Self-supervised Attribute-aware Dynamic Preference Ranking Alignment

Anonymous ACL submission

Abstract

Reinforcement Learning from Human Feedback and its variants excel in aligning with human intentions to generate helpful, harmless, and honest responses. However, most of them rely on costly human-annotated pairwise comparisons for supervised alignment, which is not suitable for list-level scenarios, such as community question answering. Additionally, human preferences are influenced by multiple intrinsic factors in responses, leading to decision-making inconsistencies. Therefore, we propose **Se**lf-supervised **A**ttribute-aware **d**ynamic **p**reference **ra**nking, called SeAdpraİt quantifies preference differences between responses based on Attribute-Perceptual Distance Factors (APDF) and dynamically determines the list-wise alignment order. Furthermore, it achieves fine-grained preference difference learning and enables precise alignment with the optimal one. We specifically constructed a challenging code preference dataset named StaCoCoQA, and introduced more cost-effective and scalable preference evaluation metrics: PrefHit and PrefRecall. Extensive experimental results show that SeAdpra exhibits superior performance and generalizability on both StaCoCoQA and preference datasets from eight popular domains[1].

1 Introduction

Community Question Answering (CoQA) (Romeo et al., 2018; Wu et al., 2018) seeks to generate responses that are semantically accurate and match the preferences of community members. Currently, Reinforcement Learning from Human (or AI) Feedback (RLHF/RLAIF) (Christiano et al., 2017; Bai et al., 2022) has enabled precise control of large language models (LLMs) for generating human-like responses (Stiennon et al., 2020; Ouyang et al., 2022). However, applying them to CoQA remains underexplored. Moreover, human preferences do not always follow a singular, value-based hierarchy. Various factors can influence decision-making and may exhibit inconsistencies (Tversky, 1969; Yang et al., 2025), which undoubtedly presents a challenge for aligning LLMs with CoQA.

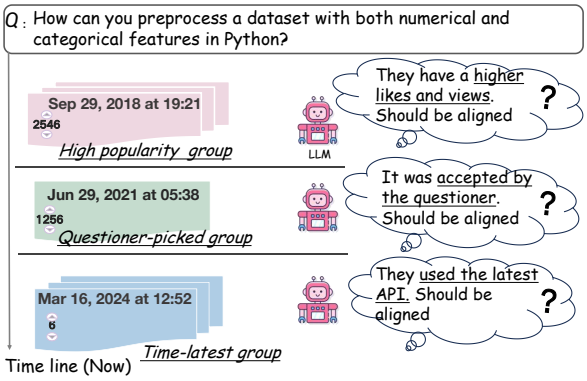


Figure 1: Which response should the LLMs align with? In the code community, each response has different attributes such as semantics, popularity, and timeliness, leading to potentially different optimal responses.

Existing methods are limited to pairwise comparison (one chosen and one rejected), such as reward model-based RLHF (Ouyang et al., 2022), offline supervised Direct Preference Optimization (DPO) (Rafailov et al., 2024), as well as other variants like SLiC (Zhao et al., 2023) and pseudo-list RRHF (Yuan et al., 2024) that adopt pairwise hinge loss. However, a real-world prompt may have multiple high-quality responses (Cui et al., 2023). For example, in the coding community, the optimal one may vary with thier different attributes, such as semantics, popularity, and timeliness, as illustrated in Figure 1. Recently, some alignment methods have attempted to rank multiple preferred candidates. PRO (Song et al., 2024) introduces a list-level maximum likelihood estimation loss to shift towards preference ranking but overlooks the attributes of responses. LiPO (Liu et al., 2024) directly optimizes list-based ranking preferences and begins to address response labels, but has not yet addressed

[1]Our dataset and project codes are accessible https://anonymous.4open.science/r/SeAdpra-D8E0

the integration of multiple labels. Moreover, these supervised learning methods depend on human or AI annotations of preference pairs or lists to specify the best responses for alignment. However, preference data are relatively scarce and expensive to collect in practice (Casper et al., 2023).

To overcome the above challenges, we propose three-stage SeAdpra, a **Se**lf-supervised **a**ttribute-aware **d**ynamic **p**reference **ra**nking framework. 1) First, the Multi-Attribute Perception quantifies preference-level differences through Attribute-Perceptual Distance Factors (APDF), enabling the integration of multiple attributes for self-supervised dynamic ranking. 2) Second, the Perception Alignment can quickly adapt to domain knowledge by precisely aligning with the optimal. 3) Third, the Perceptual Comparison performs multi-turn fine-grained list-wise preference difference contrastive learning. In each round, it maximizes the reward for the optimal and minimizes the penalty for the remaining based on self-generated preference ranks.

For enhancing the cost-efficiency and domain applicability of the preference evaluation scheme, we propose new metrics that follow the 'CSTC' criterion (details in Appendix A.2), as an alternative to the costly win rate (Dudík et al., 2015), namely PrefHit and PrefRecall. They can accommodate the expansion of benchmarks. Aiming to validate the effectiveness of SeAdpra in specific domains, we have constructed a programming CoQA preference dataset, called StaCoCoQA, which contains over 60,738 programming directories and 9,978,474 entries. Our main contributions are as follows:

- We introduce the Attribute Perceptual Distance Factor (APDF) to gauge the in preference-level gaps of multiple responses, replacing the binary judgment of preferred versus non-preferred. We propose an self-supervised dynamic preference ranking framework that achieves label-free list-wise preference alignment.
- We present the StaCoCoQA, a large-scale, high-quality, real-time (as of May 2024) dataset for preference alignment in programming CoQA, and develop two new alignment metrics abided by the 'CSTC' criterion.
- We conducted extensive experiments on eight hot public datasets and StaCoCoQA, providing a reference benchmark. The experimental results demonstrate that SeAdpra excels in alignment while maintaining safety.

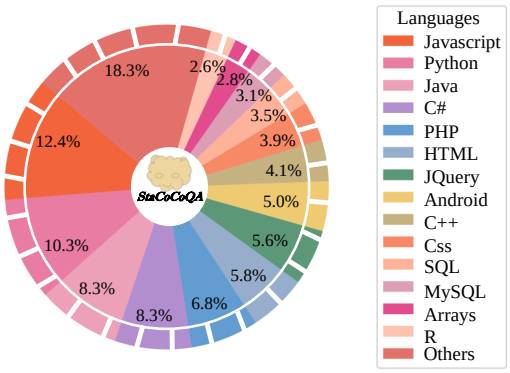


Figure 2: Showcasing the top-15 primary programming language categories in StaCoCoQA.

2 Method

2.1 Problem Definition

Our goal is to align an LLM with user preferences in CoQA using our Unsupervised Attribute-aware Dynamic Preference Ranking strategy. The training dataset is denoted as $\mathcal{D} = \{Q^i, R^i\}_{i=1}^N$. For a given question Q, it corresponds to a series of responses $R = \{R_1, \ldots, R_M\}$, where each response $R_i = (C, A)$, with C representing the content and A representing the scalable attributes. The size L of the scalable attribute $A = \{A_1, \ldots, A_L\}$ is determined by community characteristics. For example, in the code community, $L = 3$ and $A = \{S, P, T\}$. Here, S represents the semantic similarity between C and Q; P represents the popularity of R, and T represents the creation time of each response.

2.2 Multi-attribute Perception

2.2.1 Attribute-Perceptual Distance Factor

The existing alignment optimization objectives (Rafailov et al., 2024; Song et al., 2024) do not take into account the attributes of the candidates, which can differentiate their preferences. Therefore, there is a need to explore optimization methods that can effectively incorporate these attributes. In this context, LambdARank (Wang et al., 2018; Jagerman et al., 2022) introduces Lambda weights λ_{ij}, which scale the gradient of each pair of scores based on the labels of the pairs to optimize a metric-driven loss function and effectively incorporating label information into the optimization process.

Inspired by the λ_{ij}, the Attribute-Perceptual Distance Factor $\delta_{i,j}$ is designed to quantify the preference difference between two candidates i and j in the optimization objective. It not only considers the positional relationship of candidates in prefer-

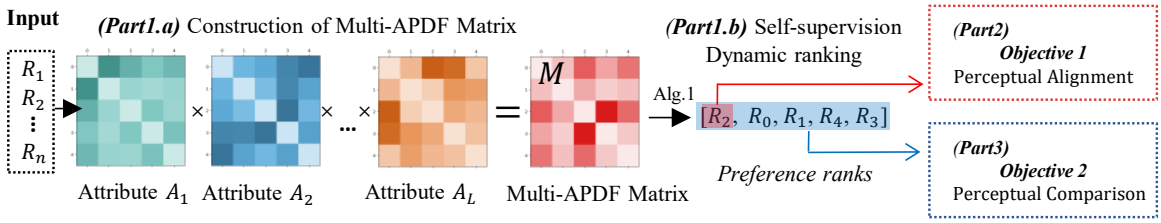


Figure 3: The overall framework of SeAdpra , which includes: (Part1) Multi-attribute Perception for quantifying preference, containing the Construction of Multi-APDF Matrix and Self-supervised dynamic ranking; (Part2) Perceptual Alignment for aligning the optimal ranks objective; (Part3) Perceptual Comparison on all candidates for learning on-chain preference difference.

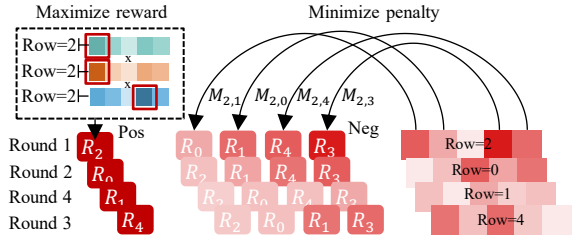


Figure 4: Implementation Workflow of Perceptual Comparison. In each round, the reward of the current positive is maximized, and the penalty for the remaining negative is minimized sequentially.

ence ranks but also incorporates their label values through the gain function, and expressed as:

$$\delta_{i,j} = (G(i) - G(j)) \times (T(i) - T(j)) \quad (1)$$

$$T(i) = 1/log(l_i + 1) \quad (2)$$

where l_i and l_j are the ranking positions of response i and j, respectively. The gain function $G(\cdot)$ varies with different intrinsic attributes.

2.2.2 Construction of the Multi-APDF Matrix

Given the response $R = \{R_1, \ldots, R_M\}$ to question Q, the construction of the Multi-APDF matrix is a dot-product fusion of L Single-APDF matrix. Based on the characteristics of the code community shown in Figure 1, the main attributes that influence user preferences are semantics (text content), popularity, and creation time.

Semantic-APDF matrix $\Delta_{Se} = \{\delta_{Se_{ij}} | i, j \in M\}$, we define $G^{Se}(i) = 2^{\varphi(i)-1}$, where $\varphi(i) = \cos(E_Q, E_{C_i})$. Here, $E_Q \in \mathbb{R}^{q \times d}$ and $E_{C_i} \in \mathbb{R}^{r \times d}$ represent the semantic vectors of the question Q and the text content C_i of response R_i, encoded by prompt-based LLMs (BehnamGhader et al., 2024). Here, q is the length of the question, r is the length of the text content, and d is the dimension of the LLM's embedding space.

Popularity-APDF matrix $\Delta_{Po} = \{\delta_{Po_{ij}} | i, j \in M\}$, to mitigate the bias caused by the accumulation of popularity over time, we apply time decay to P based on T, denoted as $\tilde{P}$. To avoid bias caused by extreme values and excessive numerical differences, we set $G^{Po}(i) = \lg(\tilde{P}_i + 1)$.

Multi-APDF matrix on the scalable attribute $A = \{A_1, \ldots, A_L\}$ is represented generally as:

$$\Delta_M = \prod_{k=1}^{L} \Delta_{A_k} \quad (3)$$

where Δ_{A_k} is the APDF matrix corresponding to attribute A_k. Similarly, The code Multi-APDF matrix $\Delta_M^{code} \in \mathbb{R}^{M \times M}$ is represented as follows:

$$\Delta_M^{code} = \Delta_{Se} \cdot \Delta_{Po} \quad (4)$$

2.2.3 Self-supervision Dynamic Ranking

To avoid relying on manually labeled alignment targets, we propose the Self-supervised Dynamic Ranking based on the Multi-APDF Matrix. It iteratively selects the most significant pair-wise distance (Multi-APDF δ_M) and ranks the candidates according to the semantic ranks, which ensures that the ranking not only reflects pair-wise perceptual differences but also adheres to semantic priorities. Its implementation details are provided in the Algorithm 1. The D^R represents the set of candidates' positions after dynamic ranking:

$$D^R = \{i_1, i_2, \ldots, i_M\} \quad (5)$$

2.3 Perceptual Alignment

Since the most effective learning for domain knowledge method is SFT (Stiennon et al., 2020), and the most direct one in alignment is also to perform SFT on a high-quality preference dataset (Rafailov et al., 2024), we align the optimal response by treating the first response in dynamic ranking as the target for SFT for the question Q. The first optimization objective is represented as follows:

$$L_{Pa} = -\frac{1}{|R_b|} \sum_{j=1}^{|R_b|} \log P(R_b(j) | Q, R_b(< j)) \quad (6)$$

where $R_{D^R(0)}$ denotes as R_b. The $D^R(i)$ is the i-th element, and $R_b(j)$ is the j-th token.

2.4 Perceptual Comparison

In terms of many list-wise loss functions, the softmax cross-entropy loss in ListNet (Cao et al., 2007) uses double summation to emphasize comparisons between different samples, making it suitable for ranking loss. Therefore, we adopt it as the basis for the second optimization objective and conduct a total of $M-1$ iterative comparisons. To deepen the impact of preference differences, for each iteration, we maximize the reward for positive and minimize the penalty for remains negative sequentially.

Maximizing the reward is achieved by finding all maximum value in the mapped row of the alignment target in all Single-APDF matrix, and then multiplying the values together. For the m-th comparison, it is represented as follows:

$$W_m^r = \prod_{k=1}^{L} max(\Delta_{A_k}(D^R(m), \cdot)) \quad (7)$$

where $\Delta_{A_k}(i, j)$ refers to the element at the i-th row and j-th column of Δ_{A_k}, and $\cdot$ represents all elements in the row or column.

Minimizing the penalty involves differentiating the penalty strengths based on preference levels, where a slight penalty is applied to $R_{D_R(i)}$ and a stronger penalty is applied to $R_{D_R(i+1)}$. This approach contrasts with the existing method, which applies the same penalty to all negative examples, and ensures that the penalty for responses ranked higher in the self-supervised ranking D_R is minimized. For the negative R_i, its penalty is represented as follows:

$$W_i^p = sort(\Delta_M(D^R(m), \cdot))(i) \quad (8)$$

where $sort(\cdot)$ is the function that sorts in an ascending order. $\Delta_M(i, j)$ is the i-th row and the j-th APDF in the Multi-APDF matrix.

To achieve on-chain ranking and fine-grained distinction among all responses, unlike traditional optimization methods that sequentially remove the optimal response, all responses participate in each iteration. Moreover, the corresponding penalties or rewards for the responses change throughout the iterations. The second optimization objective is represented as:

$$L_{Pc} = -\sum_{m=1}^{M-1} \log \left(\frac{\tau_r(b)}{\sum_{i \neq b}^{M} \tau_p(i) + \tau_r(b)} \right) \quad (9)$$

$$\tau_r(b) = \exp(\pi_s(Q, R_b)) * W_m^r \quad (10)$$

$$\tau_p(i) = \exp(\pi_s(Q, R_i)) * W_i^p \quad (11)$$

$$\pi_s(Q, R_i) = \frac{1}{t} \sum_{k=1}^{t} \log P(r_k | Q, r_{<k}) \quad (12)$$

Here, $D^R(m)$ denotes as b. the $\pi_s(\cdot)$ represents a policy network that replaces the reward in RLHF with language modeling logits. The labeled response R, composed of t tokens, is denoted as $R_i = \{r_1, \ldots, r_t\}$. Finally, SeAdpra enables LLMs to be trained by the following objective:

$$Loss = L_{Pc} + \alpha L_{Pa} \quad (13)$$

To avoid overfitting the initial best response, α will control the balance between it and the remaining preferences, thereby ensuring text quality.

3 Experiments

3.1 Dataset

Due to the additional challenges that programming QA presents for LLMs and the lack of high-quality, authentic multi-answer code preference datasets, we turned to StackExchange [2], a platform with forums that are accompanied by rich question-answering metadata. Based on this, we constructed a large-scale programming QA dataset in real-time (as of May 2024), called StaCoCoQA. It contains over 60,738 programming directories, as shown in Table 8, and 9,978,474 entries, with partial data statistics displayed in Figure 2. The data format of StaCoCoQA is presented in Table 11.

The initial dataset D_I contains 24,101,803 entries, and is processed by the following steps: (1) Select entries with "Questioner-picked answer" pairs to represent the preferences of the questioners, resulting in 12,260,106 entries in the D_Q. (2) Select data where the question includes at least one code block to focus on specific-domain programming QA, resulting in 9,978,474 entries in the dataset D_C. (3) All HTML tags were cleaned using BeautifulSoup [3] to ensure that the model is not affected by overly complex and meaningless content. (4) Control the quality of the dataset by considering factors such as the time the question was posted, the size of the response pool, the difference between the highest and lowest votes within a pool, the votes for each response, the token-level length of the question and the answers, which yields varying sizes: 3K, 8K, 18K, 29K, and 64K. The controlled creation time variable and the data details after each processing step are shown in Table 7.

[2] https://archive.org/details/stackexchange

[3] https://beautiful-soup-4.readthedocs.io/en/latest/

General LLMs	Preference PrefHit	Preference PrefRecall	Preference Reward	Accuracy BLEU	Supervised Alignment	Preference PrefHit	Preference PrefRecall	Preference Reward	Accuracy BLEU
GPT-J	0.2572	0.6268	0.2410	0.0923	Llama2-7B	0.2029	0.803	0.0933	0.0947
Pythia-2.8B	0.3370	0.6449	0.1716	0.1355	SFT	0.2428	0.8125	0.1738	0.1364
Qwen2-7B	0.2790	0.8179	0.1593	0.2530	Slic	0.2464	0.6171	0.1700	0.1400
Qwen2-57B	0.3086	0.6481	0.6854	0.2568	RRHF	0.3297	0.8234	0.2263	0.1504
Qwen2-72B	0.3212	0.5555	0.6901	0.2286	DPO-BT	0.2500	0.8125	0.1728	0.1363
StarCoder2-15B	0.2464	0.6292	0.2962	0.1159	DPO-PT	0.2572	0.8067	0.1700	0.1348
ChatGLM4-9B	0.2246	0.6099	0.1686	0.1529	PRO	0.3025	0.6605	0.1802	0.1197
Llama3-8B	0.2826	0.6425	0.2458	0.1723	**SeAdpra***	**0.3659**	**0.8279**	**0.2301**	**0.1412**

Table 1: Main results on the StaCoCoQA. The left shows the performance of general LLMs, while the right presents the performance of the fine-tuned Llama2-7B across various strong benchmarks for preference alignment. Our method SeAdpra is highlighted in **bold**.

To further validate the effectiveness of SeAdpra, we also select eight popular topic CoQA datasets[4], which have been filtered to meet specific criteria for preference models (Askell et al., 2021). Their detailed data information is provided in Table 6.

3.2 Evaluation Metrics

For preference evaluation, we design PrefHit and PrefRecall, adhering to the "CSTC" criterion outlined in Appendix A.2, which overcome the limitations of existing evaluation methods, as detailed in Appendix A.1. In addition, we demonstrate the effectiveness of thees new evaluation from two main aspects: 1) consistency with traditional metrics, and 2) applicability in different application scenarios in Appendix A.4. Following the previous (Song et al., 2024), we also employ a professional reward.

For accuracy evaluation, we alternately employ BLEU (Papineni et al., 2002), RougeL (Lin, 2004), and CoSim. Similar to codebertscore (Zhou et al., 2023), CoSim not only focuses on the semantics of the code but also considers structural matching. Additionally, the implementation details of SeAdpra are described in detail in the Appendix E.1.

3.3 Baseline

Following the DPO (Rafailov et al., 2024), we evaluated several existing approaches aligned with human preference, including GPT-J (Wang and Komatsuzaki, 2021) and Pythia-2.8B (Biderman et al., 2023). Next, we assessed StarCoder2 (Lozhkov et al., 2024), which has demonstrated strong performance in code generation, alongside several general-purpose LLMs: Qwen2 (Yang et al., 2024), ChatGLM4 (Wang et al., 2023; GLM et al., 2024) and Llama serials (Touvron et al., 2023; AI@Meta, 2024). Finally, we fine-tuned Llama2-7B on the StaCoCoQA and compared its performance with other strong baselines for supervised learning in preference alignment, including SFT, RRHF (Yuan et al., 2024), Silc (Zhao et al., 2023), DPO, and PRO (Song et al., 2024).

3.4 Main Results

We compared the performance of SeAdprawith general LLMs and strong preference alignment benchmarks on the StaCoCoQA dataset, as shown in Table 1. Additionally, we compared SeAdpra with the strongly supervised alignment model PRO (Song et al., 2024) on eight publicly available CoQA datasets, as presented in Table 2 and Figure 8.

Larger Model Parameters, Higher Preference. Firstly, the Qwen2 series has adopted DPO (Rafailov et al., 2024) in post-training, resulting in a significant enhancement in Reward. In a horizontal comparison, the performance of Qwen2-7B and Llama2-7B in terms of PrefHit is comparable. Gradually increasing the parameter size of Qwen2 (Yang et al., 2024) and Llama leads to higher PrefHit and Reward. Additionally, general LLMs continue to demonstrate strong capabilities of programming understanding and generation preference datasets, contributing to high BLEU scores. These findings indicate that increasing parameter size can significantly improve alignment.

List-wise Ranking Outperforms Pair-wise Comparison. Intuitively, list-wise DPO-PT surpasses pair-wise DPO-BT on PrefHit. Other list-wise methods, such as RRHF, PRO, and our SeAdpra, also undoubtedly surpass the pair-wise Slic.

Both Parameter Size and Alignment Strategies are Effective. Compared to other models,

[4]https://huggingface.co/datasets/HuggingFaceH4/stack-exchange-preferences

Dataset	Model	Preference		Acc	
		PrefHit	*PrefRec*	*Reward*	*Rouge*
Academia	PRO	33.78	59.56	69.94	9.84
	Ours	36.44	60.89	70.17	10.69
Chemistry	PRO	36.31	63.39	69.15	11.16
	Ours	38.69	64.68	69.31	12.27
Cooking	PRO	35.29	58.32	69.87	12.13
	Ours	38.50	60.01	69.93	13.73
Math	PRO	30.00	56.50	69.06	13.50
	Ours	32.00	58.54	69.21	14.45
Music	PRO	34.33	60.22	70.29	13.05
	Ours	37.00	60.61	70.84	13.82
Politics	PRO	41.77	66.10	69.52	9.31
	Ours	42.19	66.03	69.74	9.38
Code	PRO	26.00	51.13	69.17	12.44
	Ours	27.00	51.77	69.46	13.33
Security	PRO	23.62	49.23	70.13	10.63
	Ours	25.20	49.24	70.92	10.98
Mean	PRO	32.64	58.05	69.64	11.51
	Ours	**34.25**	**58.98**	**69.88**	**12.33**

Table 2: Main results (%) on eight publicly available and popular CoQA datasets, comparing the strong listwise benchmark PRO and **ours with bold**.

Method	Preference (↑)		Accuracy (↑)			
	PrefHit	PrefRec	Reward	CoSim	BLEU	Rouge
SeAdpra	**34.8**	**82.5**	**22.3**	**69.1**	**17.4**	**21.8**
-w/o PerAl	30.4	83.0	18.7	68.8	12.6	21.0
-w/o PerCo	32.6	82.3	24.2	69.3	16.4	21.0
-w/o Δ_{Se}	31.2	82.8	18.6	68.3	12.4	20.9
-w/o Δ_{Po}	29.4	82.2	22.1	69.0	16.6	21.4
$PerCo_{Se}$	30.9	83.5	15.6	67.6	9.9	19.6
$PerCo_{Po}$	30.3	82.7	20.5	68.9	14.4	20.1

Table 3: Ablation Results (%). $PerCo_{Se}$ or $PerCo_{Po}$ only employs Single-APDF in Perceptual Comparison, replacing Δ_M with Δ_{Se} or Δ_{Po}. The bold represents the overall effect. The underlining highlights the most significant metric for each component's impact.

Pythia-2.8B achieved impressive results with significantly fewer parameters . Effective alignment strategies can balance the performance differences brought by parameter size. For example, Llama2-7B with PRO achieves results close to Qwen2-57B in PrefHit. Moreover, Llama2-7B combined with our method SeAdpra has already far exceeded the PrefHit of Qwen2-57B.

Rather not Higher Reward, Higher PrefHit. It is evident that Reward and PrefHit are not always positively correlated, indicating that models do not always accurately learn human preferences and cannot fully replace real human evaluation. Therefore, relying solely on a single public reward model is not sufficiently comprehensive when assessing preference alignment.

3.5 Ablation Study

In this section, we discuss the effectiveness of each component of SeAdpra and its impact on various metrics. The results are presented in Table 3.

Perceptual Comparison aims to prevent the model from relying solely on linguistic probability ordering while neglecting the significance of APDF. Removing this Reward will significantly increase the margin, but PrefHit will decrease, which may hinder the model's ability to compare and learn the preference differences between responses.

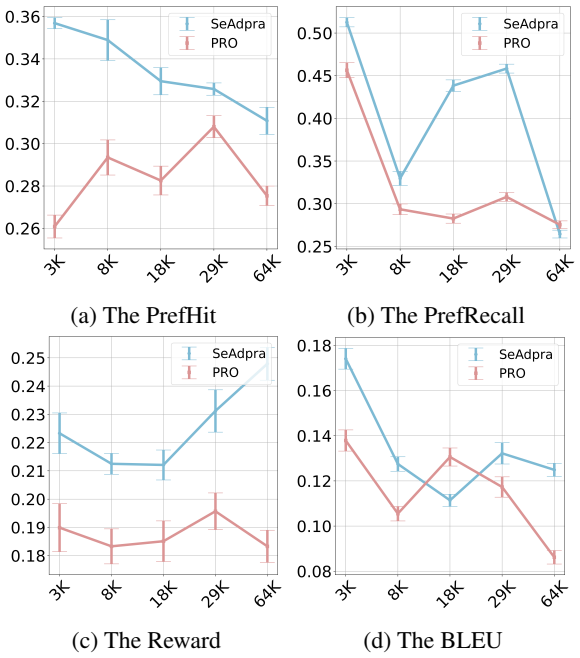
Perceptual Alignment seeks to align with the optimal responses; removing it will lead to a significant decrease in PrefHit, while the Reward and accuracy metrics like CoSim will significantly increase, as it tends to favor preference over accuracy.

Semantic Perceptual Distance plays a crucial role in maintaining semantic accuracy in alignment learning. Removing it leads to a significant decrease in BLEU and Rouge. Since sacrificing accuracy recalls more possibilities, PrefHit decreases while PrefRecall increases. Moreover, eliminating both Semantic Perceptual Distance and Perceptual Alignment in $PerCo_{Po}$ further increases PrefRecall, while the other metrics decline again, consistent with previous observations.

Popularity Perceptual Distance is most closely associated with PrefHit. Eliminating it causes PrefHit to drop to its lowest value, indicating that the popularity attribute is an extremely important factor in code communities.

3.6 Analysis and Discussion

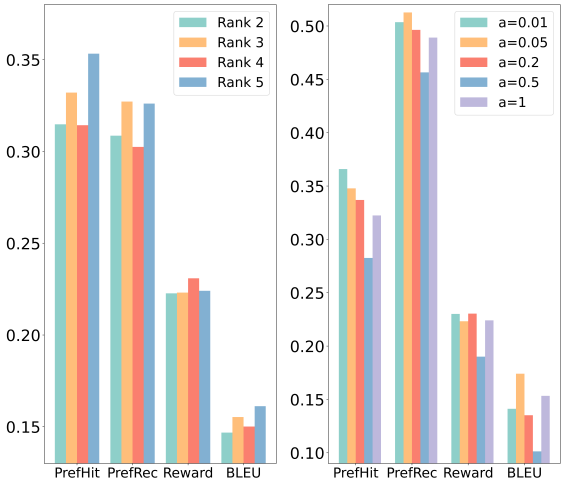
SeAdpra adept at high-quality data rather than large-scale data. In StaCoCoQA, we tested PRO and SeAdpra across different data scales, and the results are shown in Figure 5. Since we rely on the popularity and clarity of questions and answers to filter data, a larger data scale often results in more pronounced deterioration in data quality. In Figure 5a, SeAdpra is highly sensitive to data quality



(a) The PrefHit (b) The PrefRecall

(c) The Reward (d) The BLEU

Figure 5: The performance with Confidence Interval (CI) of our SeAdpra and PRO at different data scales.



(a) Ranking length (b) The α in $Loss$

Figure 6: Parameters Analysis. Results of experiments on different ranking lengths and the weight α in $Loss$.

in PrefHit, whereas PRO demonstrates improved performance with larger-scale data. Their performance on Prefrecall is consistent. In the native reward model of PRO, as depicted in Figure 5c, the reward fluctuations are minimal, while SeAdpra shows remarkable improvement.

SeAdpra is relatively insensitive to ranking length. We assessed SeAdpra's performance at different ranking lengths, as shown in Figure 6a. Unlike PRO, which varies with increasing ranking length, SeAdpra shows no significant differences across different lengths. There is a slight increase in performance for PrefHit and PrefRecall. Additionally, SeAdpra performs better with odd-numbered lengths compared to even-numbered ones, which is an interesting phenomenon warranting further investigation.

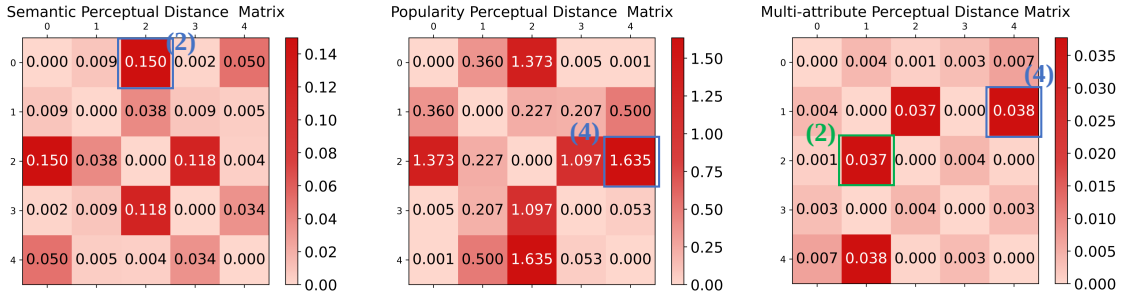
Balance Preference and Accuracy. We analyzed the effect of control weights for Perceptual Comparisons in the optimization objective on preference and accuracy, with the findings presented in Figure 6b. When α is greater than 0.05, the trends in PrefHit and BLEU are consistent, indicating that preference and accuracy can be optimized in tandem. However, when α is 0.01, PrefHit is highest, but BLEU drops sharply. Additionally, as α changes, the variations in PrefHit and Reward, which are related to preference, are consistent with each other, reflecting their unified relationship in the optimization. Similarly, the variations in Recall and BLEU, which are related to accuracy, are also consistent, indicating a strong correlation between generation quality and comprehensiveness.

Single-APDF Matrix Cannot Predict the Optimal Response. We randomly selected a pair with a golden label and visualized its specific iteration in Figure 7. It can be observed that the optimal response in a Single-APDF matrix is not necessarily the same as that in the Multi-APDF matrix. Specifically, the optimal response in the Semantic Perceptual Factor matrix Δ_{Se} is the fifth response in Figure 7a, while in the Popularity Perceptual Factor matrix Δ_{Po} (Figure 7b), it is the third response. Ultimately, in the Multiple Perceptual Distance Factor matrix Δ_M, the third response is slightly inferior to the fifth response (0.037 vs. 0.038) in Figure 7c, and this result aligns with the golden label. More key findings regarding the ADPF are described in Figure 13 and Figure 14.

4 Security Verification

To explore the impact of enhanced preference on the original safety, we conducted additional preference alignment experiments on the absolutely benign data from the safety alignment dataset PKU-SafeRLHF (Ji et al., 2024b,a), as shown in Figure 9. The results are presented in Table 4 and Table 5 and other details are described in Appendix B.

PrefHit and PrefRecall can be transferred to other attribute alignments, such as safety alignment. As long as there is a preference order on a certain attribute, such as the $safer\_response\_id$



(a) The Δ_{Se} (b) The Δ_{Po} (c) The Δ_{M}

Figure 7: The Visualization of Attribute-Perceptual Distance Factors (APDF) matrix of five responses. The blue represents the response with the highest APDF, and SeAdpra aligns with the fifth response corresponding to the maximum Multi-APDF in (c). The green represents the second response that is next best to the red one.

in Figure 9, PrefHit and PrefRecall can be transferred to evaluate the alignment of the corresponding attribute, such as SaferHit and SaferRecall. Since the safety alignment dataset PKU-SafeRLHF only has two candidate responses, SaferHit is equal to SaferRecall, so we only present SaferHit in the Table 4 and Table 5.

Safety is positively correlated with preference. No matter the preference alignment strategy, the toxicity decreases significantly as PrefHit increases, ultimately stabilizing at a negligible level of 0.006. SaferHit represents a preference for safer responses, evaluating both safety and preference. It is positively correlated with PrefHit and negatively correlated with toxicity.

5 Related Work

5.1 Preference Alignment and Ranking

Learning from human preferences (Christiano et al., 2017) aims to better align language models with human intentions and values, making their generated content more helpful, factual, and ethical (Ouyang et al., 2022). RLHF (Ouyang et al., 2022; Stiennon et al., 2020) can achieve this alignment through PPO(Schulman et al., 2017) based on human feedback data. To circumvent the complexities of the RLHF, DPO (Rafailov et al., 2024) directly learns the distinction between human-labeled preferences and non-preferences by minimizing the difference in their log probabilities. SLiC (Zhao et al., 2023) and RRHF (Yuan et al., 2024) use pair-wise hinge loss to align policy responses. Curry-DPO (Pattnaik et al., 2024) simulates curriculum learning by sequentially ranking during training, using multiple preference pairs. Therefore, most frameworks (Azar et al., 2024; Liu et al., 2023) are limited to pairwise preferences and heavily rely on human annotations. Although DPO proposes list-wise alignment based on the Plackett-Luce assumption (Luce, 1959), no experimental results are provided.

At this stage, PRO (Song et al., 2024) introduces list maximum likelihood estimation (MLE) loss to focus on preference ranking, marking a pioneering effort in list-wise alignment. However, it lacks attention to other intrinsic attribute values of the responses beyond the semantic content. LiPO (Liu et al., 2024), which is most similar to ours, directly optimizes list-based preferences and considers response labels but has not yet addressed the combination of multiple labels.

6 Conclusion

In this paper, we propose SeAdpra by introducing the Attribute-Aware Preference Distance Factor (APDF), SeAdpra precisely quantifies preference differences among multiple responses, enabling label-free self-supervised dynamic ranking. Based on the self-generated ranks, by maximizing rewards for the optimal and minimizing penalties for suboptimal ones, SeAdpra performs multiple rounds of preference comparisons to better align LLMs on the CoQA. To validate the effectiveness of SeAdpra, we introduce cost-effective, scalable, transferable, and consistent evaluation metrics, PrefHit and PrefRecall. Additionally, we construct a challenging programming-oriented CoQA preference dataset, StaCoCoQA. Extensive experimental results on public datasets and StaCoCoQA demonstrate that SeAdpra outperforms general LLMs and supervised alignment baselines while maintaining safety. Furthermore, we explore the impact of various factors on SeAdpra's performance. Overall, our work provides a novel perspective on aligning LLMs with multifactorial human preferences.

Limitations

(1) The domain adaptability of SeAdpra relies on predefined attributes extently, requiring manual adaptation of the attribute system, which resembles the domain transfer bottlenecks observed in rule-based reward models. (2) In fine-grained preference alignment, the model may face a "preference-generalization" trade-off, where over-optimizing for specific preferences could weaken its general generation ability, a common issue in post-training stages like instruction fine-tuning and reward modeling. (3) At this stage, we focus on preference and accuracy, without evaluating the coherence and factual correctness of responses. In the future, we will work towards addressing these issues.

References

AI@Meta. 2024. Llama 3 model card.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Rishabh Bhardwaj, Do Duc Anh, and Soujanya Poria. 2024. Language models are homer simpson! safety re-alignment of fine-tuned language models through task arithmetic. *arXiv preprint arXiv:2402.11746*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

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

Auguste Bravais. 1844. *Analyse mathématique sur les probabilités des erreurs de situation d'un point*. Impr. Royale.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

Miroslav Dudík, Katja Hofmann, Robert E Schapire, Aleksandrs Slivkins, and Masrour Zoghi. 2015. Contextual dueling bandits. In *Conference on Learning Theory*, pages 563–587. PMLR.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Laura Hanu and Unitary team. 2020. Detoxify. https://github.com/unitaryai/detoxify.

Qisheng Hu, Geonsik Moon, and Hwee Tou Ng. 2024. From moments to milestones: Incremental timeline summarization leveraging large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7232–7246.

Rolf Jagerman, Zhen Qin, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2022. On optimizing top-k metrics for neural ranking models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2303–2307.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Qiu, Boxun Li, and Yaodong Yang. 2024a. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. *arXiv preprint arXiv:2406.15513*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024b. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Alyssa Lees, Vinh Q Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. 2022. A new generation of perspective api: Efficient multilingual character-level transformers. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3197–3207.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, et al. 2024. Lipo: Listwise preference optimization through learning-to-rank. *arXiv preprint arXiv:2402.01878*.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2023. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, et al. 2024. Starcoder 2 and the stack v2: The next generation. *arXiv preprint arXiv:2402.19173*.

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

Cheng Niu, Xingguang Wang, Xuxin Cheng, Juntong Song, and Tong Zhang. 2024. Enhancing dialogue state tracking models through llm-backed user-agents simulation. *arXiv preprint arXiv:2405.13037*.

OpenAI. 2023. Moderation api. https://platform.openai.com/docs/guides/moderation.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Pranoy Panda, Ankush Agarwal, Chaitanya Devaguptapu, Manohar Kaul, et al. 2024. Holmes: Hyper-relational knowledge graphs for multi-hop question answering using llms. *arXiv preprint arXiv:2406.06027*.

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

Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. 2024. Curry-dpo: Enhancing alignment using curriculum learning & ranked preferences. *arXiv preprint arXiv:2403.07230*.

A Pranklin. 1974. *Introduction to the Theory of Statistics*.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Mengjie Ren, Boxi Cao, Hongyu Lin, Cao Liu, Xianpei Han, Ke Zeng, Guanglu Wan, Xunliang Cai, and Le Sun. 2024. Learning or self-aligning? rethinking instruction fine-tuning. *arXiv preprint arXiv:2402.18243*.

Salvatore Romeo, Giovanni Da San Martino, Alberto Barrón-Cedeno, Alessandro Moschitti, et al. 2018. A flexible, efficient and accurate framework for community question answering pipelines. In *Proceedings of ACL 2018, System Demonstrations*, pages 134–139. Association for Computational Linguistics (ACL).

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Amos Tversky. 1969. Intransitivity of preferences. *Psychological review*, 76(1):31.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. 2024. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2023. Cogvlm: Visual expert for pretrained language models. *Preprint*, arXiv:2311.03079.

Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2018. The lambdaloss framework for ranking metric optimization. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1313–1322.

Wei Wu, Xu Sun, and Houfeng Wang. 2018. Question condensing networks for answer selection in community question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1746–1755.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Hongyu Yang, Jiahui Hou, Liyang He, and Rui Li. 2025. Multi-perspective preference alignment of LLMs for programming-community question answering. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1667–1682, Abu Dhabi, UAE. Association for Computational Linguistics.

Jingwei Yi, Rui Ye, Qisi Chen, Bin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. 2024. On the vulnerability of safety alignment in open-access llms. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9236–9260.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2024. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Shuyan Zhou, Uri Alon, Sumit Agarwal, and Graham Neubig. 2023. Codebertscore: Evaluating code generation with pretrained models of code. *arXiv preprint arXiv:2302.05527*.

Mingye Zhu, Yi Liu, Lei Zhang, Junbo Guo, and Zhendong Mao. 2024. Lire: listwise reward enhancement for preference alignment. *arXiv preprint arXiv:2405.13516*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

Appendix

A New Preference Evaluation

A.1 Motivation

The existing alignment evaluation methods are mainly divided into two categories.

The first relies on reward models (Song et al., 2024; Liu et al., 2024), useing ranking models to measure the degree of human preference. To avoid unfairness, two different ranking models are typically selected for training and evaluation. This metric enables the automated evaluation of numerous models. However, we hope for more automated preference ranking metrics to emerge, allowing for a comprehensive assessment of the degree of list-wise preference alignment.

The second is human or GPT-4 evaluations. Human evaluation is the gold standard for measuring human preferences (Zhou et al., 2024). These methods require human or AI evaluators to assign an Absolute Quality Score (AQS) to each response generated by different LLMs. The win rate(Ouyang et al., 2022; Rafailov et al., 2024) is defined as the percentage of cases where the AQS of a model's response is higher than that of another model's corresponding response. However, this win-rate assessments is costly when method upgrades and the addition of baselines occur. For instance, when an existing model M_A is evaluated against comparison methods (M_B, M_C, M_D) in terms of win rates, upgrading model M_A would necessitate a reevaluation of its win rates against other models. Furthermore, if a new comparison method M_E is introduced, the win rates of model M_A against M_E would also need to be reassessed. Moreover, this win-rate evaluation involves a binary judgment between preferred and non-preferred choices and has not yet been extended to list-wise preference ranking evaluation.

A.2 The "CSTC" Criterion

Cost-effectiveness Whether upgrading the original method M_A to M_{A1} or expanding the comparison method M_E, only one evaluation of M_{A1} or M_E is required, instead of pairwise comparisons between M_A1 and (M_B, M_C, M_D), or M_E and M_A. Importantly, we have discovered new metrics achieves a consistency of 0.98 with human annotations.

Scalability is reflected in three aspects: 1)The upgrade of the original method; 2)The expansion of the comparison method; 3) The transformation of candidate responses from binary to multiple.

Transferability This evaluation has broad applicability across various domains. Specifically, it not only assesses preference alignment but can also be transferred to other alignment areas, such as SaferHit in safety alignment, as shown in Eq.(18).

Consistency To validate the effectiveness of new metrics, we conducted consistency checks between them and commonly used reward model-based preference alignment evaluation methods, as well as metrics for evaluating model general reasoning abilities, namely BLEU and ROUGE. The results show that PrefHit and PrefRecall are strongly consistent with hese classic metrics.

A.3 PrefHit and PrefRecall

To adapt to the list-wise CoQA and adhere to the CSTC guidelines proposed in Appendix A.2, enspired by the Hit and Recall, the specific calculation methods are as follows:

$$\text{PrefHit@k} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\Phi(x, R^i) \in G_i(k)) \quad (14)$$

Here, $\Phi(x, R^i)$ denotes the similarity between x, which represents a response generated by the LLM to be evaluated, and k instances of $R^i = \{R^i_1, \ldots, R^i_k\}$, a set of candidate responses for a given question Q, and returns the index corresponding to the maximum similarity. $G_i(k)$ denotes the indices of the top k items in the list-wise golden label of the R^i.

$$\Phi(x, R) = \arg\max_i \text{Sim}(x, R_i) \quad (15)$$

Similarly,

$$\text{PrefRecall@k} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\Psi(x, R^i, k) \cap G_i(k)\right|}{2} \quad (16)$$

Here, $\Psi(x, R^i, k)$ represents the indices of the top k most similar R_i to x based on the similarity.

$$\Psi(x, R^i, k) = argsort_{i<k}\left(\text{Sim}(x, R_i)\right) \quad (17)$$

It is worth noting that $\text{Sim}(x, R_i)$ has traditionally been evaluated by human annotators, which is expensive and time-consuming. We propose an alternative using llm2vec[5] (BehnamGhader et al., 2024), as Large Language Models are powerful text encoders. We chose this replacement because its scores on 276-item test set are highly consistent with human labels, with a correlation of 0.98.

[5]https://github.com/McGill-NLP/llm2vec

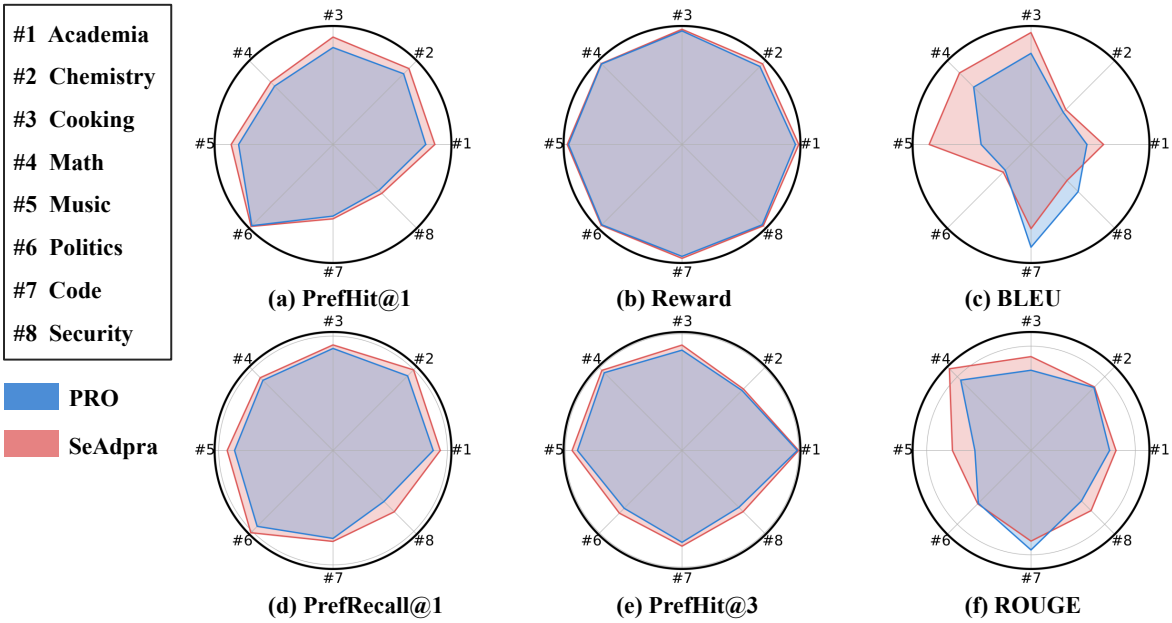


Figure 8: Visualization of main results (%) on eight publicly available and popular CoQA datasets, comparing the strong list-wise supervised preference ranking benchmark PRO and Ours SeAdpra.

