methods RANDOM, ARENA and ALPACAEVAL as baselines, which have been discussed in §2 and listed in Table 1.

**Metrics.** To assess the effectiveness of the CBE methods, we evaluate the accuracy of both the estimated model pair-wise win rates and the model scores. We calculate the average absolute error between the estimated win rates and corresponding ground truth (the estimates when  $T = \hat{T}$ ). We calculate the Spearman correlation coefficient  $r_s$  between the predicted model scores and the corresponding ground truth to evaluate the accuracy of the model’s rank-order relationship, and the Pearson correlation coefficient  $r_p$  to assess the accuracy of the linear relationship.

**Details.** To ensure the reliability of the experimental results, for each setting, we randomly select  $M$  (default to 15 for AlpacaEval and 5 for MT-Bench) models and  $N$  (default to 805 for AlpacaEval and 700 for MT-Bench) samples, and report the average results across 10,000 random seeds. We don’t observe obvious performance difference in preliminary experiments when varying  $\alpha$  within the range of [1.5, 3] (we conduct a detailed discussion about this in Appendix F.1), thus we set the default value of  $\alpha$  as 2 in our experiments.

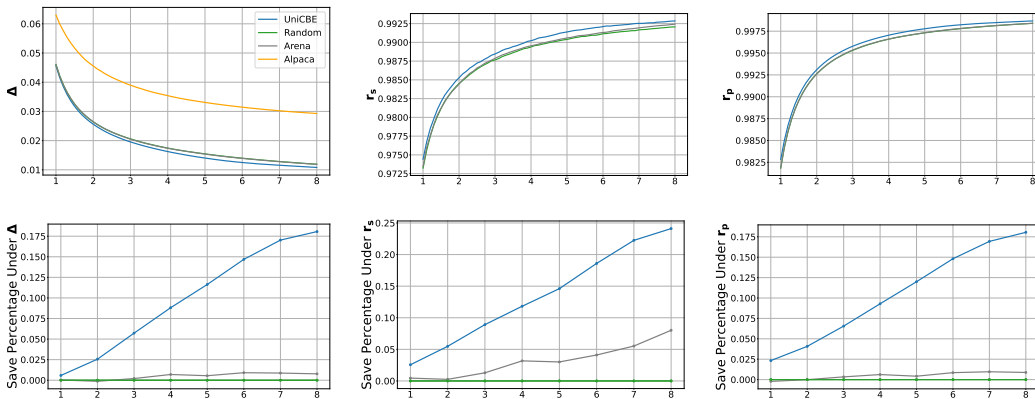


Figure 3: Results of compared CBE methods with GPT-4o as the judge on AlpacaEval benchmark. The X-axis (applicable to all plots below) represents the preference budget ( $k$ ).  $\Delta$  denotes the mean absolute error of the estimated win rate.  $r_s$  and  $r_p$  denote the Spearman and Pearson correlations between the the estimated model scores and the ground truth respectively.

### 5.2 MAIN RESULTS

**Accuracy and Convergence.** The results of compared CBE methods on AlpacaEval benchmark with GPT4-turbo as the judge are shown in Figure 3. To better illustrate the results, we also calculate the percentage of preference budget saved by each method compared to RANDOM baseline when

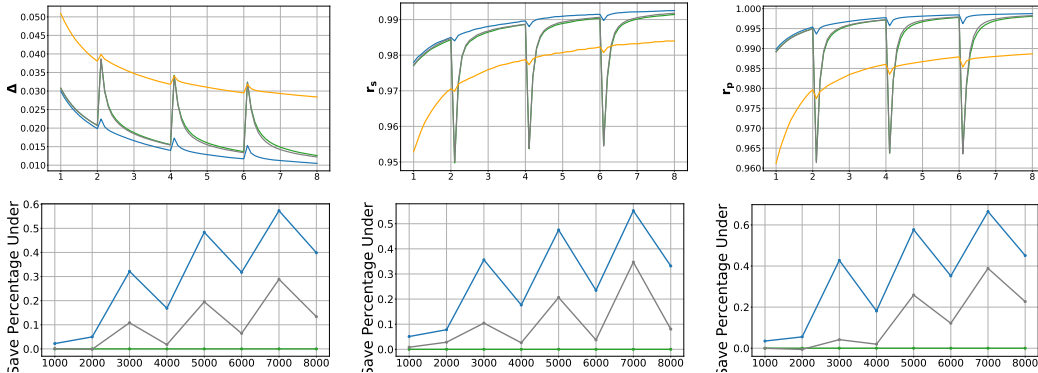
<sup>2</sup>[https://tatsu-lab.github.io/alpaca\\_eval/](https://tatsu-lab.github.io/alpaca_eval/), <https://lmarena.ai/>

378 achieving the same performance. In terms of performance, ALPACAEVAL  $\ll$  RANDOM  $<$  ARENA  
 379  $<$  UNICBE. To understand the differences in the performance of each method, we quantitatively  
 380 analyze them based on the guidelines summarized in § 3. To achieve accuracy, convergence, and  
 381 scalability, it is supposed to allocate the preference budget in a way that ensures uniformity across  
 382 tuples, uniformity across model pairs in win-rate uncertainty, and uniformity across models. We  
 383 calculate the cosine similarity between the allocation results of these methods and the correspond-  
 384 ing expected uniform vectors for each objective as a measure, denoted as  $\beta_{acc}$ ,  $\beta_{con}$ , and  $\beta_{sca}$ ,  
 385 respectively (see Appendix E for calculating process). As shown in Table 2, the fixed inclusion of  
 386 the reference model in the tuple selection of ALPACAEVAL compromises uniformity across multi-  
 387 ple aspects, thereby resulting in lower  $\beta$  values and significantly poorer performance. ARENA and  
 388 RANDOM respectively improve the balance of uncertainty and suppression of sampling bias, result-  
 389 ing in higher  $\beta_{con}$  and  $\beta_{acc}$  values. Following our guidelines, UNICBE improves  $\beta_{con}$ ,  $\beta_{acc}$ , and  
 390  $\beta_{sca}$  simultaneously and save over 17% of the preference budget compared to RANDOM with a  $\Delta$   
 391 close to 0.01, showcasing improved accuracy and convergence.

392 Table 2: The measurement results of the achievement of objectives in §3 for the compared methods.

| 393 Methods       | RANDOM | ARENA | ALPACAEVAL | UNICBE |
|-------------------|--------|-------|------------|--------|
| 394 $\beta_{acc}$ | .5803  | .5725 | .0925      | .7364  |
| 395 $\beta_{con}$ | .9081  | .9172 | .3515      | .9228  |
| 396 $\beta_{sca}$ | .9972  | .9945 | .4987      | .9997  |

400 **Scalability.** To analyze scalability, we establish a scenario where we initially have 11 models  
 401 awaiting assessment, and new models are sequentially added every 2000 samplings. As shown in  
 402 Figure 4, Whenever a new model is introduced, UNICBE can rapidly stabilize the performance  
 403 through adaptive preference allocation skewing for the new model, saving over 50% of the budget  
 404 compared to the RANDOM baseline. In contrast, ARENA and RANDOM exhibit poorer scalability  
 405 since they do not consider scalability as optimization objective. Although the budget allocated to  
 406 the reference model is significantly more than that for other models, resulting in a lower  $\beta_{sca}$  for  
 407 ALPACAEVAL, the strategy of automatically allocating the budget to the new introduced models  
 408 also provides it with good scalability.



422 Figure 4: Results of compared CBE methods in the scenario where new model are consistently  
 423 introduced every 2000 iterations.

424  
 425 **5.3 VARIANTS ABLATIONS**

427 **Budget Allocation Objectives.** We test the impact of different optimization objectives by remov-  
 428 ing  $P^{acc}$ ,  $P^{con}$ , and  $P^{sca}$  from equation 13 separately. As shown in Figure 5, the significant perfor-  
 429 mance degradation observed when removing  $P^{acc}$  from UNICBE indicates that mitigating sampling  
 430 bias to improve accuracy is the most critical factor in achieving efficient CBE. Furthermore, we find  
 431 that  $P^{con}$  has a considerable impact on  $r_s$ . We hypothesize that this is because balancing the un-  
 certainty among different models helps prevent any one model from having a significant ranking





**Variied Number of Models and Samples.** Finally, as shown in Figure 7, we conduct experiments by varying the number of models  $M$  and samples  $N$ . It can be observed that UNICBE achieves significantly better results compared to all the baselines under these settings, especially when  $M$  and  $N$  are large.

**List-wise Preference.** UNICBE can also be applied to list-wise preference. Suppose the judge provides a preference ranking for  $K$  models each time. We can compute a  $K + 1$  dimensional  $P$ , similar to equation 13, and sample to obtain a tuple  $(m^{l-1}, \dots, m^{l-K}, s^l)$ . From the judge’s ranking of this tuple, we derive  $\frac{K(K-1)}{2}$  pair-wise preferences. Figure 8 shows the results for the case where  $K = 3$ . It can be seen that UNICBE achieves a savings compared to RANDOM of over 30% in this setting. This may be due to the fact that list-wise preference results in  $\frac{K(K-1)}{2}$  pair-wise preferences concentrated among the  $K$  models, exacerbating the sampling bias. Therefore, UNICBE is even more needed to suppress this effect with list-wise preference.

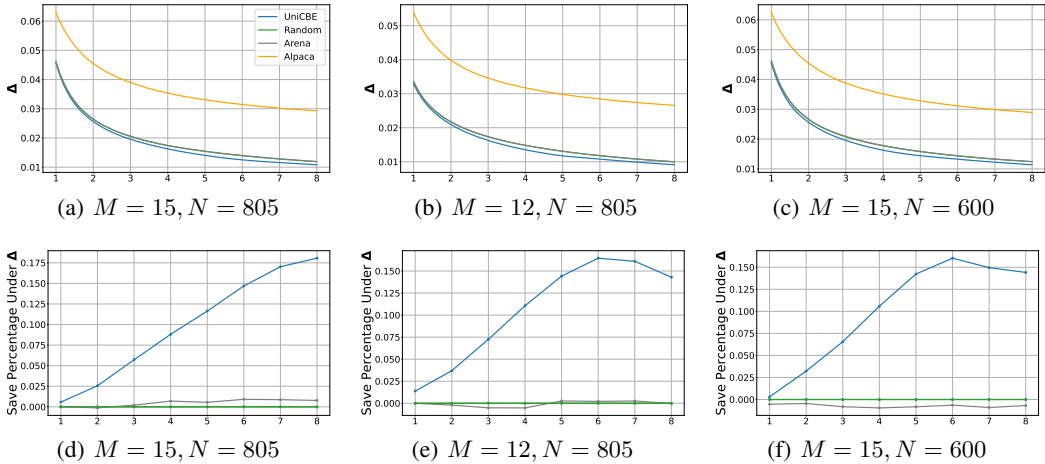


Figure 7: Effects of number of models under evaluation  $M$  and number of samples  $N$ . The results are obtained with GPT-4o as the judge on AlpacaEval.

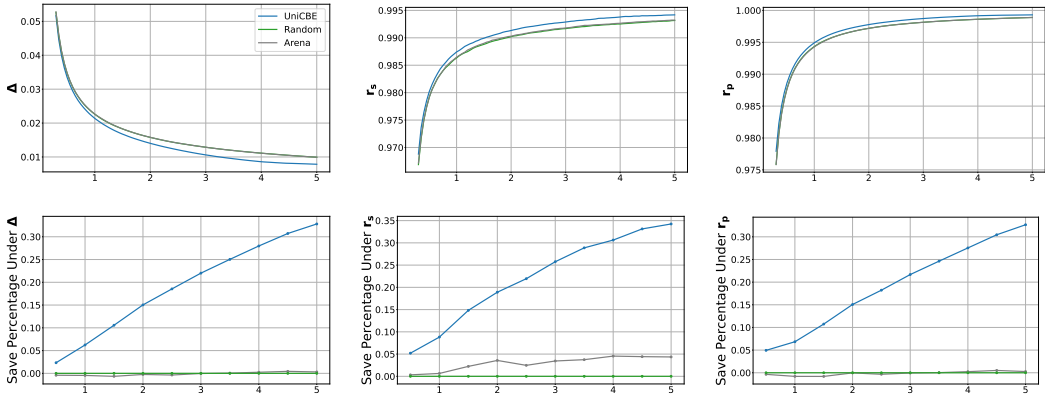


Figure 8: Performance of CBE methods with list-wise preference of GPT-4o on AlpacaEval.

## 6 CONCLUSIONS

The existing comparing-based evaluation methods are ineffective in fully utilizing valuable preference signals due to their constrained optimization objectives. Our in-depth analysis reveals that the key to enhancing CBE lies in mitigating sampling bias, balancing the descent process of uncertainty, and suppressing the updating uncertainty. Based on this, we propose the UNICBE framework that simultaneously optimizes the aforementioned objectives by promoting uniformity in corresponding aspects to enhance accuracy, convergence, and scalability. Comprehensive experiments and analyses confirm the strong effectiveness, improved scalability, and good generalizability of UNICBE.

## REFERENCES

- 540  
541  
542 Meriem Boubdir, Edward Kim, Beyza Ermis, Sara Hooker, and Marzieh Fadaee. Elo uncovered: Ro-  
543 bustness and best practices in language model evaluation. *CoRR*, abs/2311.17295, 2023. doi: 10.  
544 48550/ARXIV.2311.17295. URL <https://doi.org/10.48550/arXiv.2311.17295>.
- 545  
546 Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method  
547 of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- 548  
549 Tianchi Cai, Xierui Song, Jiyang Jiang, Fei Teng, Jinjie Gu, and Guannan Zhang. ULMA: uni-  
550 fied language model alignment with demonstration and point-wise human preference. *CoRR*,  
551 abs/2312.02554, 2023. doi: 10.48550/ARXIV.2312.02554. URL <https://doi.org/10.48550/arXiv.2312.02554>.
- 552  
553 Steffi Chern, Ethan Chern, Graham Neubig, and Pengfei Liu. Can large language models be  
554 trusted for evaluation? scalable meta-evaluation of llms as evaluators via agent debate. *CoRR*,  
555 abs/2401.16788, 2024. doi: 10.48550/ARXIV.2401.16788. URL <https://doi.org/10.48550/arXiv.2401.16788>.
- 556  
557 Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li,  
558 Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Sto-  
559 ica. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first*  
560 *International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.  
561 OpenReview.net, 2024. URL <https://openreview.net/forum?id=3MW8GKNyzI>.
- 562  
563 Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu,  
564 and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *CoRR*,  
565 abs/2310.01377, 2023. doi: 10.48550/ARXIV.2310.01377. URL <https://doi.org/10.48550/arXiv.2310.01377>.
- 566  
567 Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled  
568 alpacaeval: A simple way to debias automatic evaluators. *CoRR*, abs/2404.04475, 2024. doi: 10.  
569 48550/ARXIV.2404.04475. URL <https://doi.org/10.48550/arXiv.2404.04475>.
- 570  
571 Arpad E Elo and Sam Sloan. The rating of chessplayers: Past and present. (*No Title*), 1978.
- 572  
573 Julien Fageot, Sadegh Farhadkhani, Lê-Nguyễn Hoàng, and Oscar Villemaud. Generalized bradley-  
574 terry models for score estimation from paired comparisons. In *Thirty-Eighth AAAI Confer-*  
575 *ence on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications*  
576 *of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Arti-*  
577 *ficial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 20379–20386.  
AAAI Press, 2024. doi: 10.1609/AAAI.V38I18.30020. URL <https://doi.org/10.1609/aaai.v38i18.30020>.
- 578  
579 Sebastian Farquhar, Yarin Gal, and Tom Rainforth. On statistical bias in active learning: How and  
580 when to fix it. In *9th International Conference on Learning Representations, ICLR 2021, Virtual*  
581 *Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL [https://openreview.net/](https://openreview.net/forum?id=JiYq3eqTKY)  
582 [forum?id=JiYq3eqTKY](https://openreview.net/forum?id=JiYq3eqTKY).
- 583  
584 Kehua Feng, Keyan Ding, Kede Ma, Zhihua Wang, Qiang Zhang, and Huajun Chen. Sample-  
585 efficient human evaluation of large language models via maximum discrepancy competition.  
586 *CoRR*, abs/2404.08008, 2024. doi: 10.48550/ARXIV.2404.08008. URL <https://doi.org/10.48550/arXiv.2404.08008>.
- 587  
588 Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen,  
589 Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved  
590 safety alignment of LLM via a human-preference dataset. In *Advances in Neural*  
591 *Information Processing Systems 36: Annual Conference on Neural Information*  
592 *Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16,*  
593 *2023*. URL [http://papers.nips.cc/paper\\_files/paper/2023/](http://papers.nips.cc/paper_files/paper/2023/hash/4dbb61cb68671edc4ca3712d70083b9f-Abstract-Datasets_and_Benchmarks.html)  
[hash/4dbb61cb68671edc4ca3712d70083b9f-Abstract-Datasets\\_and\\_](http://papers.nips.cc/paper_files/paper/2023/hash/4dbb61cb68671edc4ca3712d70083b9f-Abstract-Datasets_and_Benchmarks.html)  
[Benchmarks.html](http://papers.nips.cc/paper_files/paper/2023/hash/4dbb61cb68671edc4ca3712d70083b9f-Abstract-Datasets_and_Benchmarks.html).

- 594 Ruili Jiang, Kehai Chen, Xuefeng Bai, Zhixuan He, Juntao Li, Muyun Yang, Tiejun Zhao, Liqiang  
595 Nie, and Min Zhang. A survey on human preference learning for large language models. *CoRR*,  
596 abs/2406.11191, 2024. doi: 10.48550/ARXIV.2406.11191. URL [https://doi.org/10.](https://doi.org/10.48550/arXiv.2406.11191)  
597 [48550/arXiv.2406.11191](https://doi.org/10.48550/arXiv.2406.11191).
- 598 Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient  
599 model evaluation. In *Proceedings of the 38th International Conference on Machine Learning,*  
600 *ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning*  
601 *Research*, pp. 5753–5763. PMLR, 2021. URL [http://proceedings.mlr.press/v139/](http://proceedings.mlr.press/v139/kossen21a.html)  
602 [kossen21a.html](http://proceedings.mlr.press/v139/kossen21a.html).
- 603 Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval:  
604 NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Con-*  
605 *ference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, De-*  
606 *cember 6-10, 2023*, pp. 2511–2522. Association for Computational Linguistics, 2023. doi:  
607 10.18653/V1/2023.EMNLP-MAIN.153. URL [https://doi.org/10.18653/v1/2023.](https://doi.org/10.18653/v1/2023.emnlp-main.153)  
608 [emnlp-main.153](https://doi.org/10.18653/v1/2023.emnlp-main.153).
- 609 Yinong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulic, Anna Korhonen, and Nigel  
610 Collier. Aligning with human judgement: The role of pairwise preference in large language model  
611 evaluators. *CoRR*, abs/2403.16950, 2024. doi: 10.48550/ARXIV.2403.16950. URL <https://doi.org/10.48550/arXiv.2403.16950>.
- 612 Frederic M Lord and Melvin R Novick. *Statistical theories of mental test scores*. IAP, 2008.
- 613 OpenAI. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>, 2024. Accessed:  
614 2024-05-26.
- 615 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin,  
616 Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser  
617 Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan  
618 Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In  
619 *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Informa-*  
620 *tion Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December*  
621 *9, 2022*, 2022. URL [http://papers.nips.cc/paper\\_files/paper/2022/hash/](http://papers.nips.cc/paper_files/paper/2022/hash/blefde53be364a73914f58805a001731-Abstract-Conference.html)  
622 [blefde53be364a73914f58805a001731-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2022/hash/blefde53be364a73914f58805a001731-Abstract-Conference.html).
- 623 Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail  
624 Yurochkin. tinybenchmarks: evaluating llms with fewer examples. In *Forty-first International*  
625 *Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenRe-  
626 view.net, 2024. URL <https://openreview.net/forum?id=qAml3FpfhG>.
- 627 Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu,  
628 Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. Large language models are  
629 effective text rankers with pairwise ranking prompting. In *Findings of the Association for Compu-*  
630 *tational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 1504–1518. As-  
631 sociation for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-NAACL.97.  
632 URL <https://doi.org/10.18653/v1/2024.findings-naacl.97>.
- 633 Tsegaye Misikir Tashu and Tomás Horváth. Pair-wise: Automatic essay evaluation using word  
634 mover’s distance. In *Proceedings of the 10th International Conference on Computer Supported*  
635 *Education, CSEDU 2018, Funchal, Madeira, Portugal, March 15-17, 2018, Volume 1*, pp. 59–66.  
636 SciTePress, 2018. doi: 10.5220/0006679200590066. URL [https://doi.org/10.5220/](https://doi.org/10.5220/0006679200590066)  
637 [0006679200590066](https://doi.org/10.5220/0006679200590066).
- 638 Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Niko-  
639 lay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher,  
640 Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy  
641 Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn,  
642 Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel  
643 Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya  
644 Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar

- 648 Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan  
649 Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen  
650 Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan  
651 Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez,  
652 Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-  
653 tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL  
654 <https://doi.org/10.48550/arXiv.2307.09288>.
- 655 Andrius Vabalas, Emma Gowen, Ellen Poliakoff, and Alexander J Casson. Machine learning algo-  
656 rithm validation with a limited sample size. *PloS one*, 14(11):e0224365, 2019.
- 657 Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking mod-  
658 els with much fewer examples. In *Proceedings of the 18th Conference of the European Chapter*  
659 *of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Ju-*  
660 *lian’s, Malta, March 17-22, 2024*, pp. 1576–1601. Association for Computational Linguistics,  
661 2024. URL <https://aclanthology.org/2024.eacl-long.95>.
- 662 Minghao Wu and Alham Fikri Aji. Style over substance: Evaluation biases for large language  
663 models. *CoRR*, abs/2307.03025, 2023. doi: 10.48550/ARXIV.2307.03025. URL [https://](https://doi.org/10.48550/arXiv.2307.03025)  
664 [doi.org/10.48550/arXiv.2307.03025](https://doi.org/10.48550/arXiv.2307.03025).
- 665 Peiwen Yuan, Shaoxiong Feng, Yiwei Li, Xinglin Wang, Boyuan Pan, Heda Wang, Yao Hu, and  
666 Kan Li. Batcheval: Towards human-like text evaluation. In *Proceedings of the 62nd Annual*  
667 *Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024,*  
668 *Bangkok, Thailand, August 11-16, 2024*, pp. 15940–15958. Association for Computational Lin-  
669 guistics, 2024. URL <https://aclanthology.org/2024.acl-long.846>.
- 670 Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao  
671 Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez,  
672 and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances*  
673 *in Neural Information Processing Systems 36: Annual Conference on Neural Informa-*  
674 *tion Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 -*  
675 *16, 2023*, 2023. URL [http://papers.nips.cc/paper\\_files/paper/2023/](http://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_)  
676 [hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets\\_and\\_](http://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_)  
677 [Benchmarks.html](http://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_).
- 678 Jin Peng Zhou, Christian K. Belardi, Ruihan Wu, Travis Zhang, Carla P. Gomes, Wen Sun, and  
679 Kilian Q. Weinberger. On speeding up language model evaluation. *CoRR*, abs/2407.06172, 2024.  
680 doi: 10.48550/ARXIV.2407.06172. URL [https://doi.org/10.48550/arXiv.2407.](https://doi.org/10.48550/arXiv.2407.06172)  
681 [06172](https://doi.org/10.48550/arXiv.2407.06172).
- 682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701

## 702 A PROOF OF THEOREM IN §3.2

703  
704 Given that  $\sum_{i=1}^U X_i = 0$ , we want to attain sampling set  $\mathcal{S}$  that satisfies  $|\mathcal{S}| = V$  and:

$$706 \mathcal{S} = \underset{\mathcal{S}}{\operatorname{argmin}} |\sum_{i \in \mathcal{S}} X_i| = \underset{\mathcal{S}}{\operatorname{argmin}} (\sum_{i \in \mathcal{S}} X_i)^2 \quad (14)$$

707  
708 Firstly, it is easy to know that for any sampling set  $\mathcal{S}$ :

$$710 \mathbb{E}[\sum_{i \in \mathcal{S}} X_i] = 0 \quad (15)$$

711  
712 Thus,

$$713 \mathbb{E}[(\sum_{i \in \mathcal{S}} X_i)^2] = \mathbb{E}[(\sum_{i \in \mathcal{S}} X_i - 0)^2] = \mathbb{E}[(\sum_{i \in \mathcal{S}} X_i - \mathbb{E}[\sum_{i \in \mathcal{S}} X_i])^2] = \mathbb{E}[\operatorname{Var}[\sum_{i \in \mathcal{S}} X_i]] \quad (16)$$

