

# 1 RESPONSE TO REVIEWER 8LCX

Dear Reviewer 8LCX, we sincerely thank you for your valuable feedback on our submission. Below is our responses to the concerns you raised. We have incorporated the following contents into the updated version of our paper, which we believe will help enhance the quality of our submission.

**Q1: The novelty of balancing accuracy, convergence, and scalability needs further justification, as similar uniform sampling strategies have been discussed in prior works that highlight the uniformity, such as Vabalas et al. (2019) for sampling biases, which could diminish its uniqueness.**

**A1:** We agree with your suggestion to discuss related work concerning sampling uniformity in connection with our work. Here is our discussion:

Previous studies have discussed the risks of introducing sampling bias in incomplete sampling scenarios. 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 Section 3, we propose the UNICBE method, which effectively alleviates biases in this scenario.

**Q2: Although the experiment of MT-Bench is based on human evaluator, larger portion of the evaluation is relied on AlpacaEval, as larger number of models and samples are used for the evaluation with AlpacaEval. The reliance on GPT-4 and GPT-3.5-turbo as evaluators, while useful, could benefit from validation against human judgments or additional LLMs, such as Claude, to establish greater reliability and generalizability across evaluator types.**

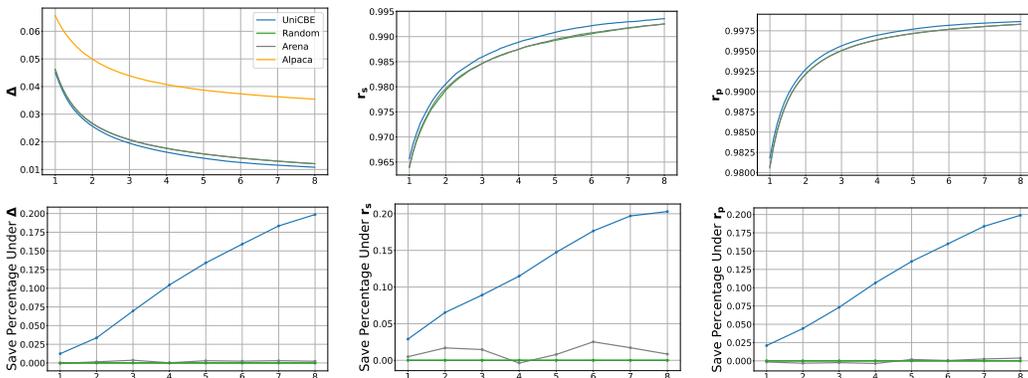


Figure 1: Results of compared CBE methods with Qwen-Plus as the judge on AlpacaEval.

**A2:** Yes, in our experiments, we test the performance of UNICBE with humans, GPT-4o, and GPT-3.5-turbo as judges. We recognize that involving a broader range of evaluators helps enhance the robustness and generalizability of our experimental conclusions. Therefore, below we additionally test the compared methods using Qwen-Plus (Yang et al., 2024) as the judge. As shown in Figure 1,

when Qwen-Plus is used as the judge, the experimental results are similar to those obtained with other types of judges. UNICBE achieves significant preference budget savings, exceeding 20%. This experimental result further validates the generalizability of UNICBE to different sources of preference signals.

**Q3: Minor details, but the readability of all figures could be enhanced by widening the lines in each plot, which would improve clarity and interpretation for readers.**

**A3:** Thank you for your valuable suggestions. We have increased the width of the lines in the figure in the revised version to enhance readability. You can refer to the latest version.