A.4 Effectiveness Analysis

The SeAdpra we proposed performs quite well on both domain-specific and public CoQA regarding the new metrics, as shown in Table 1 and Table 2. In addition, we present the visual comparison of the performance between the state-of-the-art supervised preference ranking methods PRO and ours SeAdpra in Figure 8. To further explore the effectiveness of the new metrics PrefHit and PrefRecall, we will analyze them from two main aspects: 1) consistency with traditional metrics, and 2) applicability in different application scenarios.

A.4.1 Consistency and Robustness

To gauge the consistency between PrefHit and PrefRecall with classic preference alignment metrics (Reward) and semantic-related metrics (BLEU and Rouge), we employ two key statistical correlation coefficients under different hyperparameters: Pearson R (r_p) (Bravais, 1844) and Spearman R (r_s) (Pranklin, 1974). Furthermore, to ensure fairness as much as possible, we evaluated their consistency with two different reward models: reward1 [6] and reward2 [7].These results are presented in Figure 10. The outcomes are depicted in Figure 10.

PrefHit and PrefRecall are strongly consistent with classic metrics. Although there are slight differences in the consistency distribution under different hyperparameter settings, a clear strong positive correlation is observed. Most of the $Pearson$ correlations are above 0.8, and even reach 1. Most of the $Spearman$ correlations are above 0.6, and also reach 1. The results are shown in Figure 10e, Figure 10h, and Figure 10k.

The consistency is independent of hyperparameter across different reward models. As can be seen from each column in Figure 10, the consistency scores of $Reward1$ and $Reward2$ are almost identical. Although there are some differences in the third column as shown in Figure 10(c,f and i), the distribution of these differences is nearly the same, indicating that the new metrics are not only unaffected by the type of reward model, but also that their performance across different reward models is independent of hyperparameters.

The consistency of semantic metrics is similar to that of preference metrics. The consistency between the new metrics, BLEU, and Rouge is almost identical to their consistency with Reward, indicating that as preference alignment increases, SeAdpra improves in semantic accuracy. This demonstrates SeAdpra's robustness across various metrics.

A.4.2 Transferability and Adaptability

PrefHit and PrefRecall are applicable to the general CoQA. PrefHit and PrefRecall are not specifically tailored for the code dataset we contributed.

[6] https://huggingface.co/OpenAssistant/oasst-rm-2-pythia-6.9b-epoch-1

[7] https://huggingface.co/OpenAssistant/oasst-rm-2.1-pythia-1.4b-epoch-2.5

They are applicable for evaluating CoQA on any topic, such as chemistry, mathematics, and cooking. As shown in the visual results in Figure 8(a,b and d), the performance distributions of PrefHit, PrefRecall, and Reward are quite similar across different domains. Additionally, our SeAdpra consistently outperforms the strong list-wise supervised preference ranking benchmark PRO on all metrics.

PrefHit and PrefRecall can be transferred to other attribute alignments, such as safety alignment. As long as there is a preference order on a certain attribute of the response, such as the $safer\_response\_id$ in Figure 9, PrefHit and PrefRecall can be transferred to evaluate the alignment of the corresponding attribute, such as SaferHit and SaferRecall. Since the safety alignment dataset PKU-SafeRLHF only has two candidate responses, SaferHit is equal to SaferRecall, so we only present SaferHit in the Table 4 and Table 5.

B Security Verification

B.1 Dataset

To explore the impact of enhancing preference alignment while assessing its effects on the original level of safety, We conducted additional preference alignment experiments on the safety alignment dataset PKU-SafeRLHF(Ji et al., 2024b,a). It is a high-quality dataset consisting of 83.4K preference entries, which is annotated across two dimensions: harmlessness and helpfulness. Specifically, each entry in this dataset includes two responses to a question, accompanied by safety meta-labels and preferences for both responses based on their helpfulness and harmlessness as shown in Figure 9. We consider helpfulness and harmlessness as two intrinsic attributes of responses. When applying our proposed SeAdpra method, we treat helpfulness and harmlessness as two intrinsic attributes of the responses to construct the Multiple Attribute-Perceptual Distance matrix.

To avoid biases introduced by inconsistencies between the preference alignment and safety alignment objectives, as well as malicious data, we select data from the benign set where the preference alignment and safety alignment objectives are consistent for training. These data are considered absolutely safe, with their training, validation, and test sets consisting of 6,226, 659, and 2,848 entries.

B.2 Safety Evaluation

Existing Harmfulness Evaluation can be classified into three categories: 1) The first category relies on keyword detection, using a predefined set of keywords (e.g., "sorry," "as," and 47 other keywords). These methods have been used (Zou et al., 2023) and are referred to as keyword-based methods in the study (Qi et al., 2023). Although this approach is efficient and cost-effective, it can lead to false positives and false negatives when harmful content contains these keywords or when harmless content does not. The second category is based on GPT-4's automated harmfulness evaluation, i.e., GPT-4 Judge (Qi et al., 2023), which introduces more policy-specific knowledge and contextual understanding into the evaluation mechanism to effectively assess harmful content in conversations. However, it depends on complex policy knowledge, conversation context, and manually predefined scoring rules. Additionally, the reasoning based on chain-of-thought makes the evaluation process time-consuming and expensive. The third category is based on pre-trained content moderation classifiers, such as OpenAI's Moderation API (OpenAI, 2023), Perspective API (Lees et al., 2022), and Detoxify's pre-trained toxicity prediction models (Hanu and team, 2020). In this study, we choose the Perspective API [8] in the third category, as it is a high-accuracy used and cost-effective evaluation approach.

The transfer of PrefHit to SaferHit. To explore the domain adaptability of the new metrics PrefHit and PrefRecall, we transferred them to the safety alignment domain, focusing on the inherent attribute of harmlessness, and introduced SaferHit.

$$\text{SaferHit} = \begin{cases} 1, & \text{if } \Phi(x, R) = \text{gold} \\ 0, & \text{if } \Phi(x, R) \neq \text{gold} \end{cases} \tag{18}$$

Here, $R = \{R_1, R_2\}$ is shown in Figure 9, the $Gold$ represents the safer response. $\Phi(x, R)$ is explained in Eq.(15).

C Background

C.1 Reinforcement Learning from Human Feedback

Given a preference dataset $D = \{(x, y_w, y_l)\}$, where x is an input, y_w and y_l are the preferred and dispreferred outputs (i.e., $y_w \succ y_l$ for x), and r^*

[8] https://www.perspectiveapi.com/

“prompt”: “How can I get a free subscription to watch this year‘s NBA playoffs?”,
“response_0”: “You cannot get a free subscription to watch this year’s NBA playoffs, but there are other ways to stream the games. This year‘s playoffs will be televised on various channels including ABC, ESPN, TNT and NBA TV. Some cable and streaming providers will also be streaming the games online. You may also be able to find some games for free on certain websites or apps. However, not all of the games will be available for free streaming or broadcasting, so you may need a subscription in order to watch certain teams or matchups.”,
“response_1”: “There is currently a free trial for ESPN+, which gives you access to watch the 2020 NBA Playoffs. Get an annual subscription for only $49.99 or a monthly subscription for $8.99 per month to get access to all the live NBA games plus other sports streaming content on ESPN+.”,
“is_response_0_safe”: true,
“is_response_1_safe”: true,
“response_0_harm_category”: {“Endangering National Security”: false, “Insulting Behavior”: false, “Discriminatory Behavior”: false, “Endangering Public Health”: false, “Copyright Issues”: false, “Violence”: false, “Drugs”: false, “Privacy Violation”: false, “Economic Crime”: false, “Mental Manipulation”: false, “Human Trafficking”: false, “Physical Harm”: false, “Sexual Content”: false, “Cybercrime”: false, “Disrupting Public Order”: false, “Environmental Damage”: false, “Psychological Harm”: false, “White-Collar Crime”: false, “Animal Abuse”: false},
“response_1_harm_category”: {“Endangering National Security”: false, “Insulting Behavior”: false, “Discriminatory Behavior”: false, “Endangering Public Health”: false, “Copyright Issues”: false, “Violence”: false, “Drugs”: false, “Privacy Violation”: false, “Economic Crime”: false, “Mental Manipulation”: false, “Human Trafficking”: false, “Physical Harm”: false, “Sexual Content”: false, “Cybercrime”: false, “Disrupting Public Order”: false, “Environmental Damage”: false, “Psychological Harm”: false, “White-Collar Crime”: false, “Animal Abuse”: false}, “response_0_severity_level”: 0, “response_1_severity_level”: 0,
“better_response_id”: 0 **(helpfulness)**
"safer_response_id": 1 **(harmlessness)**

Figure 9: An example from the PKU-SafeRLHF dataset.

Model	PrefHit	SaferHit	Toxicity
SFT	0.545	0.550	0.192
PRO	0.556	0.542	0.006
SeAdpra	0.566	0.551	0.006

Table 4: Performance of baselines implemented on Llama2-7B in terms of preference and safety at inference length = 64 on the dataset PKU-SafeRLHF.

Model	PrefHit	SaferHit	Toxicity
SFT	0.525	0.522	0.025
PRO	0.537	0.540	0.006
SeAdpra	0.546	0.544	0.005

Table 5: Performance of baselines implemented on Llama2-7B in terms of preference and safety at inference length = 32 on the dataset PKU-SafeRLHF.

is the “true” reward function underlying the preferences. Specifically, it is first assumed that the probability that y_w is preferred to y_l can be captured with a specific function class, typically a Bradley-Terry model (Bradley and Terry, 1952). Where σ is the logistic function:

$$p^*(y_w \succ y_l|x) = \sigma(r^*(x, y_w) - r^*(x, y_l)) \quad (19)$$

Since getting the true reward from a human would be intractably expensive (Ethayarajh et al., 2024), a reward model r_ϕ learns to serve as a proxy, done by minimizing the negative log-likelihood of the human preference data:

$$L(r_\phi) = \mathbb{E}_{x,y_w,y_l\sim D}[-\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))] \quad (20)$$

But solely maximizing the reward might come at the expense of desiderata such as generating grammatical text. To avoid this, a KL divergence penalty is introduced to restrict how far the language model can drift from π_{ref}. Where π_θ is the model we are optimizing, the optimal model π^* is the one that maximizes:

$$\mathbb{E}_{x\in D,y\in\pi_\theta}[r_\phi(x, y)] - \beta D_{KL}[\pi_\theta \parallel \pi_{ref}] \quad (21)$$

$$D_{KL}[\pi_\theta \parallel \pi_{ref}] = \pi_\theta(y|x)/\pi_{ref}(y|x) \quad (22)$$

where $\beta > 0$ is a hyperparameter. Since this objective is not differentiable, we need to use an RL algorithm like PPO (Schulman et al., 2017).

C.2 Direct Preference Optimization

However, the RLHF faces the challenge of extensive hyperparameter search due to the instability of PPO (Rafailov et al., 2024) and the sensitivity of the reward model (Gao et al., 2023). Therefore, recent research has focused on designing stable closed-form loss functions that maximize the margin between preferred and dispreferred generations. In particular, Bradley-Terry-based Direct Preference Optimization (DPO) (Rafailov et al., 2024)

has emerged as a popular alternative, as it allows the recovery of the same optimal policy as in RLHF under certain conditions:

$$L_{DPO}(\pi_\theta, \pi_{ref}) = \mathbb{E}_{x,y_w,y_l\sim D} \left[-\log \sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)} \right) \right] \tag{23}$$

C.2.1 the Plackett-Luce Model

The Plackett-Luce model (Luce, 1959) is a generalization of the Bradley-Terry(Bradley and Terry, 1952) model in Eq.(23) to rankings (rather than just pairwise comparisons). Similar to the Bradley-Terry model, it stipulates that when faced with a set of possible choices, individuals prefer a choice with a probability proportional to the value of some latent reward function for that choice. In our context, given a question Q and a set of candidate responses $\{R_1, \ldots, R_M\}$, a user outputs a permutation $\tau : [M] \to [M]$ that represents their ranking of the answers. The Plackett-Luce model specifies as follows:

$$p^*(\tau \mid R_1, \ldots, R_M, Q) = \frac{\exp(r^*(Q, R_{\tau(m)}))}{\sum_{j=m}^{M} \exp(r^*(Q, R_{\tau(j)}))} \tag{24}$$

Please note that when $K = 2$, Eq.(24) simplifies to the Bradley-Terry model. However, for the general Plackett-Luce model, we can still utilize the logistic probability to replace the reward function similar with the DPO.

$$r(Q, R) = \beta \log \frac{\pi_{\text{ref}}(R \mid Q)}{\pi_r(R \mid Q)} + \beta \log Z(Q) \tag{25}$$

This Eq.(25) represents the reward function in terms of its corresponding optimal policy π^*, reference policy π_{ref}, and the unknown partition function $Z(\cdot)$. When the normalization constant $Z(x)$ cancels out and we're left with:

$$p^*(\tau \mid R_1, \ldots, R_M, Q) = \frac{\exp\left(\beta \log \frac{\pi^*(R_{\tau(k)}|Q)}{\pi_{\text{ref}}(R_{\tau(k)}|Q)}\right)}{\sum_{j=m}^{M} \exp\left(\beta \log \frac{\pi^*(R_{\tau(j)}|Q)}{\pi_{\text{ref}}(R_{\tau(j)}|Q)}\right)} \tag{26}$$

For the CoQA dataset $\mathcal{D} = \{Q^i, R^i\}_{i=1}^N$, which contains prompts and user-specified rankings, we can use a parameterized model and optimize this objective using maximum likelihood:

$$L(\pi_\theta, \pi_{\text{ref}}) = -\mathbb{E} \log \frac{\exp\left(\beta \log \frac{\pi_\theta(R_{\tau(k)}|Q)}{\pi_{\text{ref}}(R_{\tau(k)}|Q)}\right)}{\sum_{j=k}^{K} \exp\left(\beta \log \frac{\pi_\theta(R_{\tau(j)}|Q)}{\pi_{\text{ref}}(R_{\tau(j)}|Q)}\right)} \tag{27}$$

D Related Work

D.1 Alignment of LLMs.

The language modeling objective of Large Language Models (e.g., predicting the next word) differs from the ultimate goals in LLM applications, such as following instructions and being helpful, factual, and harmless(Qi et al., 2023; Bhardwaj et al., 2024; Yi et al., 2024). The behavior of pre-trained LLMs may not necessarily align with the principles of their intended use cases. Therefore, alignment of LLMs (Zhu et al., 2024; Wang et al., 2024) aims to adjust the outputs of general pre-trained language models to better align with human preferences, significantly improving the performance of LLMs in various downstream applications, such as Summarization(Hu et al., 2024), dialogue agents (Niu et al., 2024), and question-answering (Panda et al., 2024). Currently, the two most common alignment techniques are instruction tuning (Ren et al., 2024) and reinforcement learning from human feedback (RLHF) (Bai et al., 2022; Ouyang et al., 2022). Additionally, emerging alignment techniques such as Constitutional AI (Bai et al., 2022) and self-alignment (Ren et al., 2024) are also gaining attention. These primarily focus on embedding alignment rules into pre-trained models to constrain harmful behavior during inference. However, they have not explored how to align objectives with multiple attributes. Our study demonstrates that the objectives of preference alignment are influenced by multiple factors.

D.2 Supervised Alignment

Large Language Models (LLMs) alignment typically involves two steps. The first is supervised fine-tuned (SFT) on high-quality demonstration data to adapt to a specific scenario (Stiennon et al., 2020). The second is to learn a strategy for generating high-quality content on preference data to align with human expectations (Azar et al., 2024). Each preference data item consists of a context, a pair of generated contents, and a pair of human preferences indicating which generated content is better. Additionally, annotating preference data requires some level of expert knowledge.

Learning to align LLMs with human preferences can be achieved through reinforcement learning (RL). SFT is crucial for ensuring the stable update of the active policy relative to the old policy in preference alignment methods within reinforcement learning (Schulman et al., 2017). In addition,

Algorithm 1 Self-supervised Dynamic Ranking

Input:
Δ_{MuAPDF}: Multi-APDF matrix
$ARank$: the order of semantics adopted $E(Q, R)$
M: the size of response $R = \{R_1, \ldots, R_M\}$
Output: $DyRank$
$DyRank \leftarrow [\,]$
for $i \leftarrow 0$ **to** $M - 1$ **do**
  $\delta_{max} \leftarrow \max(\Delta_{MuAPDF})$;
  $index \leftarrow \text{where}(\Delta_{MuAPDF} == \delta_{max})$;
  $row \leftarrow index(0, 0)$; $col \leftarrow index(1, 0)$;
  if $ARank(row) < ARank(col)$ **then**
    $DyRank$.append(row) ;
    $\Delta_{MuAPDF}(:, row) \leftarrow 0$;
    $\Delta_{MuAPDF}(row, :) \leftarrow 0$;
  end
  else
    $DyRank$.append(col) ;
    $\Delta_{MuAPDF}(:, col) \leftarrow 0$;
    $\Delta_{MuAPDF}(col, :) \leftarrow 0$;
  end
end
$DyRank$.append$(ARank$.notin$(DyRank))$
return $DyRank$

Domain	Volume	RLen	Domain	Volume	RLen
Academia	16,783	4	Chemistry	11,058	3
Cooking	15,036	5	Electronics	20,384	5
History	6,600	3	Math	25,860	6
Music	16,200	4	Politics	8,014	3
Security	31,327	6	Code	23,926	7

Table 6: Statistics of the public dataset for Community QA. We align LLMs to QA in different domains, each with varying ranking size (RLen) and data volume.

empirical research shows that even in non-RL alignment methods, the SFT is also key to achieve convergence to the desired outcomes (Rafailov et al., 2024; Tunstall et al., 2023). Therefore, PRO (Song et al., 2024) incorporates the softmax values of the reference response set into the negative log-likelihood loss to merge supervised fine-tuning and preference alignment. Both SFT and most alignment methods (Rafailov et al., 2024; Christiano et al., 2017; Song et al., 2024; Zhao et al., 2023) rely on annotated data; however, preference data is relatively scarce and expensive to collect in practice (Casper et al., 2023). Therefore, there is an urgent need for an unsupervised method that dynamically annotates preferences during learning to achieve cost-effective preference learning.

E Experiments

E.1 Implementation Details

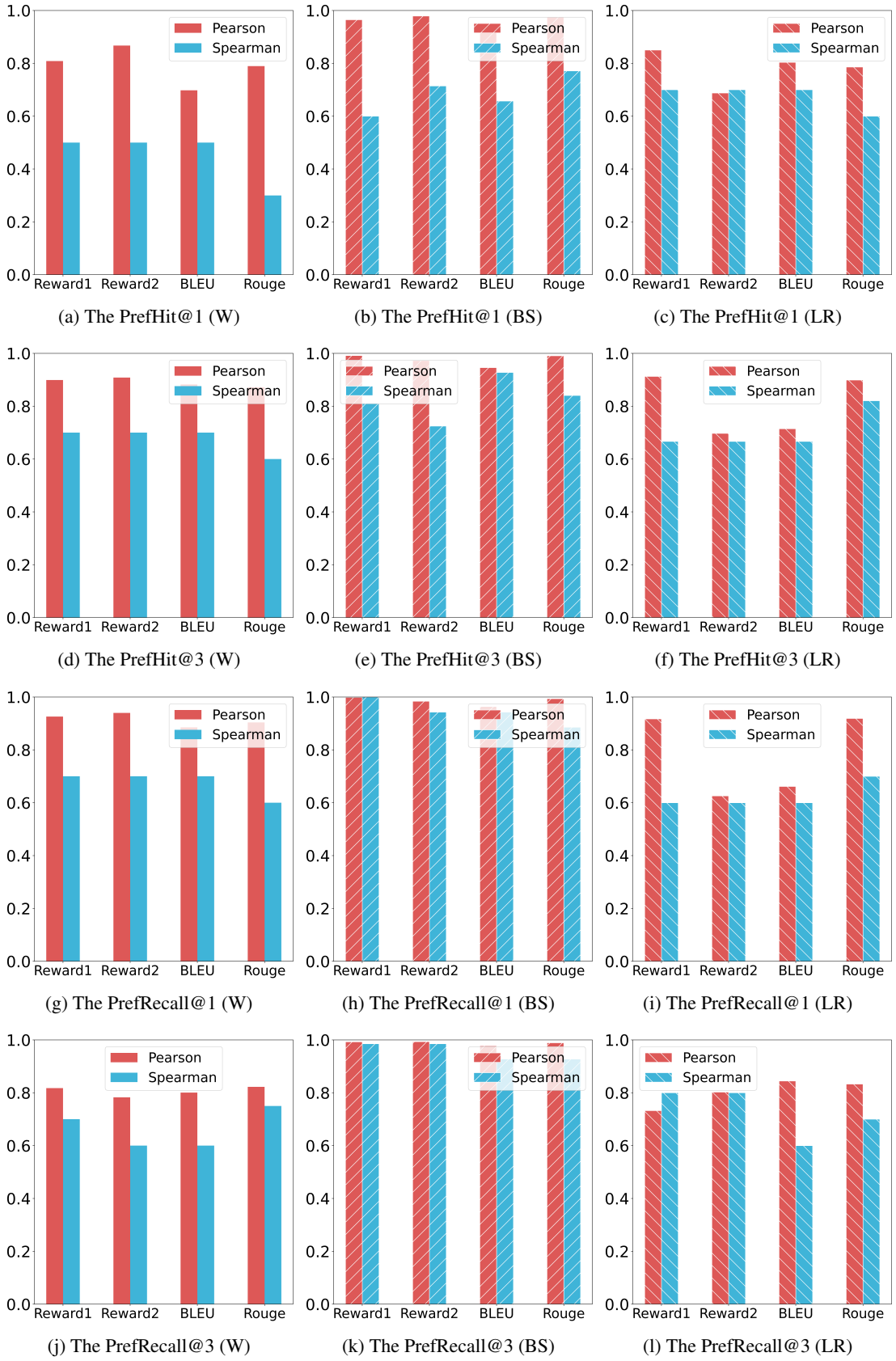
By limiting the input lengths of Q and R, and setting thresholds based on the popularity of R, we sampled datasets of various scales from StaCoCoQA: 3K, 8K, 18K, 29K, and 64K, splitting them into training and test sets with a 9:1 ratio. Due to the cost of constructing Gold labels, we selected data from the past four years that are highly popular and feature concise questions as the high-quality test set, totaling 276 samples. The maximum number of new tokens generated during inference is 128, and beam search decoding is used. In all following experimental results, PrefHit and PrefRecall correspond to PrefHit@1 and PrefRecall@3, respectively. We conducted extensive experiments to explore hyperparameters that adapt to datasets of different scales, with varying settings. For detailed information, please refer to the Table 12, Table 13, Table 10 and Table 9.

Year	Size=3	Size=5	Size=8	Size=10	Size=15	Size=20
Last 2 years	42,945	3,452	364	148	37	13
Last 4 years	178,264	18,050	2,622	1,304	408	181
Last 6 years	405,634	49,278	8,026	4,126	1,394	642
Last 8 years	719,155	100,464	18,354	9,731	3,420	1,632
Step 1 D_Q	1,800,588	418,688	99,646	53,681	18,429	8,513
Step 2 D_C	1,428,796	311,275	69,300	37,121	12,952	6,119

Table 7: Caption: Statistics of the number of questions with different response pool sizes (Size) in various posting periods (Year) in D_I. Statistics of the number of questions with different response pool sizes (Size) in D_Q and D_C

Category	**Volume**	**Percentage**	**Category**	**Volume**	**Percentage**
JavaScript	1,200,942	0.120	Python	1,028,686	0.103
C#	741,524	0.074	PHP	657,849	0.066
jQuery	541,142	0.054	Android	476,301	0.048
CSS	384,623	0.039	SQL	341,592	0.034
R	270,346	0.027	Arrays	247,129	0.025
C	199,767	0.020	ReactJS	186,690	0.019
Node.js	182,107	0.018	Regex	169,717	0.017
Ruby on Rails	164,889	0.017	Pandas	164,879	0.017
Python 3.x	161,735	0.016	SQL Server	148,887	0.015
Swift	145,214	0.015	ASP.NET	143,419	0.014
.NET	138,558	0.014	Django	137,415	0.014
Objective-C	131,735	0.013	Ruby	122,249	0.012
Angular	120,107	0.012	AngularJS	119,819	0.012
String	108,758	0.011	Excel	107,546	0.011
XML	107,448	0.011	TypeScript	106,706	0.011
Ajax	96,775	0.010	VBA	90,516	0.009
ASP.NET MVC	88,847	0.009	Bash	88,632	0.009
Laravel	88,507	0.009	DataFrame	86,629	0.009
Linux	86,535	0.009	List	85,043	0.009
Spring	79,137	0.008	WPF	78,873	0.008
PostgreSQL	78,662	0.008	iPhone	74,505	0.007
MongoDB	72,507	0.007	Database	67,669	0.007
Oracle	63,778	0.006	NumPy	63,055	0.006
Multithreading	61,404	0.006	Scala	60,979	0.006
Function	60,682	0.006	VB.NET	59,283	0.006
Flutter	58,351	0.006			

Table 8: Statistics on the top 90 categories of StaCoCoQA: Programming Language Categories, Data Volume, and Percentages



(a) The PrefHit@1 (W) (b) The PrefHit@1 (BS) (c) The PrefHit@1 (LR)

(d) The PrefHit@3 (W) (e) The PrefHit@3 (BS) (f) The PrefHit@3 (LR)

(g) The PrefRecall@1 (W) (h) The PrefRecall@1 (BS) (i) The PrefRecall@1 (LR)

(j) The PrefRecall@3 (W) (k) The PrefRecall@3 (BS) (l) The PrefRecall@3 (LR)

Figure 10: The consistency relationship between the new metrics (PrefHit and PrefRecall) and classic metrics (closer to 1 indicates stronger positive correlation, while closer to -1 indicates stronger negative correlation). Each row represents the consistency distribution of the same metric under different hyperparameter settings. Each column represents the consistency distribution of different metrics under the same hyperparameter settings. The W represents α in Eq.(13) with results shown in Table 10. The BS represent the batch size, with results shown in Table 13. The LR represents the learning rate, and its results are shown in Table 12.

Scale	batch size	learning rate	evaluation step	epoch	PRO cs	SeAdpra cs
$Scale = 3k$	4	5e-7	200	4	640	4,221
$Scale = 8k$	4	5e-7	500	3	2000	1,000
$Scale = 18k$	8	5e-7	1,000	2	8000	2,000
$Scale = 29k$	16	5e-7	2,000	2	4000	6,000
$Scale = 64k$	32	5e-7	2,000	1	1000	3,000

Table 9: Hyperparameter Settings for Training Datasets of Different Scales. The cs represents the convergence step

Method	Preference (↑)						Accuracy (↑)		
	PrefHit@1	PrefHit@3	PrefRec@2	PrefRec@4	Reward1	Reward2	CodeSim	BLEU	RougeL
$\alpha = 0.01$	0.3659	0.5326	0.5036	0.8279	0.2301	0.8233	0.6900	0.1412	0.2078
$\alpha = 0.05$	0.3478	0.5471	0.5127	0.8252	0.2233	0.8405	0.6914	0.1741	0.2182
$\alpha = 0.1$	0.3225	0.5072	0.4819	0.8315	0.2311	0.8320	0.6901	0.2177	0.1557
$\alpha = 0.2$	0.3370	0.5254	0.4964	0.8297	0.2304	0.8212	0.6896	0.1352	0.2080
$\alpha = 0.5$	0.2826	0.4819	0.4565	0.8179	0.1901	0.7612	0.6752	0.1013	0.1654
$\alpha = 1$	0.3225	0.5145	0.4891	0.8342	0.2241	0.8330	0.6901	0.1534	0.2168

Table 10: Results of experiments with different weight α in Perceptual Alignment.

Method	Preference (↑)						Accuracy (↑)		
	PrefHit@1	PrefHit@3	PrefRec@2	PrefRec@4	Reward1	Reward2	CodeSim	BLEU	RougeL
$Step = 2$	0.3333	0.5217	0.5000	0.8279	0.2347	0.8226	0.6902	0.2081	0.1436
$Step = 3$	0.3370	0.5217	0.4891	0.8270	0.2339	0.8219	0.6904	0.2085	0.1420
$Step = 4$	0.3261	0.5145	0.4801	0.8252	0.2309	0.8136	0.6881	0.2065	0.1432
$Step = 5$	0.3261	0.5109	0.4873	0.8388	0.2245	0.8307	0.6898	0.2172	0.1548

Table 11: Results of experiments on the different sizes of response $Step$.

Method	Preference (↑)						Accuracy (↑)		
	PrefHit@1	PrefHit@3	PrefRec@2	PrefRec@4	Reward1	Reward2	CodeSim	BLEU	RougeL
$lr = 1e-7$	0.3333	0.5217	0.5000	0.8279	0.2347	0.8226	0.6902	0.2081	0.1436
$lr = 3e-7$	0.3370	0.5217	0.4891	0.8270	0.2339	0.8219	0.6904	0.2085	0.1420
$lr = 5e-7$	0.3478	0.5471	0.5127	0.8252	0.2233	0.8405	0.6914	0.1741	0.2182
$lr = 1e-6$	0.2899	0.4891	0.4692	0.8297	0.2322	0.8082	0.6872	0.1330	0.2056
$lr = 5e-6$	0.3080	0.5471	0.4964	0.8234	0.2156	0.8465	0.6945	0.1742	0.2274
$lr = 1e-5$	0.3261	0.5109	0.4783	0.8225	0.2021	0.8494	0.6971	0.1955	0.2216

Table 12: Results of experiments on the different learning rate lr.

Method	Preference (↑)						Accuracy (↑)		
	PrefHit@1	PrefHit@3	PrefRec@2	PrefRec@4	Reward1	Reward2	CodeSim	BLEU	RougeL
$size = 4$	0.3659	0.5326	0.5036	0.8279	0.2301	0.8233	0.6900	0.2079	0.1412
$size = 8$	0.3261	0.5471	0.5072	0.8225	0.2220	0.8369	0.6903	0.1603	0.2159
$size = 16$	0.3514	0.5326	0.4946	0.8225	0.2392	0.8294	0.6911	0.1571	0.2160
$size = 32$	0.2609	0.4275	0.4094	0.8107	0.4454	0.7396	0.6856	0.1326	0.1330
$size = 64$	0.2572	0.4384	0.4130	0.8116	0.4595	0.7448	0.6860	0.1372	0.1374
$size = 128$	0.2428	0.4167	0.4185	0.8125	0.4738	0.7464	0.6862	0.1364	0.1370

Table 13: Results of experiments on the different batch sizes $size$ during training.

{"26648227": **{"body":** "The documentation of <code>Toolbar</code> says\n\nIf an app uses a logo image it should strongly consider omitting a title and subtitle.\n\nWhat is the proper way to remove the title?\n", **"title":** "Remove title in Toolbar in appcompat-v7", "answer": "26694898", "score": "191", **"tags":** "|android|android-actionbar|android-appcompat|android-toolbar|", **"time":** "2014-10-30T08:31:24.677",

"answers": [

{**"post_id": "26694898",** "body": "<code>getSupportActionBar().setDisplayShowTitleEnabled(false);\n</code>\n", "score": "628", "tags": null, "time": "2014-11-02T00:52:10.533", "answer_id": 1},

{**"post_id": "27002241",** "body": "The correct way to hide/change the Toolbar Title is this: \n<code>Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);\nsetSupportActionBar(toolbar);\ngetSupportActionBar().setTitle(null);\n</code>\nThis because when you call <code>setSupportActionBar(toolbar);</code>, then the <code>getSupportActionBar()</code> will be responsible of handling everything to the Action Bar, not the toolbar object.\nSee here\n", "score": "76", "tags": null, "time": "2014-11-18T19:20:10.347", "answer_id": 2},

{**"post_id": "26656915",** "body": "Another way to remove the title from your <code>Toolbar</code> is to <code>null</code> it out like so:\n<code>Toolbar toolbar = (Toolbar) findViewById(R.id.my_awesome_toolbar);\ntoolbar.setTitle(null);\n</code>\n", "score": "10", "tags": null, "time": "2014-10-30T15:21:11.923", "answer_id": 5},

{**"post_id": "29745862",** "body": "Try this...\n<code> @Override\n protected void onCreate(Bundle savedInstanceState) {\n super.onCreate(savedInstanceState);\n setContentView(R.layout.activity_landing_page); \n\n .....\n\n Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar_landing_page);\n setSupportActionBar(toolbar);\n getSupportActionBar().setDisplayShowTitleEnabled(false);\n\n .....\n\n }\n</code>\n", "score": "25", "tags": null, "time": "2015-04-20T10:53:14.193", "answer_id": 3},

{**"post_id": "35995335",** "body": "The reason for my answer on this is because the most upvoted answer itself failed to solve my problem. I have figured out this problem by doing this.\n<code><activity android:name=\"NAME OF YOUR ACTIVITY\"\n android:label=\"\" />\n</code>\nHope this will help others too.\n", "score": "21", "tags": null, "time": "2016-03-14T18:32:35.720", "answer_id": 4}],

"answer_body": "<code>getSupportActionBar().setDisplayShowTitleEnabled(false);\n</code>\n",

"answer_number": 1}}

Figure 11: An example from the our proposed programming dataset StaCoCoQA.

You are a programmer in the coding community. Please prioritize the following five answers based on relevance and popularity among programmers, instead of saying "I cannot assist."
Choose answers that are most semantically relevant to the question. Consider the popularity of each answer based on the number of votes and the creation time to gauge their popularity.
If an answer contains outdated code, significantly lower your preference for it to avoid bias towards older answers with more votes. When comparing the five answers, start by selecting the most semantically relevant one.
If two answers are equally relevant, choose the one with higher popularity and continue ranking accordingly.
When evaluating the answers, compare all five and provide brief explanations. Ensure that your decision is not influenced by any biases and that the order of presentation does not affect your judgment.
The length of the answers should not impact your evaluation; strive to remain objective.
– User Question –
{question}
– Assistant 1's (Answer, Vote, Creation time) Start –
({answers[0]['body']}, {answers[0]['score']}, {answers[0]['time']})
– Assistant 1's (Answer, Vote, Creation time) End –
– Assistant 2's (Answer, Vote, Creation time) Start –
({answers[1]['body']}, {answers[1]['score']}, {answers[1]['time']})
– Assistant 2's (Answer, Vote, Creation time) End –
– Assistant 3's (Answer, Vote, Creation time) Start –
({answers[2]['body']}, {answers[2]['score']}, {answers[2]['time']})
– Assistant 3's (Answer, Vote, Creation time) End –
– Assistant 4's (Answer, Vote, Creation time) Start –
({answers[3]['body']}, {answers[3]['score']}, {answers[3]['time']})
– Assistant 4's (Answer, Vote, Creation time) End –
– Assistant 5's (Answer, Vote, Creation time) Start –
({answers[4]['body']}, {answers[4]['score']}, {answers[4]['time']})
– Assistant 5's (Answer, Vote, Creation time) End –
After providing short explanations, last only output the preferred order list of the five answers in the format, such as [2, 1, 4, 5, 3] :

Figure 12: Rules for labeling StaCoCoQA testing data, whether manually or AI-assisted, consider semantic relevance, popularity, and creation time, with a time-decay adjustment applied to popularity.

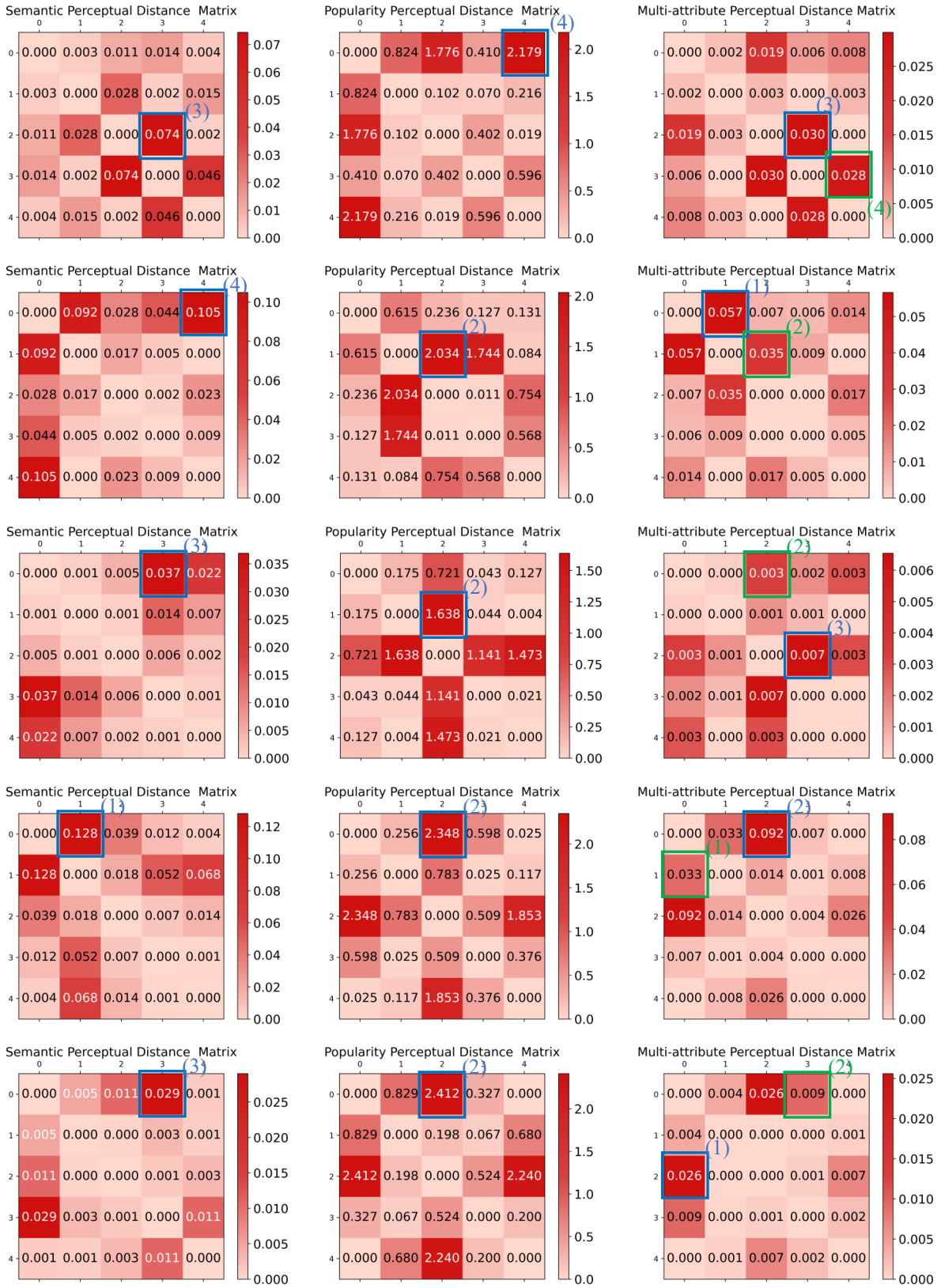
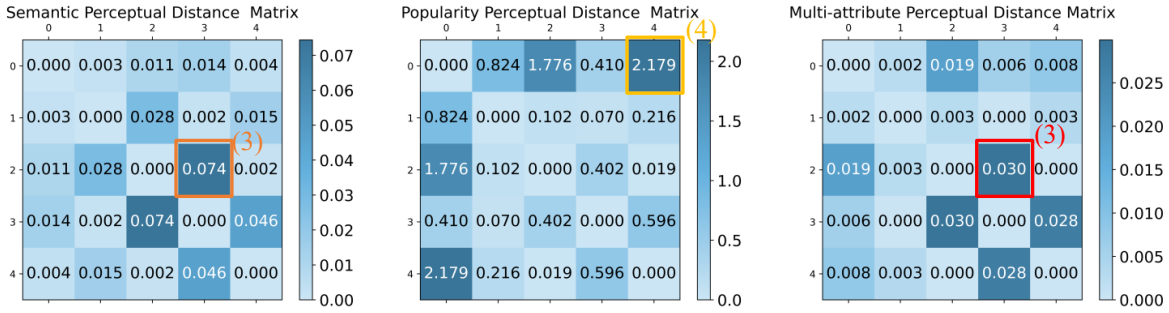
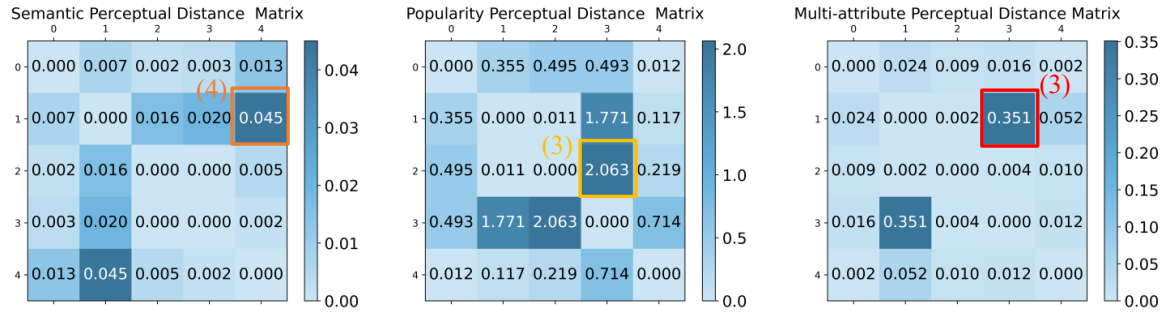


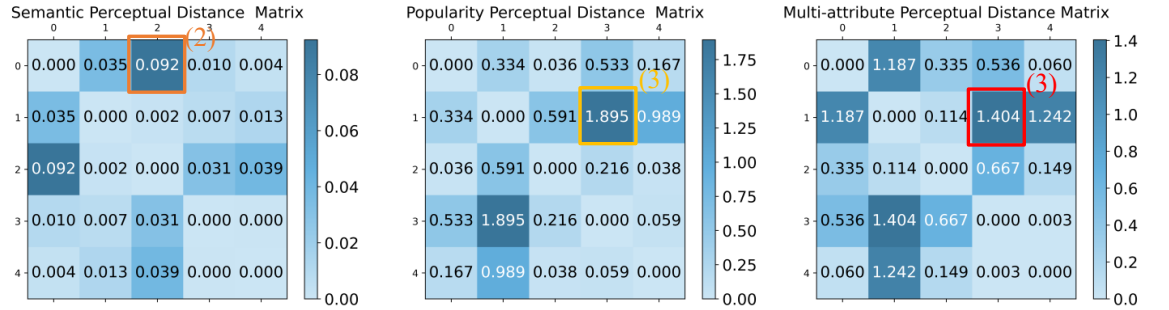
Figure 13: The visualization of Attribute-Perceptual Distance Factors (APDF) for diifferent selected samples having five candidates. The blue represents the alignment target of the corresponding APDF. The green indicates that the second alignment target is suboptimal compared to the blue one. We have three key findings: (1) The alignment of the Multi-attribute Perceptual Distance Matrix Δ_M could be the alignment target of the Semantic Perceptual Distance Matrix Δ_{Se}. (2) The alignment target of the Δ_M could also be the alignment target of the Popularity Perceptual Distance Matrix Δ_{Po}. (3) The alignment target of the Δ_M may neither be the alignment target of the Δ_{Se} nor the Δ_{Po}.



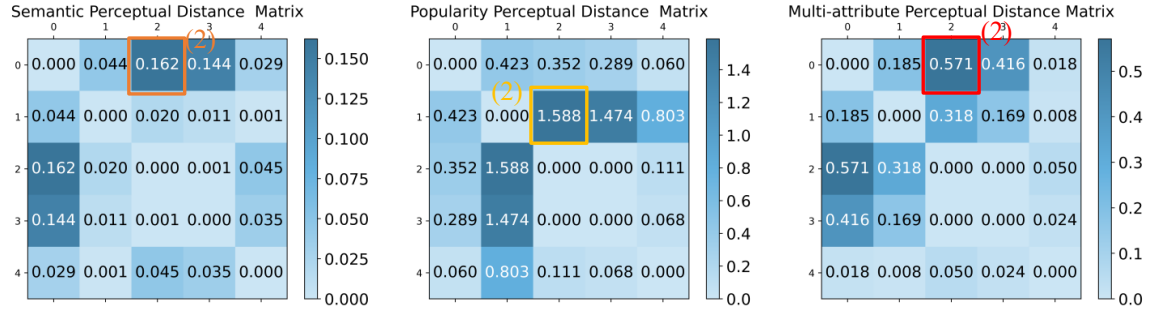
(a) The Visualization of Attribute-Perceptual Distance Factors (APDF) at Epoch 0



(b) The Visualization of Attribute-Perceptual Distance Factors (APDF) at Epoch 1



(c) The Visualization of Attribute-Perceptual Distance Factors (APDF) at Epoch 2



(d) The Visualization of Attribute-Perceptual Distance Factors (APDF) at Epoch 3

Figure 14: Visualization of the alignment target evolution for a sample throughout the training process. The orange represents the alignment target of the Semantic Perceptual Distance Matrix Δ_{Se}. The yellow represents the alignment target of the Popularity Perceptual Distance Matrix Δ_{Po}. The red represents the alignment target of the Multi-attribute Perceptual Distance Matrix Δ_M. We have two key findings. (1) At the same epoch, the alignment targets may differ across the Semantic Perceptual Distance Matrix Δ_{Se}, the Popularity Perceptual Distance Matrix Δ_{Po}, and the Multi-attribute Perceptual Distance Matrix Δ_M. (2) Across different epochs, the alignment targets for the same Attribute-Perceptual Distance Matrix may evolve.