714  
715 Considering that:

$$716 \operatorname{Var}[\sum_{i \in \mathcal{S}} X_i] = \sum_{i \in \operatorname{set}(\mathcal{S})} c_i^2 \operatorname{Var}(X_i) \quad (17)$$

717  
718 where  $c_i$  denotes the number of  $X_i$  in  $\mathcal{S}$ . On this basis, we derive that:

$$\begin{aligned} 719 \mathbb{E}[(\sum_{i \in \mathcal{S}} X_i)^2] &= \mathbb{E}[\operatorname{Var}(X)] \sum_{i \in \operatorname{set}(\mathcal{S})} c_i^2 \\ 720 &\geq \mathbb{E}[\operatorname{Var}(X)] (\sum_{i \in \operatorname{set}(\mathcal{S})} c_i)^2 |\operatorname{set}(\mathcal{S})|^{-1} \\ 721 &= V^2 \mathbb{E}[\operatorname{Var}(X)] |\operatorname{set}(\mathcal{S})|^{-1} \\ 722 &\geq V^2 \mathbb{E}[\operatorname{Var}(X)] \min(U, V)^{-1} \end{aligned} \quad (18)$$

723  
724 The equality condition of the first inequality is: the number of samples taken from each category is equal. The equality condition of the second inequality is: the number of sampled categories equals to  $\min(U, V)$ . These two conditions imply that a completely uniform sampling strategy is optimal.

## 725 B INTRODUCTION OF ELO RATING SYSTEM AND BRADLEY-TERRY MODEL

### 726 B.1 ELO RATING SYSTEM

727  
728 The Elo rating system (Elo & Sloan, 1978) is widely used to rank participants based on their relative performance in competitive settings. Given two models,  $A$  and  $B$ , with initial ratings  $R_A$  and  $R_B$ , the expected score of model  $A$  in a pairwise comparison is calculated as:

$$729 E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}}$$

730  
731 Similarly, the expected score for model  $B$  is:

$$732 E_B = \frac{1}{1 + 10^{(R_A - R_B)/400}}$$

733  
734 After the comparison, the actual results are used to update the ratings. If model  $A$  wins, its new rating  $R'_A$  is updated as:

$$735 R'_A = R_A + K(S_A - E_A)$$

736  
737 where  $S_A$  is the actual result of the match (1 for a win, 0 for a loss, and 0.5 for a draw), and  $K$  is a constant that controls the sensitivity of the rating adjustment. Model  $B$ 's rating is updated in a similar way:

$$738 R'_B = R_B + K(S_B - E_B)$$

739  
740 where  $S_B$  is the actual result for model  $B$ .

When extending the Elo rating system to multiple models, we consider a set of  $n$  models. Pairwise comparisons between the models are conducted, resulting in  $\binom{n}{2}$  unique pairs:

$$\binom{n}{2} = \frac{n(n-1)}{2}$$

Each pair is evaluated using the Elo score update rules, and the results are iteratively applied to adjust the ratings, ensuring that each model’s rating reflects its relative performance within the set.

The extension to multiple models leverages the transitive property. For any three models  $i, j, k \in \{1, 2, \dots, n\}$ , if  $R_i > R_j$  and  $R_j > R_k$ , the transitivity implies:

$$R_i > R_j \quad \text{and} \quad R_j > R_k \quad \implies \quad R_i > R_k$$

This property ensures consistency in the rankings, even when individual match outcomes vary. By iterating over all  $\binom{n}{2}$  comparisons, the Elo scores converge to reflect the overall capabilities of the models, with higher scores indicating stronger performance.

## B.2 BRADLEY-TERRY MODEL

The Bradley-Terry model (Bradley & Terry, 1952) estimates the probability that one model outperforms another in pairwise comparisons. For two models  $M_i$  and  $M_j$  with strength parameters  $\xi_i$  and  $\xi_j$ , the probability that model  $i$  beats model  $j$  is modeled as:

$$P(M_i > M_j) = \frac{1}{1 + e^{\xi_j - \xi_i}} \quad (19)$$

where  $\xi$  is an  $|M|$ -length vector of Bradley-Terry coefficients. Given a set of comparing results  $\mathcal{S} = \{(M_i^t, M_j^t, R^t)\}_{t=1}^T$  where  $R^t$  represents the degree  $M_i^t$  wins over  $M_j^t$ . We set  $\text{mean}(\xi) = 0$ . After attaining the BT scores using  $f^{pa}$ , we calculate the estimated win rate matrix with equation 19.

## C MORE EXPERIMENTAL ANALYSES

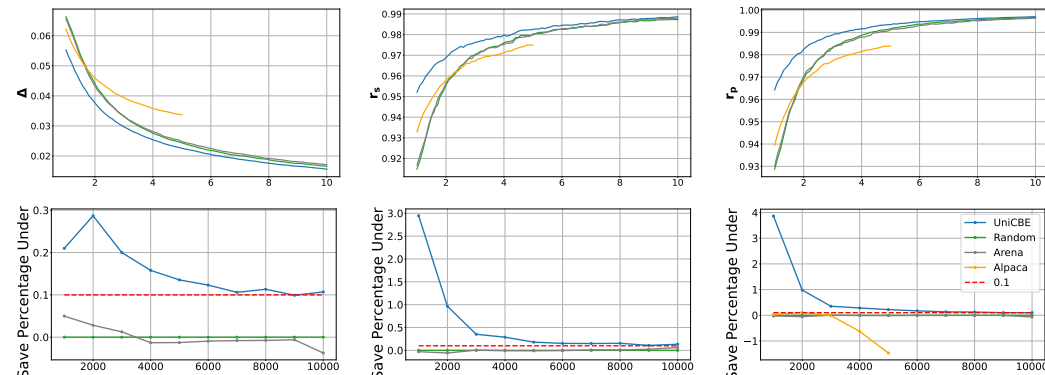


Figure 9: Results of compared CBE methods in a scenario where models and samples are dynamically added or removed at a random frequency.

### C.1 PERFORMANCE IN SCENARIOS CLOSE TO REALITY

We think that conducting experiments in settings that are closer to real-world scenarios (highly dynamic and requiring real-time evaluation) can help us more comprehensively assess UNICBE and the baseline methods. To this end, we perform the following experiments: Starting with a sample size of  $N = 600$  and model number of  $M = 12$ , we execute a random operation at each time step. The operations included: adding one model to be evaluated with a probability of 0.01, removing one model with a probability of 0.01, adding one potential sample with a probability of 0.01, randomly deleting one sample with a probability of 0.01, and taking no action with a probability of 0.96. Based on the experimental results shown in Figure 9, we have the following observations:

- The convergence speed of all baseline methods significantly slowed down compared to Figure 3. None of the baseline methods achieve a Spearman correlation coefficient of 0.96 or a Pearson correlation coefficient of 0.97 by  $T = 2000$ , highlighting the difficulty of model evaluation in this setting. In contrast, UNICBE achieve rapid convergence, reaching a Spearman coefficient of approximately 0.97 and a Pearson coefficient exceeding 0.98 by  $T = 2000$ .
- Over the long term, as  $T$  increases, UNICBE consistently demonstrates over 10% savings in preference budget across all metrics, even under this challenging setting, showcasing its strong practicality.
- An interesting observation is that ALPACAEVAL exhibits better convergence in the early stages compared to RANDOM and ARENA, supporting our previous conclusions in Table 1. However, as  $T$  increases, ALPACAEVAL’s lack of accuracy optimization objective leads to its performance being surpassed by RANDOM and ARENA.

## C.2 ABLATION STUDY OF UNIFORMITY CONSTRAINTS

Based on our analysis in §3, the degree to which uniformity is achieved is positively correlated with performance in terms of accuracy, convergence, and scalability. To explore the empirical relationship between the degree of uniformity constraints and the final outcomes, we draw inspiration from the concept of temperature-based control in random sampling. By adjusting the temperature  $T$  in the following formula for sampling  $f_T^{ts}$ , we regulate the extent of uniformity constraints according to  $P^l$  in equation 13:

$$f_T^{ts}(i, j, k) = \frac{(P_{i,j,k}^l)^{-T}}{\sum (P^l)^{-T}} \quad (20)$$

As  $T$  increases, the uniformity constraints become more relaxed. When  $T = 0$ , it corresponds to greedy sampling  $f_g^{ts}$ , which imposes the strictest uniformity constraints. When  $T = 1$ , it corresponds to probabilistic sampling  $f_p^{ts}$ , which imposes general uniformity constraints. When  $T = +\infty$ , it corresponds to random sampling, where no uniformity constraints are applied. Our experimental results are shown in Figure 10. As  $T$  increases from 0 to  $T = +\infty$ , the evaluation results progressively deteriorate. This indicates that adopting greedy sampling to impose the strictest uniformity constraints yields the optimal evaluation performance. This observation also validates the correctness of our conclusions in §3.

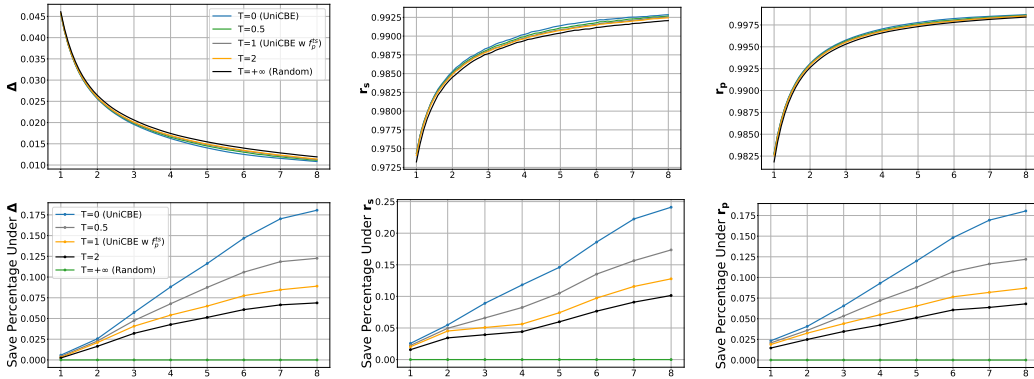


Figure 10: Results of UNICBE with different sampling temperatures.

## C.3 ADJUSTING THE WEIGHTS OF OPTIMIZATION OBJECTIVES

In equation 13, we integrate sampling matrices targeting different optimization objectives with equal weights. In practice, when faced with varying requirements, it is straightforward to prioritize a specific objective by adjusting the weights  $\theta_{acc}$ ,  $\theta_{con}$ , and  $\theta_{sca}$  for these matrices, as shown in equation 21.

$$P^l = \frac{(P^{acc-l})^{\theta_{acc}} \circ (P^{con-l})^{\theta_{con}} \circ (P^{sca-l})^{\theta_{sca}}}{\sum ((P^{acc-l})^{\theta_{acc}} \circ (P^{con-l})^{\theta_{con}} \circ (P^{sca-l})^{\theta_{sca}})} \quad (21)$$

As demonstrated in Table 3, we set different settings and calculate the degree of achievement  $\beta$  for each optimization objective following the procedure described in §E. Compared to equal-weight



integration, users can increase the corresponding  $\beta$  (e.g.,  $\beta_{acc}$ ) by assigning a larger weight to a specific optimization objective ( $\theta_{acc}$ ), thereby better meeting their practical needs (accuracy). We also observe that enhancing a specific optimization objective often comes with a slight decrease in the achievement of other objectives. In Figure 11, we illustrate an example of improving accuracy, where  $\theta_{acc}$  is increased from 1 to 2. We find that the increased focus on accuracy objective slightly slows down the convergence speed. As a result, when  $T$  is relatively small, the performance of  $\theta_{acc} = 2$  lags behind that of  $\theta_{acc} = 1$ . However, in the later stages, after convergence, the enhanced accuracy objective enables  $\theta_{acc} = 2$  to outperform  $\theta_{acc} = 1$ , resulting in greater savings in the preference budget.

Table 3: The measurement results of the achievement of objectives in §3 for UNICBE with varied hyperparameters.

|               | $\theta_{acc} = 2$ | $\theta_{acc} = 1$ | $\theta_{acc} = 1$ | $\theta_{acc} = 1$ |
|---------------|--------------------|--------------------|--------------------|--------------------|
| Settings      | $\theta_{con} = 1$ | $\theta_{con} = 2$ | $\theta_{con} = 1$ | $\theta_{con} = 1$ |
|               | $\theta_{sca} = 1$ | $\theta_{sca} = 1$ | $\theta_{sca} = 2$ | $\theta_{sca} = 1$ |
| $\beta_{acc}$ | .7380(+.0016)      | .7355(-.0009)      | .7351(-.0013)      | <b>.7364</b>       |
| $\beta_{con}$ | .9221(-.0007)      | .9235(+.0007)      | .9217(-.0011)      | <b>.9228</b>       |
| $\beta_{sca}$ | .9996(-.0001)      | .9997(.0000)       | .9998(+.0001)      | <b>.9997</b>       |

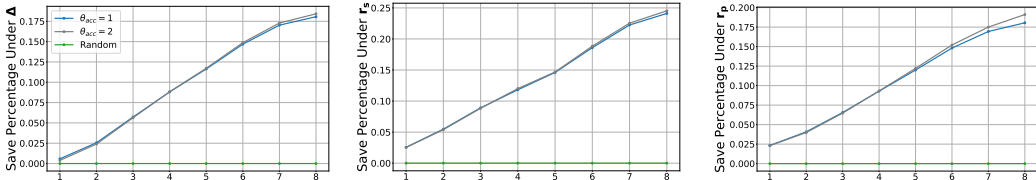


Figure 11: Results of UNICBE with different  $\theta_{acc}$ .

## D PROMPT FOR HAVING AN LLM ACT AS JUDGE

We follow AlpacaEval<sup>3</sup> to instruct the LLMs act as judge with the following prompt:

*You are a helpful assistant, that ranks models by the quality of their answers.*

*I want you to create a leaderboard of different of large-language models. To do so, I will give you the instructions (prompts) given to the models, and the responses of two models. Please give the winner model based on which responses would be preferred by humans. All inputs and outputs should be python dictionaries.*

*Here is the prompt:*

```
{
  "instruction": "{instruction}"
}
```

*Here are the outputs of the models:*

```
{
  "model": "model1",
  "answer": "{output1}"
},
{
  "model": "model2",
```

<sup>3</sup>[https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval)

918 "answer": ""{output<sub>2</sub>}"  
 919 }  
 920

921 *Now please give the winner model according to the quality of their an-*  
 922 *swers, so that the winner model has the best output. Then return the winner model*  
 923 *in the following format if model<sub>x</sub> is the winner: winner: model<sub>x</sub>*

924 *You need to strictly follow the format above. Please provide the ranking*  
 925 *that the majority of humans would give.*

926  
 927  
 928 **E THE CALCULATION PROCESS OF  $\beta$  IN TABLE 2**  
 929

930 We calculate the  $\beta$  values for each CBE method to measure how well they align with the optimization  
 931 objectives we analyzed in §3: ensuring uniformity across tuples, uniformity across model pairs in  
 932 win-rate uncertainty, and uniformity across models. Specifically, we first construct  $U^{acc1}$ ,  $U^{acc2}$ ,  
 933  $U^{con}$  and  $U^{sca}$  as follows:

934 
$$U_{i,j}^{acc1} = \begin{cases} 0, & \text{if } i = j \\ \frac{1}{M(M-1)}, & \text{else} \end{cases} \tag{22}$$

935 
$$U_{i,k}^{acc2} = \frac{1}{MN}$$

936  
 937 
$$U_{i,j}^{con} = \begin{cases} 0, & \text{if } i = j \\ \frac{1}{M(M-1)}, & \text{else} \end{cases} \tag{23}$$

938  
 939 
$$U_i^{sca} = \frac{1}{M} \tag{24}$$

940 On this basis, we calculate  $\beta_{acc}$ ,  $\beta_{con}$  and  $\beta_{sca}$  as follows:

941  
 942 
$$\beta_{acc} = \text{CosineSim}(U^{acc1}, \text{C.mean}(\text{dim} = -1)) \times \text{CosineSim}(U^{acc2}, \text{C.mean}(\text{dim} = 0)) \tag{25}$$

943  
 944 
$$\beta_{con} = \text{CosineSim}(U^{con}, \epsilon) \tag{26}$$

945  
 946 
$$\beta_{sca} = \text{CosineSim}(U^{sca}, \text{C.mean}(\text{dim} = -1).\text{mean}(\text{dim} = -1)) \tag{27}$$

947  
 948  
 949  
 950 **F FURTHER DISCUSSIONS**  
 951

952 **F.1 AFFINITY OF THE THREE OPTIMIZATION OBJECTIVES**  
 953

954 We have discussed in §3 that the keys to strengthen the accuracy, convergence and scalability of  
 955 CBE are: ensuring uniformity across tuples, uniformity across model pairs in win-rate uncertainty,  
 956 and uniformity across models. Below we discuss their compatibility. Firstly, ensuring uniformity  
 957 across models can be seen as a sub-goal of ensuring uniformity across tuples, therefore they exhibit  
 958 a strong affinity. The former can be considered a refinement of the latter, specifically focusing on  
 959 model uniformity. Moreover, as shown inequation 8, the uncertainty in win-rate between models  
 960  $\epsilon_{i,j}$  is inversely proportional to the number of comparisons between models. Therefore, improv-  
 961 ing uniformity across model pairs in win-rate uncertainty  $\epsilon$  will also contribute to a more uniform  
 962 distribution of comparisons between models. This further implies that ensuring uniformity across  
 963 model pairs in win-rate uncertainty and ensuring uniformity across tuples are compatible goals with  
 964 a strong affinity. Furthermore, as shown in Table 2, all  $\beta$  values of UNICBE are improved com-  
 965 pared to the baselines, experimentally validating the fact that optimizing the three objectives can  
 966 be mutually beneficial. Based on this analysis, we can infer that changing  $\alpha$  essentially adjusts  
 967 the optimization emphasis on different objectives. However, since the three objectives are mutually  
 968 reinforcing, the effect of changing  $\alpha$  will be relatively small.