**Q4: As the UNICBE is based on three matrix , each targeting different goal of accuracy, convergence, scaliability, can user steer between those by adding hyperparameter for each matrix? Would it be also possible to quantify it through experiment?**

**A4:** We agree with your suggestion to add hyperparameters for each matrix to achieve controllability for different optimization objectives. In the original manuscript, we integrate sampling matrices targeting different optimization objectives with equal weights:

$$P^l = \frac{P^{acc-l} \circ P^{con-l} \circ P^{sca-l}}{\sum(P^{acc-l} \circ P^{con-l} \circ P^{sca-l})} \quad (1)$$

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 2.

$$P^l = \frac{(P^{acc-l})^{\theta_{acc}} \circ (P^{con-l})^{\theta_{con}} \circ (P^{sca-l})^{\theta_{sca}}}{\sum((P^{acc-l})^{\theta_{acc}} \circ (P^{con-l})^{\theta_{con}} \circ (P^{sca-l})^{\theta_{sca}})} \quad (2)$$

As demonstrated in Table 1, we set different settings and calculate the degree of achievement level for each optimization objective  $\beta$  following the calculation procedure described in Appendix-E. Compared to equal-weight integration, users can easily 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 2, 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 1: The measurement results of the achievement of objectives in Section 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>

**Q5: While scalability is addressed by sequentially adding models, the paper could enhance this section by incorporating real-world scenarios, where models enter and exit dynamically, further proving UNICBE’s robustness in evolving benchmarks.**

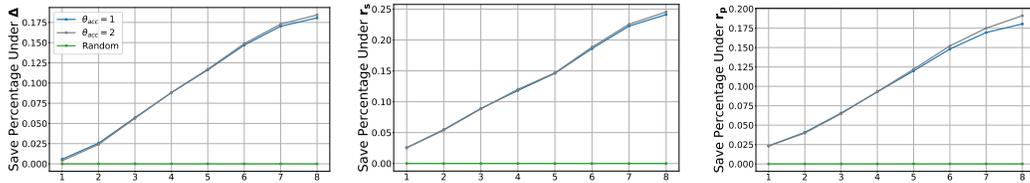


Figure 2: Results of UNICBE with different  $\theta_{acc}$ .

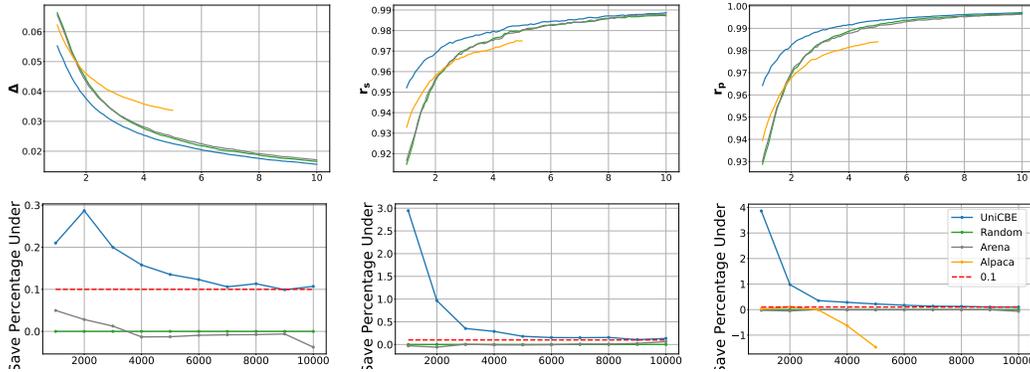


Figure 3: Results of compared CBE methods in a scenario where models and samples are dynamically added or removed at a random frequency.

**A5:** We agree with your suggestion that testing UNICBE in a highly dynamic, real-time evaluation setting can help us more comprehensively assess its performance. To this end, we conduct 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 3, we have the following observations:

- The convergence speed of all baseline methods significantly slowed down. 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 (original manuscript). However, as  $T$  increases, ALPACAEVAL’s lack of accuracy optimization objective leads to its performance being surpassed by RANDOM and ARENA.

**Q6:** The given choice of greedy sampling over probabilistic sampling and Bradley-Terry model over Elo rating system appears significant to the framework’s success. Could the authors conduct a small experiment to demonstrate that UNICBE maintains its effectiveness across different sampling and aggregation settings?

**A6:** We appreciate your valuable suggestion. In fact, as shown in Figure 4 (Figure 5 in the original manuscript), we explore the combination of UNICBE with probability sampling strategy, Elo rating and average win rate aggregation strategies, comparing these with the default configuration. Our findings are as follows:

