ZEBRA: Leveraging Model-Behavioral Knowledge for Zero-Annotation Preference Dataset Construction

Anonymous ACL submission

Abstract

Recent efforts in LLM alignment have focused on constructing large-scale preference datasets via human or Artificial Intelligence(AI) annotators. However, such approaches rely on *instance-wise* supervision, incurring substantial annotation cost and limited interpretability. In this paper, we propose **ZEBRA**—a *model behavior-wise zero-annotation* framework that constructs preference data by leveraging model behavior knowledge derived from benchmark performances.

ZEBRA binarizes response pairs by evaluating the quality and similarity of their origin models, entirely bypassing instance-level annotation. This allows scalable, controllable, and cost-effective alignment data generation. Empirical results show that ZEBRA achieves alignment performance comparable to instance-supervised methods, despite requiring no manual or model-based labeling.

1 Introduction

Aligning large language models (LLMs) with human preferences is an essential step toward making them both useful and safe. A common way to achieve this is through instance-wise labeling, where pairs of model responses are compared one by one to see which is better. Well-known methods like Reinforcement Learning from Human Feedback (RLHFOuyang et al. (2022a)) and Artificial Inteligence(AI)-based labeling(RLAIFLee et al.) often use this strategy.

However, *instance-wise* labeling faces two major challenges. First, it is very costly, whether it involves human annotators or additional computational resources for LLM-based labeling(Zhang et al., 2024; Zheng et al., 2023). Second, it lacks a global view of the model's behavior(Ji et al., 2023). Since each response pair is judged in isolation, it is difficult to consider broader factors. For example, whether fluency should outweigh factual accuracy. Or whether a model's outputs are consistently aligned with certain policies(Wang et al., 2025). This can lead to labeling noise, mistakes, and limited interpretability.

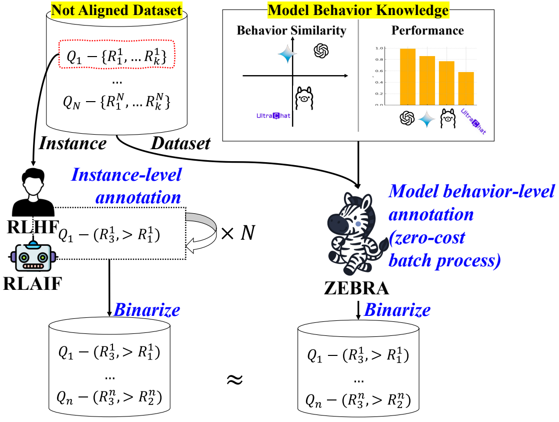


Figure 1: Human annotation(Ouyang et al., 2022a) and AI labeler-based RLAIF(Cui et al., 2023) use *instance-wise* annotations, while ZEro-annotation Behavior-based Response Alignment(ZEBRA) applies *model behavior-wise* annotation based on **model behavioral knowledge**, which captures proficiency and similarity across models. M is the model, R_i is the generated response from M_i about Instruction Q. N is the number of preference dataset and K is the number of models

To address these limitations, we propose a new preference binarization approach called **ZEro-annotation Behavior-based Response Alignment (ZEBRA)** (Figure 1). The main idea is: (1) extract each model's behavioral patterns from its past performance trajectories, (2) measure and compare these behaviors in terms of model strength or similarity, and (3) assign preferences at the *model* level rather than for each individual response pair.

We implement this idea through three key components. First, we define **Model Behavior Knowledge (MBK)** for LLMs (discuss in Section 4.1). Second, we propose a way to quantify and collect MBK from objective data sources such as benchmark performance. Third, we introduce three strategies—based on *superiority*, *similarity*, and a *hy-*

brid—to construct a binarization dataset in a zero-annotation, cost-free manner.

A major advantage of our approach is that it creates response pairs based on model superiority without any additional human or LLM labeling. By classifying models with higher benchmark scores as "positive" and those with lower scores as "negative," we can systematically label preferences throughout the dataset. This significantly reduces the cost of annotation and, because MBK visualizes each model's behavior pattern, increases the interpretability of the preference decisions.

Through extensive experiments, we show that **ZEBRA** achieves performance comparable to existing instance-wise labeling methods in the Ultrafeedback(Cui et al., 2023) dataset—*without* any extra labeling cost.

In summary, our contributions are as follows:

- We introduce ZEBRA, *a zero-annotation alignment framework* that determines preferences from quantified model-level behavior, bypassing instance-level supervision.

- We demonstrate that the *Model Behavior Knowledge (MBK)* from benchmark performance offers alignment signals comparable to instance-wise labeling.

- We empirically show that ZEBRA matches the performance of established methods such as RLHF and RLAIF, yet requires *no additional labeling cost*.

2 Preliminaries

2.1 Instance-Level Preference Construction

Most existing preference learning frameworks for LLM alignment—such as Reinforcement Learning from Human Feedback(RLHF, Ouyang et al. (2022a)) and AI-generated preference methods (RLAIF, Lee et al.)—rely on **instance-level pairwise supervision**. Given an instruction x, multiple candidate responses $\{r_1, r_2, ..., r_k\}$ are scored or ranked by either human annotators or automated scoring models. This generates preference tuples $(x, r_i \succ r_j)$, where r_i is preferred over r_j under some evaluation criteria (e.g., helpfulness, truthfulness, coherence).

The preference construction process typically involves:

- Generating multiple responses per instruction using different models or decoding strategies.

- Computing preference labels via either human judgment or model-based scoring (e.g., GPT-4(OpenAI, 2023)).

- Aggregating these labels into a pairwise dataset for alignment tuning.

2.2 Challenges of Instance-Level Supervision

While instance-level supervision has proven effective in aligning LLMs, it remains costly, noisy, and difficult to scale. Despite its popularity, instance-level supervision suffers from three core limitations:

1. **Costly evaluation**: Annotating or scoring each response pair requires substantial human or computational effort.

2. **Preference triviality**: When candidate responses differ significantly in quality, the preference label becomes trivial, contributing little to alignment learning.

3. **Instruction-level variance**: Difficulty and ambiguity in the instruction x can introduce noise into the preference signal, especially for automated labelers.

These limitations underscore the need for alternative approaches that construct preference signals without relying on per-instance scoring.

2.3 Motivation for Model Behavior-Level Preference

We propose that instead of relying on per-instance evaluation, one can leverage the **intrinsic capabilities of the response-generating models** themselves. As models exhibit differences in core competencies measurable via standardized benchmarks, we hypothesize this can replace instance-level annotations.

3 ZEro-annotation Behavior-based Response Alignment Framework

3.1 Instance-level Annotation vs. Model behavior-level Annotation

Traditional instance-level binarization relies heavily on detailed, provided by human or AI annotators(Zhang et al., 2024; Sharma et al., 2024). Each pairwise comparison demands significant effort and resources to maintain consistency and interpretability. Such an approach often results in annotation noise, limited scalability, and considerable expense.

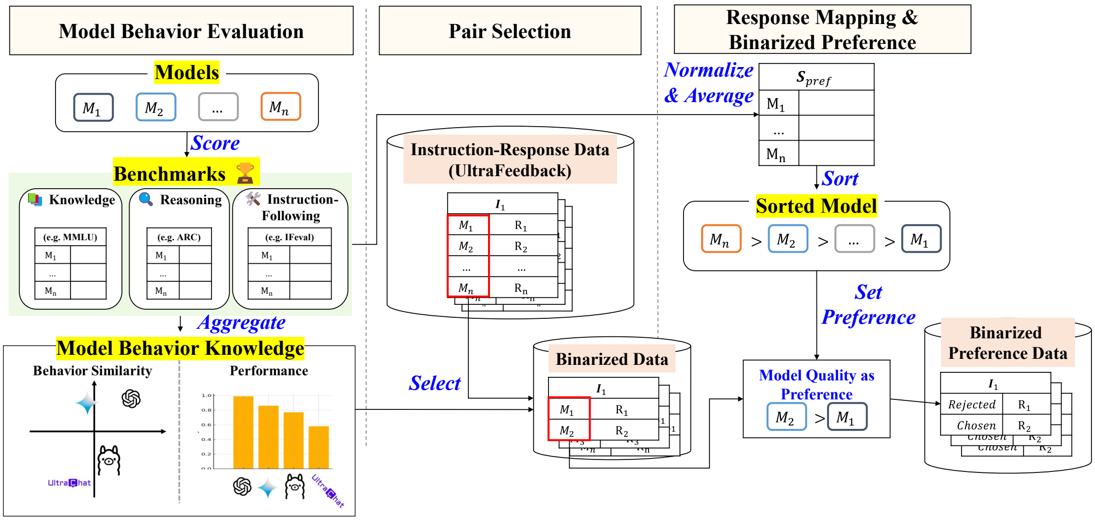


Figure 2: Editing process of ZEBRA framework. n is the number of models. $\{M_1, M_2, ..., M_n\}$ represents the total set of candidate models. $\{I_1, I_2, ..., I_n\}$ represents the instruction input of dataset. $\{R_1, R_2, ..., R_n\}$ represents the total responses from the candidate models. R_x and R_y are the chosen and rejected responses. S_{pref} score summarizes the model's benchmarks ability

In contrast, our proposed ZEBRA framework leverages intrinsic behavioral knowledge derived from model performance across various benchmarks, ZEBRA systematically matches and pairs responses. Responses from models with proven higher competencies form positive labels, while those from models with lower competencies become negative labels. This innovative model-level approach drastically reduces annotation costs, minimizes noise, and enhances scalability. Additionally, the behavior similarity-based matching provides explicit control over the difficulty and nuance of preference comparisons, leading to clearer and more meaningful alignment outcomes.

3.2 Model Behavior Knowledge

While most preference-learning pipelines focus on differences between individual responses, our approach highlights Model Behavior Knowledge (MBK)—the comprehensive record of each model's past behaviors and capabilities. We define MBK using two sets of metrics:

- **Superiority:** How much better (or worse) a model is compared to others, based on overall or task-specific proficiency.
- **Similarity:** How likely a model is to behave similarly to other models.

These metrics provide a principled basis for quantifying both a model's general strength and its behavioral proximity to its peers. Within a preference-learning pipeline, *superiority* functions as a global preference signal: when one model consistently outperforms another across standardized benchmarks, its responses are considered preferable overall.

Conversely, *behavioral similarity* facilitates the systematic construction of challenging comparison sets. When two models are behaviorally similar—for example, they exhibit comparable reasoning performance—their responses become difficult to distinguish. Training on such hard-to-distinguish pairs guides the preference learner to focus on subtle qualitative differences, resulting in more nuanced and robust alignment.

3.3 Model Behavior Evaluation using Benchmark Performances

To capture MBK in a practical, objective way, we rely on external benchmark performance data for each LLM. Many models are already evaluated across diverse, standardized tasks (e.g., reasoning, factual accuracy, instruction-following). We aggregate these benchmark scores to form each model's MBK profile.

Benchmark performance offers several advantages for extracting MBK:

- It provides *reliable metrics* for the core competencies of large language models.
- It enables *straightforward comparison and aggregation* across multiple models.
- Because benchmark scores are published and fixed at release, they impose no constraints on

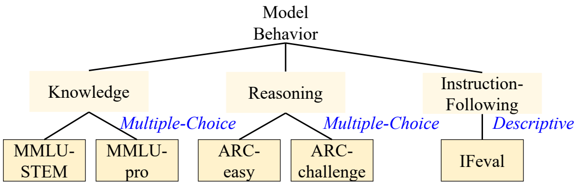


Figure 3: Selected benchmarks for evaluating the models behavior knowledge: knowledge, reasoning, and instruction-following capabilities

subsequent *data or model expansion*.

The list of benchmarks used in our analysis is provided in Figure 3.

Figure 4 illustrates how the behavioral similarities inferred from benchmark performance can reflect actual model similarities. For example, the LLaMA-2-13b model exhibits a pattern closely resembling that of its 7b counterpart. In contrast, WizardLM-7b demonstrates a markedly different behavioral trajectory.

We define a model's ability vector $v_i \in R^m$ across m benchmark tasks:

$$v_i = \left[s_i^{(1)}, s_i^{(2)}, \ldots, s_i^{(m)}\right],$$

where $s_i^{(b)}$ is the normalized score of model M_i on benchmark b, reflecting its relative capability in a specific behavioral dimension (e.g., knowledge, reasoning, instruction-following). These standardized behavior vectors serve as the foundation for both quality-based and similarity-based anchoring, enabling **zero-annotation binarization** of preference pairs without per-instance supervision.

3.4 Benchmark-based Model Behavior Quantification

To effectively binarize preference data, it is essential to quantify model behavior from two complementary perspectives: *behavior quality* and *behavior similarity*. ZEBRA leverages benchmark-derived measures of these aspects to systematically pair positive responses with suitable negative counterparts. By quantifying both the absolute competency of individual models and the relative similarity between models, ZEBRA ensures that each positive-negative pair captures meaningful contrasts in model capabilities, thus maximizing alignment informativeness.

Model Behavior Superiority (MB-SUP) quantifies the overall behavioral competency of model M_i as the aggregate of its normalized benchmark scores:

$$\text{MB-SUP}(M_i) = \frac{1}{m}\sum_{k=1}^{m} s_i^{(b)}. \qquad (1)$$

This scalar value serves as the ranking basis for constructing *Behavioral Superiority Anchors*.

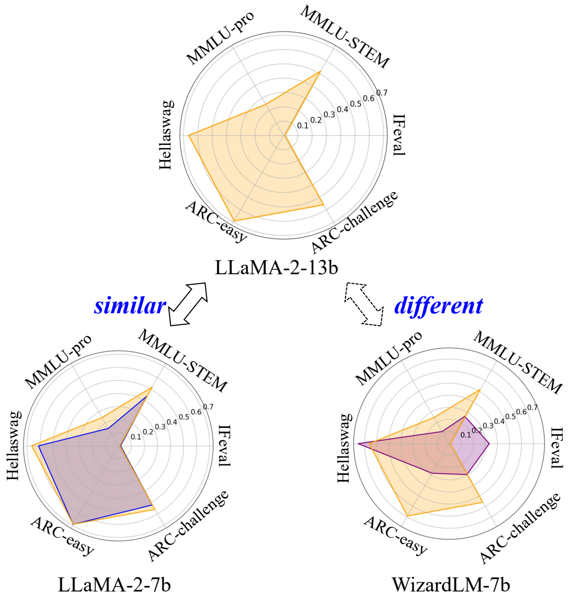


Figure 4: Hexagonal radar plots visualizing model ability profiles. Models judged as *similar* exhibit comparable benchmark performance trajectories, while *different* models show clearly divergent patterns.

Model Behavior Similarity (MB-SIM) between models M_i and M_j is defined as the similarity between their behavior vectors:

$$\text{MB-SIM}(M_i, M_j) = \text{similarity}(v_i, v_j).$$

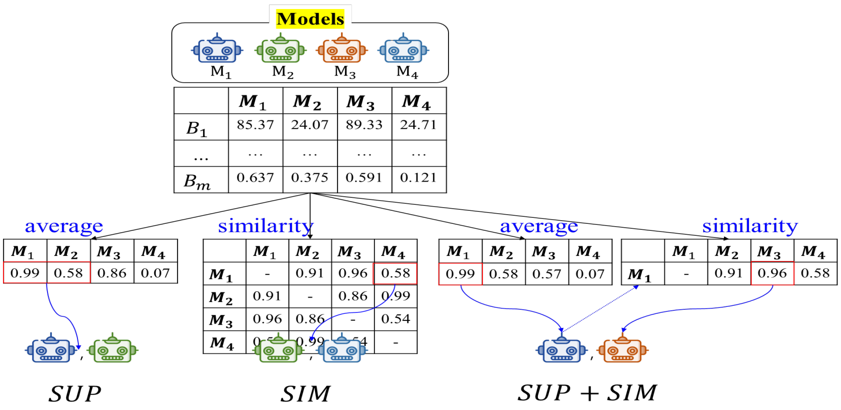
Higher values indicate stronger alignment in general-purpose capabilities, and MB-SIM serves as the criterion for selecting comparable model pairs in *Behavioral Similarity Anchoring*.

3.5 Strategy of Preference Binarization

ZEBRA introduces multiple strategies to systematically convert benchmark-derived model behaviors into binary preference pairs. Based on how MB-SUP and MB-SIM are utilized, our strategies can be clearly categorized as follows on Figure 5.

We detail each strategy below:

Strategy 1: Superiority-first Anchoring (SUP) In this strategy, we explicitly select the top-two models based on their MB-SUP scores: the highest-scoring model (top-1) and the second-highest-scoring model (top-2). Responses from the top-1 model serve as positive anchors, while responses



(a) Example of strategy SUP, SIM and SUP+SIM

Alias	Strategy
SUP	Superiority-first Anchoring
SIM	Similarity-first Anchoring
SUP+SIM	Hybrid (Superiority-first + similarity-second)

(b) Symbolic aliases of strategies

Figure 5: Anchoring strategies symbolic aliases and the example of strategy SUP, SIM and SUP+SIM with model and labeled preference.

from the top-2 model become negative counterparts. This approach emphasizes explicit quality distinctions, clearly defining superior responses and ensuring meaningful, informative alignment contrasts.

Strategy 2: Similarity-first Anchoring (SIM) Responses from models sharing similar behavioral patterns (high MB-SIM) are paired first. Within these pairs, the response from the model with higher MB-SUP is selected as the anchor (positive response). This strategy emphasizes behavioral similarity, enhancing nuanced alignment comparisons.

Strategy 3: Hybrid Anchoring (SUP+SIM) Model pairs are selected by simultaneously considering both MB-SUP and MB-SIM criteria. This balanced approach ensures each response pair reflects meaningful contrasts in model quality, while maintaining behavioral similarity for refined granularity.

These strategies enable ZEBRA to flexibly and effectively tailor preference construction according to the desired granularity, alignment objectives, and computational resources. Figure 5 represent the example of each strategy.

3.6 Zero-Annotation Preference Construction

Given a set of instructions $\mathcal{X}$ and a pool of response-generating models $\mathcal{M}$, the construction proceeds as follows:

1. **Model Behavior Evaluation:** Each model $M_i \in \mathcal{M}$ is benchmarked across m tasks to obtain model's ability vector v_i.

2. **Pair Selection:** Using a chosen strategy, a set of model pairs $\mathcal{P} = \{(M_i, M_j)\}$ is selected where MB-SIM$(M_i, M_j) \geq \tau$. we set τ at 0.1, as cosine similarity values below this threshold indicate a lack of meaningful similarity between models.

3. **Response Mapping:** For each instruction $x \in \mathcal{X}$, the corresponding responses $\{r_i, r_j\}$ from (M_i, M_j) are retrieved.

4. **Preference Assignment:** A binary label is assigned via:

$$\text{Pref}(r_i, r_j) = \begin{cases} 1 & \text{if } S_{\text{pref}}(M_i) > S_{\text{pref}}(M_j), \\ 0 & \text{otherwise.} \end{cases}$$

This pipeline constructs a binarized preference dataset at scale without any manual or per-instance scoring. Figure 2 shows the total process of the pipeline.

4 Experimental Setup

In this paper, we propose the ZEBRA framework. To validate the functionality of this framework and the characteristics of Alignment Tuning, we conducted experiments to address the following Research Questions (RQs):

- **RQ1**: Does MB-SIM represent the models' similarity?

- **RQ2**: Does strategies affect the binarization process?

- **RQ3**: Does ZEBRA reduce annotation cost and computational overhead compared to instance-level methods?

4.1 Alignment Dataset

For preference binarization, we utilized the UltraFeedback dataset (Cui et al., 2023), which includes diverse responses generated by both commercial and open-source language models. It provides multiple model responses, making it suitable for constructing a strong RLAIF baseline, for which we adopted the original score aggregation method. In contrast, ZEBRA uses only the response pool without referencing any scores, ensuring a clean comparison focused on the binarization strategy.

Model coverage. UltraFeedback contains outputs from 17 LLMs: GPT-4(OpenAI, 2023), GPT-3.5 Turbo(Ouyang et al., 2022b), and Bard(Waisberg et al., 2024). Additionally, several models from the Llama family were included, such as Llama-2 (7B, 13B, and 70B)-chat(Touvron et al., 2023), UltraLM-13B(Cui et al., 2024), WizardLM (7B, 13B, and 70B)(Xu et al., 2023), Vicuna-33B(Zheng et al., 2024), and Alpaca-7B(Taori et al., 2023). Beyond the Llama-based architectures, the dataset also features responses from other notable models, including Falcon-40B-instruct(Almazrouei et al., 2023), MPT-30B-chat(Team, 2023), StarChat-Beta(Tunstall et al., 2023), and Pythia-12B(Biderman et al., 2023).

Although the Ultrafeedback dataset contains results from UltraLM-65B(Cui et al., 2024), its performance could not be accurately assessed. To maintain the reliability of our evaluation, we excluded these results from the dataset composition.

4.2 Models

We fine-tuned and evaluated Llama-3.1-(3B, 8B)-Instruct(Dubey et al., 2024) and Qwen2.5-(3B, 8B)-Instruct(Yang et al., 2024) on the ZEBRA-binarized dataset. This setup enables cross-family comparison and quantifies the alignment gains from behavior-aware preference construction.

4.3 Training Algorithm for Alignment Tuning

We tested two different learning method for alignment tuning using the ZEBRA Framework:

- Supervised Fine-Tuning (SFT)
- Direct Preference Optimization (DPO) (Rafailov et al., 2024)

The implementation details are provided in Appendix A. The full source code, benchmark results, and preference binarization scripts will be released after publication via our project repository under the MIT License. The snapshot corresponding to this paper is version-tagged.

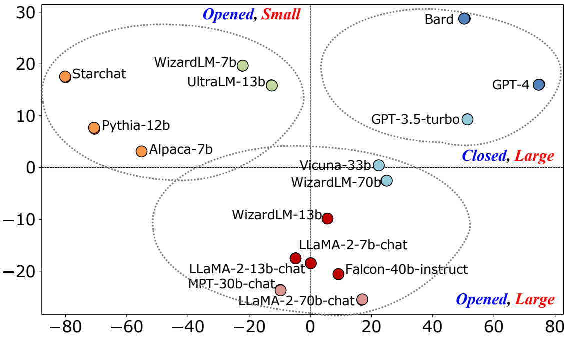


Figure 6: PCA-based visualization of relationships among models(Maćkiewicz and Ratajczak, 1993). The axes reflect similarity-based variance. Similar models are positioned closer to each other. Circle colors indicate clusters identified by hierarchical clustering.

5 Experimental Results

5.1 RQ1: Dataset Reconstruction using ZEBRA

The proposed ZEBRA framework enables the application of a unified Total Ranking map across the entire dataset, facilitating a structured and consistent preference mapping process. Utilizing this reconstructed preference dataset, we conducted model training while ensuring that each critic's binarized response was systematically incorporated. The effectiveness of this reconstructed dataset was assessed by evaluating the trained models on standardized benchmarks. The detailed results of these evaluations are presented in Appendix C.

For evaluation, we employed prediction-based assessment methodologies across all benchmark tasks. Specifically, for ARC, MMLU, and MMLU-Pro, we adapted the MMLU-Pro evaluation framework, modifying only the multiple-choice options to align with our dataset. For IFeval, we leveraged its native evaluation framework to ensure consistency in assessment.

To analyze the relationships between models, we computed SUP and SIM by normalizing evaluation results across six benchmark tasks. Figure 6 presents a Principal Component Analysis (PCA)(Maćkiewicz and Ratajczak, 1993) visualization of these relationships, illustrating distinct clustering patterns among models. Generally, smaller-scale models tend to cluster in the first quadrant, models with closer Model Behavior relationships in the second quadrant, while Llama-based models and models exceeding 10B parameters are predominantly distributed in the third and fourth quadrants. This distribution underscores the critical role

			Knowledge		Reasoning		Instruction-Following
Category	Strategy	Average	MMLU	MMLU-Pro	ARC-Easy	ARC-Challenge	IFeval
Baseline	RLAIF	**0.31**	**0.36**	**0.15**	0.40	**0.40**	0.28
ZEBRA(Ours)	SUP	**0.31**(-0.00)	0.33	**0.15**	0.40	0.37	**0.30**
	SIM	0.29(-0.02)	0.34	**0.15**	0.36	0.33	0.29
	SUP+SIM	0.29(-0.02)	0.30	0.14	**0.41**	0.39	0.23

Table 1: Performance Comparison Between Instance-wise Binarized Data (Baseline) and Model Behavior-wise Binarized Data (Ours). The baseline corresponds to instance-wise scored RLAIF (Cui et al., 2023). The highlighted cells indicate performance that is equal to or higher than the baseline. **Bold** text shows the best performance on the benchmarks.

Comparison	t	p_t	Mean Δ	p_{perm}
SUP vs BL	-0.3541	0.7282	-0.0066	0.7345
SIM vs BL	-0.4374	0.6680	-0.0047	0.7585
SUP+SIM vs BL	-1.4472	0.1684	-0.0296	0.2011

Table 2: Paired t-test and permutation test (10,000 shuffles) comparing each anchoring strategy with the baseline (BL).

of model training data, training algorithms, and scale in determining Model Behavior relationships among models.

We further verify that anchor pairs selected by SIM indeed yield semantically closer responses. Using a TF–IDF(Ramos, 2003), we obtain an average response-pair similarity of 0.4623 for the *most similar model pair* (MB-SIM ↑) versus 0.4129 for the *least similar model pair* (MB-SIM ↓). This 12% relative gap confirms that high-MB-SIM anchoring indeed surfaces finer-grained yet coherent preference signals.

These findings highlight the effectiveness of the ZEBRA framework in reconstructing preference datasets, providing a more structured and informative approach to model alignment.

5.2 RQ2: Performance of ZEBRA Binarization

To assess the effectiveness of ZEBRA binarization, we examined whether model performance can serve as an indicator of data quality. The results of this comparison are presented in table 1. Across all benchmark tasks, the Model Behavior-based scoring metric, SUP, demonstrates performance levels nearly equivalent to the instance-wise RLAIF method, with a minimal deviation of only 0.008.

Notably, for MMLU-Pro and IFeval, ZEBRA-based binarization even surpasses the RLAIF baseline by approximately 0.01, indicating that structured preference mapping via Model Behavior can yield competitive or superior alignment outcomes. Table 1 shows the results of the average scores for each methodology. The detailed results, including average scores for each methodology, are summarized in Appendix E.

Statistical significance. Table 2 reports paired t-tests and permutation tests (10,000 shuffles) that compare each anchoring strategy with the baseline instance-wise RLAIF data across the same 1 024 evaluation prompts. None of the strategies achieves statistical significance at α=0.05, although the hybrid SUP+SIM shows the largest absolute gain (-0.0296, p_t=0.17, p_{perm}=0.20). These numbers indicate that ZEBRA's zero-annotation binarization maintains baseline-level alignment quality (within 3 pp) without incurring per-instance scoring cost, even if a decisive improvement is not yet observed.

5.3 RQ3: Cost & Efficiency Analysis

Table 3 contrasts the *labeling cost* of ZEBRA with canonical RLHF and RLAIF pipelines. The standing out points is below: **Absolute cost gap.** LLM-annotated RLAIF corpora lower this to \$0.6–\$4 K by outsourcing each pairwise judgment to GPT-4, yet they *still purchase every label*. ZEBRA, by reusing benchmark leaderboards, pays **no** marginal cost (\$0) for preference construction. **Relative efficiency.** Normalised per comparison, GPT-4 labels (≈\$0.063) are ≈ ×10 cheaper than human labels (≈\$0.67), but ZEBRA is *orders of magnitude* cheaper than both because it dispenses with pairwise annotation altogether.

Cost matters only if quality survives. Despite a zero-dollar label budget, ZEBRA *matches or surpasses* RLAIF baseline (Table 1); the mean performance difference across six benchmarks is ≤ 0.02.

Consequently, ZEBRA offers a **cost-minimal, scalable, and annotation-free** route to high-quality preference data. Because the price of human labor or GPT-4 tokens scales linearly with data volume, traditional pipelines become progressively more expensive as models, tasks, and safety domains proliferate. ZEBRA decouples alignment from annotation cost: adding a new model needs no further labels, and incorporating an extra bench-

Category	Method	Pairs / Units	Unit Cost (USD)	Total Cost (USD)
Instance-wise RLAIF	UltraFeedback(Cui et al., 2023)	64,000	0.252	16,128
	Safer-Instruct(Shi et al., 2024)	10,254	0.063	646
	OpenHermesPreferences(Huang et al., 2024)	989,000	0.126	124,614
Model behavior-wise	ZEBRA (ours)	64,000	0	0

Table 3: Labeling-cost comparison between LLM-labeled RLAIF datasets, and our benchmark-table approach. cost is estimated assuming equivalent labor per rating.

mark simply augments the behavior matrix.

6 Conclusion

We presented **ZEBRA**, a zero-annotation framework that replaces *per-instance* preference labeling with *model-behavior knowledge* distilled from public benchmark tables. By pairing responses according to either behavioral superiority or similarity—or a hybrid of the two—ZEBRA constructs 64k high-quality preference pairs at zero marginal cost, completely eliminating the dominant expense in RLHF or RLAIF pipelines.

Experiments across 6 standardized benchmarks show that models fine-tuned with *SUP* achieve performance on par with, and sometimes better than, an RLAIF baseline. SIM and the hybrid SUP+SIM remain within a small, statistically insignificant margin of the baseline. These results confirm that benchmark-derived behavioral knowledge can serve as an effective proxy for preference supervision. Crucially, ZEBRA reduces marginal labeling cost to zero, reducing tens of thousands of dollars otherwise required for large-scale preference annotation.

Future work will explore richer behavioral axes such as safety, toxicity, and multilingual ability; investigate adaptive schedules that incorporate additional refined pairing strategies during training; and evaluate how well ZEBRA preferences correlate with explicit human judgments.

7 Related Work and Background

7.1 Alignment Tuning

Alignment tuning has become a central focus in enhancing LLMs to meet user expectations and ethical standards(Kumar et al., 2024). Various preference-based learning techniques, particularly reinforcement learning methods(Schulman et al., 2017; Rafailov et al., 2024; Hong et al., 2024), have been developed to facilitate this tuning process.

These methods rely on preference datasets, which typically contain pairs of responses generated by models based on given instructions, with each pair ranked according to human or model-based evaluations. Prominent alignment datasets like *Ultrafeedback(Cui et al., 2023)*, which gathers extensive human feedback, and *HH-RLHF(Bai et al., 2022)*, which uses human-annotated preferences, provide foundational resources for alignment. To minimize the reliance on labor-intensive data curation, recent automated approaches have been introduced to filter or regenerate preference data based on specific criteria, promoting data consistency and scalability. However, such approaches often lack nuanced control over data quality, as they fail to consider the role of the origin model's capabilities in shaping alignment effectiveness (Shi et al., 2024).

7.2 Limitations in Existing Preference Data Approaches

Effective preference data construction requires a clear, rigorous set of criteria to ensure alignment quality across generated response pairs. Common criteria, including *Reasoning*, *Truthfulness*, and *Instruction-Following*, guide the selection of data that aligns with key ethical and functional standards(Cui et al., 2023; Bai et al., 2022). High-performing models, capable of producing responses that meet these standards, are often evaluated using benchmarks like *ARC(Clark et al., 2018)*, *MMLU(Hendrycks et al., 2020)*, and the *Instruction-Following eval(Zhou et al., 2023)* suite, which assess a model's factual accuracy, reasoning ability, and compliance with instructions.

The *ZEBRA Framework* addresses this limitation by introducing a model behavior-level approach that emphasizes model compatibility in preference data selection. By curating preference pairs based on model behavior knowledge with similar core competencies in knowledge, reasoning, and instruction-following—the ZEBRA Framework enhances alignment coherence and ensures stable, high-quality data.

Limitation

In this paper, we demonstrated that ZEBRA can achieve performance comparable to existing RLAIF without requiring annotations for SFT and DPO. However, due to resource and time constraints, we were unable to validate our approach across a broader range of alignment tuning techniques. Further evaluation is needed for methods.

Additionally, our evaluation primarily focused on fundamental abilities such as Reasoning, Knowledge, and Instruction-Following. However, we did not assess ZEBRA's performance on other important values, including factuality and ethical standard. Future work should incorporate evaluations reflecting these aspects.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. 2024. Ultrafeedback: Boosting language models with scaled ai feedback. In *Forty-first International Conference on Machine Learning*.

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.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, Florida, USA. Association for Computational Linguistics.

Shengyi Costa Huang, Agustín Piqueres, Kashif Rasul, Philipp Schmid, Daniel Vila, and Lewis Tunstall. 2024. Open hermes preferences. `https://huggingface.co/datasets/argilla/OpenHermesPreferences`.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Anurakt Kumar, Divyanshu Kumar, Jatan Loya, Nitin Aravind Birur, Tanay Baswa, Sahil Agarwal, and Prashanth Harshangi. 2024. Sage-rt: Synthetic alignment data generation for safety evaluation and red teaming. *Preprint*, arXiv:2408.11851.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. In *Forty-first International Conference on Machine Learning*.

Andrzej Maćkiewicz and Waldemar Ratajczak. 1993. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

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

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.

Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries.

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

Archit Sharma, Sedrick Keh, Eric Mitchell, Chelsea Finn, Kushal Arora, and Thomas Kollar. 2024. A critical evaluation of ai feedback for aligning large language models. *arXiv preprint arXiv:2402.12366*.

Taiwei Shi, Kai Chen, and Jieyu Zhao. 2024. Safer-instruct: Aligning language models with automated preference data. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7636–7651, Mexico City, Mexico. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

MosaicML NLP Team. 2023. Introducing mpt-30b: Raising the bar for open-source foundation models. Accessed: 2023-06-22.

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, Nathan Lambert, Nazneen Rajani, Edward Beeching, Teven Le Scao, Leandro von Werra, Sheon Han, Philipp Schmid, and Alexander Rush. 2023. Creating a coding assistant with starcoder. *Hugging Face Blog*. Https://huggingface.co/blog/starchat.

Ethan Waisberg, Joshua Ong, Mouayad Masalkhi, Nasif Zaman, Prithul Sarker, Andrew G Lee, and Alireza Tavakkoli. 2024. Google's ai chatbot "bard": a side-by-side comparison with chatgpt and its utilization in ophthalmology. *Eye*, 38(4):642–645.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2025. Helpsteer2-preference: Complementing ratings with preferences. *Preprint*, arXiv:2410.01257.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Michael JQ Zhang, Zhilin Wang, Jena D. Hwang, Yi Dong, Olivier Delalleau, Yejin Choi, Eunsol Choi, Xiang Ren, and Valentina Pyatkin. 2024. Diverging preferences: When do annotators disagree and do models know? *Preprint*, arXiv:2410.14632.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Implementation Details for Model Training

The model was trained for a single epoch using bfloat16 (bf16) quantization, which optimizes memory efficiency while preserving numerical precision. The training configuration incorporated the following hyperparameters: a per-device batch size of 6, gradient accumulation steps set to 4, and a learning rate of 5×10^{-5}, with 500 warm-up steps to facilitate stable convergence.

Training was conducted on an L40 4-GPU setup, leveraging an optimized deep learning framework to enhance computational efficiency. The training pipeline focused on performance optimization through checkpointing and logging, without intermediate model evaluation during training.

B Prompt template for Evaluation Benchmarks

To determine behavior knowledge between models, this paper evaluates benchmark performance and

Prompt for Benchmark Evaluation

Q: {Instruction}
Options:
(A) {Option 1}
(B) {Option 2}
(C) {Option 3}
(D) {Option 4}
Output:

Figure 7: Evaluation template for multiple-choice benchmark evaluation. The descriptive benchmark (e.g., IFeval) was evaluated using the same template, excluding the option part.

compares the similarity of these numerical results. The benchmark performance of all 17 models considered in this study is presented in Table 7. For the start and end tokens, the template recommended in the paper was used.

C Evaluation Model Benchmark Performance

To determine behavior knowledge relationships between models, this paper evaluates benchmark performance and compares the similarity of these numerical results. The benchmark performance of all 17 models considered in this study is presented in Table 4. The actual calculation example for $MB - SUP$ and $MB - SIM$ can be found in Figure 10.

Due to the unavailability of Bard, its performance metrics have been substituted with those of Gemini-1.5-Flash.

D Benchmark Performance Representation

What benchmark represent the model capability: knowledge, instruction-following, and reasoning? Models exhibit distinct similarity patterns based on their capabilities, with these patterns varying across different evaluation metrics. Notably, model behavior knowledge is influenced by factors such as model scale and training methodology, leading to variations in clustering behavior across different capability dimensions.

Analysis Framework To systematically investigate these relationships, we analyzed model similarities across three key capability dimensions:

- **Knowledge-Based Tasks**: Unlike reasoning capability, clustering in instruction-following tasks is more strongly aligned with *model families* rather than size. This indicates that training methodology and architectural choices exert a greater influence on instruction adherence.

- **Reasoning Capability**: Models tend to cluster primarily based on parameter count, suggesting that *model size* plays a dominant role in shaping reasoning performance.

- **Instruction-Following (IF) Tasks**: A hierarchical influence pattern emerges, where:
  - At the initial hierarchy, the *model family* is the primary determinant.
  - At the higher hierarchy, the *model size* becomes a stronger predictor of performance.

Figure 11 visualizes these relationships through dendrograms, illustrating the hierarchical clustering patterns that emerge across different capability dimensions.

Figure 8 shows the frequency of each model being selected as the positive or negative under each strategy. Overall, larger and more familiar models tend to be chosen more frequently.

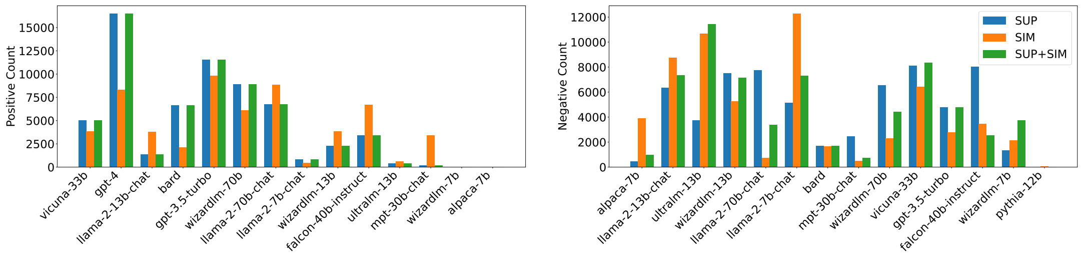
E Total Performance

Table 5 provides an aggregated view of the overall performance of each model across all benchmark datasets. The total performance scores were computed by averaging the normalized scores across the selected evaluation metrics, offering a holistic comparison of model capabilities.

F Model Size and ZEBRA Strategies

The impact of different Strategies varies significantly with model size, influencing how models learn from preference data. To investigate this relationship, we conducted experiments using models from the same family but with different parameter counts, specifically comparing small models (3B parameters) and larger models (7-8B parameters). Our analysis reveals a clear pattern in how model size determines the optimal strategy.

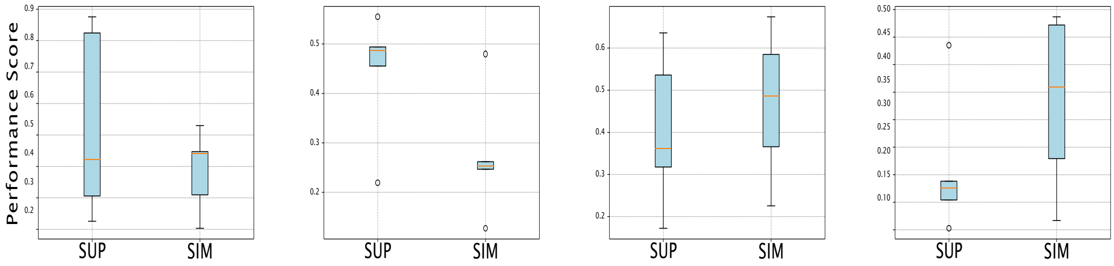
Smaller models (3B) exhibit superior performance when trained using the SUP algorithm, which prioritizes learning from response quality



(a) Positive

(b) Negative

Figure 8: Selected Model Frequency for the positive and negative pairs.



(a) Qwen-2.5-3B

(b) Llama-3.1-3B

(c) Qwen-2.5-7B

(d) Llama-3.1-8B

Figure 9: Visualization of performance across different models. This represents the performance when using the DPO training method. For smaller models, using SUP results in better performance, whereas for larger models, SIM yields better performance

within the binarized dataset. In contrast, larger models (7-8B) achieve better results with the SIM algorithm, suggesting that response similarity becomes increasingly important as model size grows.

This trend indicates that model size fundamentally influences how different models leverage preference data. While smaller models benefit more from explicit learning based on absolute quality differences, larger models demonstrate greater sensitivity to the nuanced relationships between similar responses in the training data. This relationship is visualized in Figure 9, illustrating the distinct learning behaviors observed across different model scales.

Benchmark Performances

Normalize(min-max)

Model	IFeval	MMLU -STEM	MMLU -pro	Hellaswag	ARC -Easy	ARC -Challenge
GPT-4	0.939	1.000	1.000	1.000	1.000	1.000
GPT-3.5-turbo	0.743	0.724	0.498	0.856	0.952	0.847
Bard	1.000	0.754	0.912	0.852	0.847	0.779
Llama-2-7b-chat	0.017	0.439	0.164	0.612	0.693	0.521
Llama-2-13b-chat	0.011	0.519	0.261	0.664	0.691	0.561
Llama-2-70b-chat	0.000	0.705	0.496	0.760	0.819	0.671
UltraLM-13b	0.473	0.380	0.146	0.424	0.510	0.446
WizardLM-7b	0.333	0.261	0.128	0.755	0.283	0.296
WizardLM-13b	0.151	0.426	0.098	0.801	0.703	0.530
WizardLM-70b	0.390	0.433	0.296	0.833	0.800	0.720
Vicuna-33b	0.440	0.623	0.217	0.717	0.811	0.635
Alpaca-7b	0.100	0.184	0.055	0.000	0.331	0.283
Falcon-40b-instruct	0.007	0.682	0.043	0.787	0.749	0.545
MPT-30b-chat	0.102	0.394	0.165	0.013	0.881	0.706
Starchat	0.065	0.221	0.000	0.025	0.000	0.000
Pythia-12b	0.010	0.000	0.006	0.028	0.095	0.260

→ ***MB − SUP***

Calculate MB − SIM

MB − SIM

	GPT-4	GPT-3.5 -turbo	Bard	Llama-2 -7b -chat	Llama-2 -13b -chat	Llama-2 -70b -chat	UltraL M -13b	Wizard LM-7b	Wizard LM-13b	Wizard LM-70b	Vicuna -33b	Alpaca -7b	Falcon -40b -instruct	MPT -30b -chat	Starchat	Pythia -12b
GPT-4	1.000	0.983	0.994	0.869	0.890	0.911	0.956	0.869	0.872	0.942	0.949	0.803	0.832	0.767	0.546	0.589
GPT-3.5 -turbo	0.983	1.000	0.973	0.915	0.922	0.919	0.990	0.899	0.928	0.979	0.987	0.846	0.886	0.813	0.517	0.645
Bard	0.994	0.973	1.000	0.822	0.843	0.864	0.951	0.864	0.836	0.921	0.926	0.769	0.779	0.726	0.518	0.548
Llama-2 -7b-chat	0.869	0.915	0.822	1.000	0.996	0.978	0.888	0.851	0.985	0.962	0.957	0.830	0.984	0.845	0.422	0.679
Llama-2 -13b-chat	0.890	0.922	0.843	0.996	1.000	0.992	0.885	0.855	0.977	0.960	0.956	0.818	0.979	0.831	0.455	0.663
Llama-2 -70b-chat	0.911	0.919	0.864	0.978	0.992	1.000	0.871	0.829	0.947	0.942	0.941	0.813	0.954	0.827	0.483	0.634
UltraLM -13b	0.956	0.990	0.951	0.888	0.885	0.871	1.000	0.890	0.913	0.963	0.981	0.852	0.868	0.807	0.533	0.643
WizardLM- 7b	0.869	0.899	0.864	0.851	0.855	0.829	0.890	1.000	0.920	0.925	0.903	0.566	0.856	0.535	0.439	0.479
WizardLM -13b	0.872	0.928	0.836	0.985	0.977	0.947	0.913	0.920	1.000	0.977	0.968	0.776	0.978	0.778	0.419	0.643
WizardLM -70b	0.942	0.979	0.921	0.962	0.960	0.942	0.963	0.925	0.977	1.000	0.985	0.822	0.930	0.811	0.406	0.694
Vicuna-33b	0.949	0.987	0.926	0.957	0.956	0.941	0.981	0.903	0.968	0.985	1.000	0.857	0.945	0.833	0.536	0.647
Alpaca-7b	0.803	0.846	0.769	0.830	0.818	0.813	0.852	0.566	0.776	0.822	0.857	1.000	0.783	0.990	0.418	0.785
Falcon-40b -instruct	0.832	0.886	0.779	0.984	0.979	0.954	0.868	0.856	0.978	0.930	0.945	0.783	1.000	0.788	0.529	0.605
MPT-30b -chat	0.767	0.813	0.726	0.845	0.831	0.827	0.807	0.535	0.778	0.811	0.833	0.990	0.788	1.000	0.335	0.798
Starchat	0.546	0.517	0.518	0.422	0.455	0.483	0.533	0.439	0.419	0.406	0.536	0.418	0.529	0.335	1.000	0.021
Pythia-12b	0.589	0.645	0.548	0.679	0.663	0.634	0.643	0.479	0.643	0.694	0.647	0.785	0.605	0.798	0.021	1.000

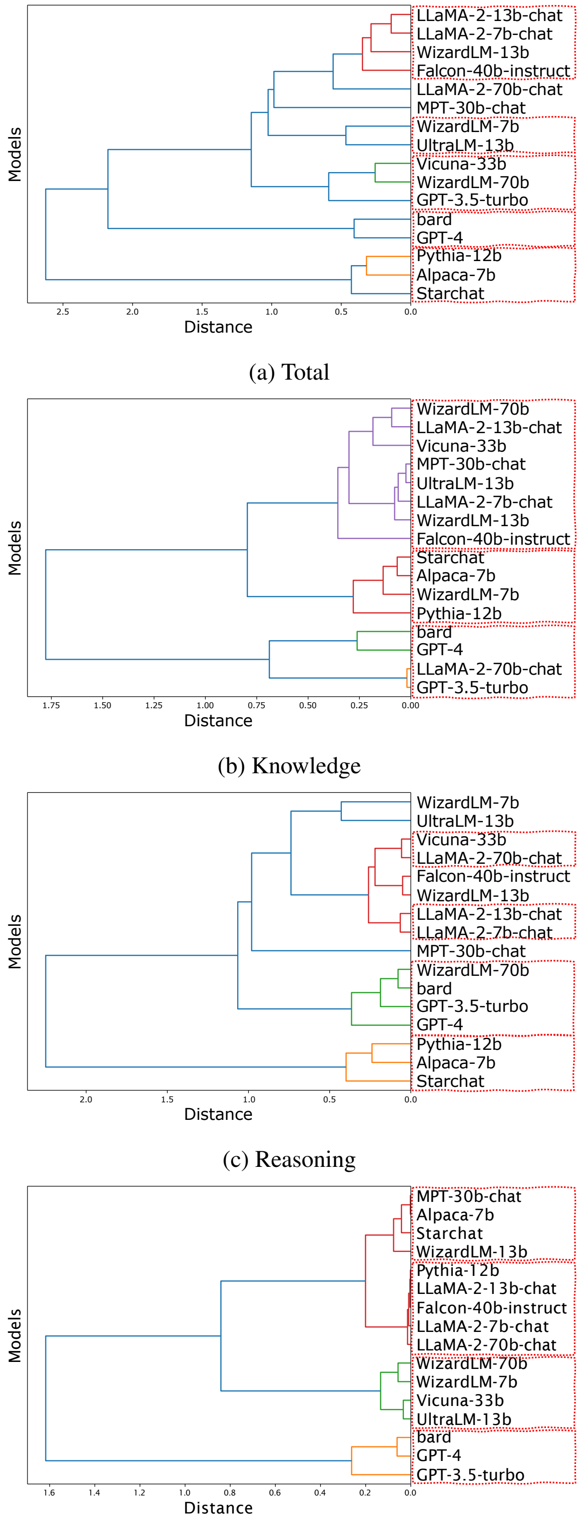
Figure 10: The process of calculating $MB - SUP$ and $MB_{S}IM$. Benchmark performance is normalized using min-max normalization, and the overall SIM is performed using cosine similarity.

Model	IFeval	MMLU-STEM	MMLU-pro	Hellaswag	ARC-easy	ARC-Challenge	Average Score
GPT-4	85.37	86.40	0.64	95.30	96.63	96.40	0.99
GPT-3.5-turbo	72.54	70.00	0.38	85.00	92.80	83.02	0.77
bard	89.33	71.80	0.59	84.70	84.43	77.13	0.86
Llama-2-7b-chat	25.19	53.10	0.20	67.50	72.14	54.61	0.41
Llama-2-13b-chat	24.82	57.80	0.25	71.20	72.05	58.02	0.45
Llama-2-70b-chat	24.07	68.90	0.38	78.10	82.20	67.66	0.58
UltraLM-13b	54.92	49.58	0.19	54.00	57.58	48.04	0.40
WizardLM-7b	45.83	42.50	0.18	77.70	39.48	34.90	0.34
WizardLM-13b	33.92	52.30	0.17	81.00	72.94	55.38	0.45
WizardLM-70b	49.51	52.70	0.27	83.30	80.68	71.93	0.58
Vicuna-33b	52.76	64.00	0.23	75.00	81.57	64.51	0.57
Alpaca-7b	30.58	37.92	0.15	23.60	43.31	33.79	0.16
Falcon-40b-instruct	24.54	67.50	0.14	80.00	76.60	56.70	0.47
MPT-30b-chat	30.70	50.40	0.20	24.53	87.12	70.73	0.38
Starchat	28.30	40.12	0.12	25.40	16.96	9.07	0.05
Pythia-12b	24.71	27.00	0.12	25.60	24.49	31.80	0.07

Table 4: Benchmark Scores for Trained Models. Multiple-choice benchmarks (MMLU-STEM, HellaSwag, ARC-Easy, and ARC-Challenge) are evaluated based on accuracy. IFeval and MMLU-Pro are assessed using its own metric. The average score is computed after min-max normalization.

Benchmark	Model		Baseline	SUP	SIM	SUP+SIM
MMLU-STEM	Llama-3.1-3B	SFT	0.2331	**0.2679**	0.2434	0.2200
		DPO	0.3130	**0.4868**	0.2528	0.3940
	Llama-3.1-8B	SFT	**0.2800**	0.2460	0.2660	0.2240
		DPO	**0.4902**	0.4349	0.4867	0.2880
	Qwen2.5-3B	SFT	**0.2760**	0.2200	0.2460	0.2500
		DPO	**0.4706**	0.4212	0.4400	0.4580
	Qwen2.5-7B	SFT	**0.2800**	0.2280	0.2620	0.2780
		DPO	**0.5120**	0.3620	0.4860	0.2800
MMLU-pro	Llama-3.1-3B	SFT	0.1189	0.1230	0.1045	**0.1332**
		DPO	0.1455	**0.2193**	0.1270	0.1516
	Llama-3.1-8B	SFT	0.1004	0.1148	0.1045	**0.1311**
		DPO	**0.1168**	0.1025	**0.1168**	0.1107
	Qwen2.5-3B	SFT	0.1025	0.0840	**0.1230**	0.0820
		DPO	**0.2275**	0.2254	0.2029	0.2111
	Qwen2.5-7B	SFT	0.0861	0.1762	**0.2275**	0.1352
		DPO	**0.2377**	0.1721	0.2254	0.1700
IFeval	Llama-3.1-3B	SFT	**0.2494**	0.2410	0.2490	0.2206
		DPO	0.3765	**0.4940**	0.4796	0.3033
	Llama-3.1-8B	SFT	0.2494	**0.4210**	0.2470	0.2218
		DPO	0.2421	0.1882	0.2292	**0.2494**
	Qwen2.5-3B	SFT	**0.2190**	0.2134	0.2122	0.2050
		DPO	**0.3633**	0.3058	0.3094	0.1715
	Qwen2.5-7B	SFT	**0.2290**	0.2134	0.2122	0.2083
		DPO	0.3321	0.3177	**0.3657**	0.2407
ARC-easy	Llama-3.1-3B	SFT	**0.2660**	0.2460	0.2000	0.2340
		DPO	0.6111	0.5547	0.2618	**0.6679**
	Llama-3.1-8B	SFT	0.2180	0.2330	**0.2400**	0.2360
		DPO	0.2176	0.1540	**0.4720**	0.0680
	Qwen2.5-3B	SFT	0.2410	**0.2560**	0.2480	**0.2560**
		DPO	0.8497	**0.8754**	0.5295	0.8157
	Qwen2.5-7B	SFT	0.2550	0.2260	0.2280	**0.2900**
		DPO	0.5700	0.6359	0.6738	**0.7134**
ARC-challenge	Llama-3.1-3B	SFT	**0.2556**	0.2492	0.2266	0.2019
		DPO	0.5122	0.4551	0.2466	**0.5712**
	Llama-3.1-8B	SFT	0.2320	0.2297	**0.2761**	0.2268
		DPO	**0.5463**	0.1763	0.3596	0.2946
	Qwen2.5-3B	SFT	**0.2645**	0.2483	0.2227	0.2343
		DPO	0.7398	**0.8241**	0.4470	0.7078
	Qwen2.5-7B	SFT	0.2343	**0.2552**	0.2483	0.2390
		DPO	0.4432	0.5358	**0.5847**	0.3870

Table 5: Model Performance Comparisons on Knowledge, Instruction-Following, and Reasoning-Related Tasks. The baseline is Instance-wise RLAIF (Cui et al., 2023). "Llama 3.1" refers to the Llama-3.1-Instruct series, and "Qwen-2.5" refers to the Qwen2.5-Instruct series. The training methods include Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO). The **bold** text indicates the best performance for each model and training method.



(a) Total

(b) Knowledge

(c) Reasoning

(d) Instruction-following

Figure 11: Dendrogram for Evaluation Categories: Knowledge, Reasoning, and Instruction-Following. Clusters exceeding a specific distance threshold (0.4) are highlighted in different colors. Models deemed similar share the same color line. The red dotted line shows the major groups of models in the dendrogram.