969 **F.2 DISCUSSION ON SAMPLING BIAS IN INCOMPLETE SAMPLING SCENARIOS**  
 970

971 Previous studies have discussed the risks of introducing sampling bias in incomplete sampling sce-  
 narios. Specifically, Vabalas et al. (2019) demonstrated through simulation experiments that K-fold

cross-validation (K-fold CV) can produce significant performance estimation bias when dealing with small sample sizes. This bias persists even when the sample size reaches 1000. In contrast, methods like nested cross-validation (Nested CV) and train/test split have been shown to provide robust and unbiased performance estimates regardless of sample size. Kossen et al. (2021) introduced a weighting scheme, as described in (Farquhar et al., 2021), to mitigate sampling bias in active testing scenarios. Vivek et al. (2024) proposed leveraging information obtained from source models to select representative samples from the test set, thereby reducing sampling bias. Additionally, Polo et al. (2024) employed Item Response Theory (Lord & Novick, 2008) to correct sample bias in addressing this issue.

These studies inspired us to investigate the bias problem in the CBE scenario. Unlike the aforementioned studies, we found that in CBE scenario, not only does sample bias exist, but model bias also plays a role, and the two are coupled. This coupling poses greater challenges for analyzing and mitigating these biases. To address this, based on the analyses outlined in §3, we propose the UNICBE method, which effectively alleviates biases in this scenario.

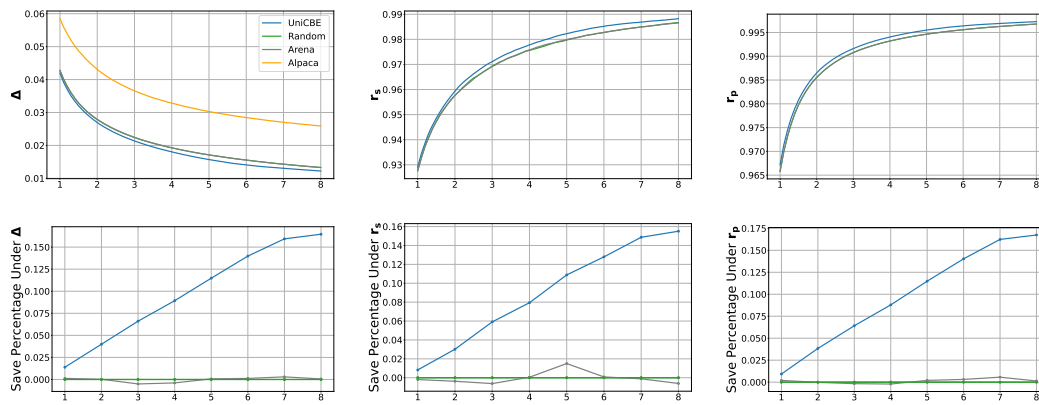


Figure 12: Results of compared CBE methods with GPT-3.5-turbo as the judge on AlpacaEval.

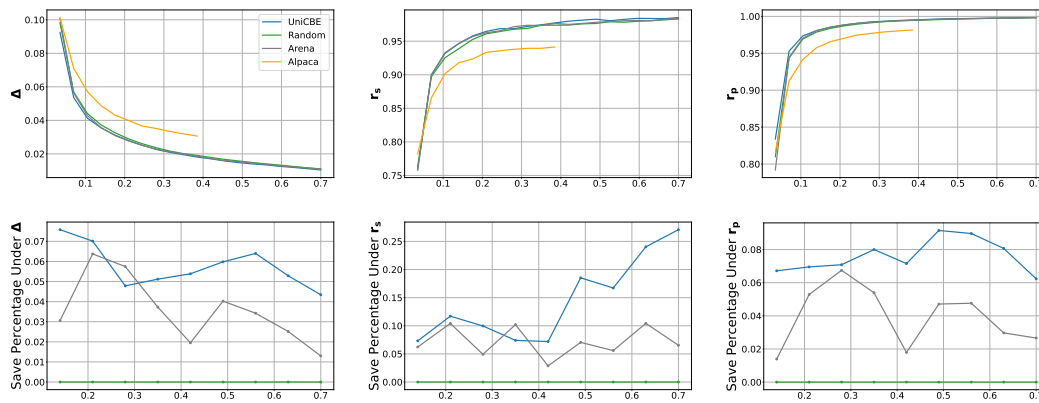


Figure 13: Results of compared CBE methods with human as the judge on MT-Bench.

1026  
 1027  
 1028  
 1029  
 1030  
 1031  
 1032  
 1033  
 1034  
 1035  
 1036  
 1037  
 1038  
 1039  
 1040  
 1041  
 1042  
 1043  
 1044  
 1045  
 1046  
 1047  
 1048  
 1049  
 1050  
 1051  
 1052  
 1053  
 1054  
 1055  
 1056  
 1057  
 1058  
 1059  
 1060  
 1061  
 1062  
 1063  
 1064  
 1065  
 1066  
 1067  
 1068  
 1069  
 1070  
 1071  
 1072  
 1073  
 1074  
 1075  
 1076  
 1077  
 1078  
 1079

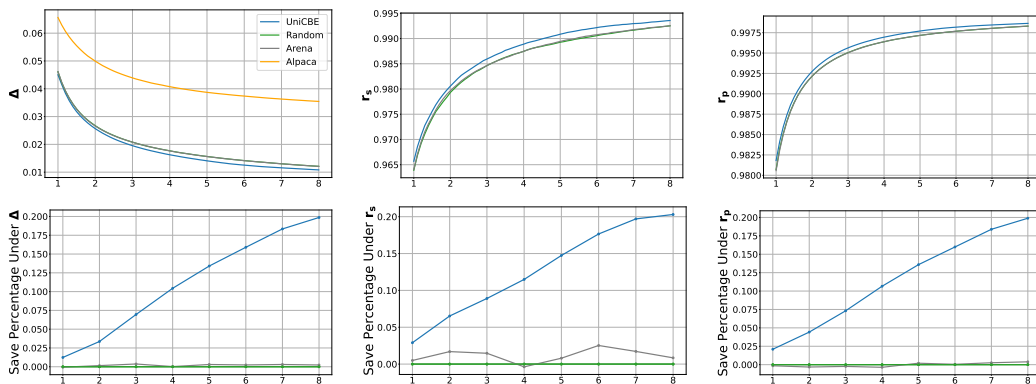


Figure 14: Results of compared CBE methods with Qwen-Plus as the judge on AlpacaEval.