A Systematic Analysis of Base Model Choice for Reward Modeling

Anonymous ACL submission

Abstract

Reinforcement learning from human feedback (RLHF) and, at its core, reward modeling have become a crucial part of training powerful large language models (LLMs). One commonly overlooked factor in training high-quality reward models (RMs) is the effect of the base model, which is becoming more challenging to choose given the rapidly growing pool of LLMs. In this work, we present a systematic analysis of the effect of base model selection on reward modeling performance. Our results show that the performance can be improved by up to 14% compared to the most common (*i.e.,* default) choice. Moreover, we showcase the strong statistical relation between some existing benchmarks and downstream performances. We also demonstrate that the results from a small set of benchmarks could be combined to boost the model selection (+18% on average in the top 5-10). Lastly, we illustrate the impact of different post-training steps on the final performance and explore using estimated data distributions to reduce performance prediction error.

1 Introduction

Reinforcement learning from human feedback (RLHF) (Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022) has been a critical part of recent advancements in large language models (LLMs) such as OpenAI's O1 (OpenAI, 2024), Anthropic's Claude (Anthropic, 2024), and Google's Gemini (Gemini Team, 2023). At the core of RLHF methods, Reward Models (RMs) are used to guide the LLM (*i.e.,* policy) training by scoring generated responses (Schulman et al., 2017; Ahmadian et al., 2024). Most commonly, RMs are evaluated on RewardBench[1] (Lambert et al., 2024b), consisting of 2985 binary preference tasks, 23 subtasks, and four subcategories. The RewardBench leaderboard reflects a bias toward a single model family, with

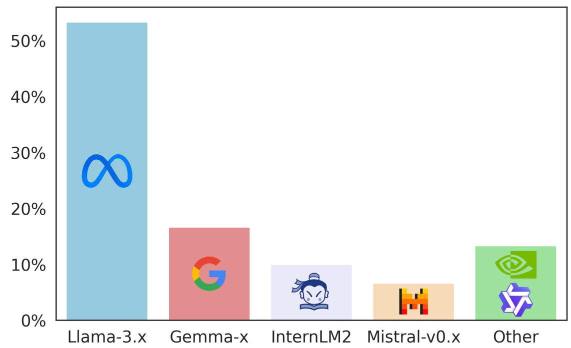


Figure 1: **Ratio of the base models used in the top 30 entries of RewardBench (Dec 2024).** Almost all the entries are trained on top of a small set of base models (*e.g.,* Llama-3.x models comprise 50% of the entries).

more than 50% of the top 30 entries (see Figure 1) built on top of a Llama-3.x model (Dubey et al., 2024) However, relying on a single model family without exploration is inherently suboptimal, regardless of Llama-3.x models' quality.

Considering this suboptimality, we hypothesize that the base model is a critical hyperparameter that substantially impacts the downstream performance. To test this hypothesis, we compare 40 popular models across various sizes and families (see Appendix C for more details). Our experiments show that replacing the popular base model (*i.e.,* LLama-3.x) with the best model of similar size leads to gains ranging from 3% to 14%. While these results prove our hypothesis, running such a search over the plethora of available models is extremely expensive. This obstacle inspires the need for robust approaches that could either limit the search perimeter or help us make a selection apriori. However, the criteria for selecting a model apriori are often unclear and multifaceted.

Prior works in RLHF (Stiennon et al., 2020; Gao et al., 2023a) have examined the relation between the model size and performance. Moreover, recent works (Ruan et al., 2024; Polo et al., 2024) have

[1] allenai/reward-bench

used compute metrics (*e.g.,* training tokens) and simple capabilities measured by standard benchmarks (*e.g.,* MMLU (Hendrycks et al., 2021)) to predict emergent capabilities of LLMs. Inspired by these works, we use these features to systematically analyze the base models to identify core capabilities and attributes that yield high-quality RMs. Our experiments show that while performances on many benchmarks and reward modeling have strong statistical correlations, they are insufficient for the broader model selection problem. Moreover, we show significant improvements (+18% on average in the top 5-10) can be gained over any single benchmark-based selection, only using a small subset of benchmarks.

While our analysis covers various elements, it does not investigate the effect of different training stages of a model, which have grown in numbers with recent advancements. Hence, we separately investigate the pre-training and post-training stages, relying on publicly available intermediate checkpoints (Lambert et al., 2024a). For the post-training stage, we demonstrate the positive impact of the supervised fine-tuning (SFT) stage (+15.5%) while showcasing the negative effect of the following alignment steps (3-5% drop). For the pre-training stage, we focus on estimating (Bakman et al., 2024) and analyzing the data composition, which has emerged as a key driving factor in recent developments (Abdin et al., 2024a,b; Yang et al., 2024). Our experiments show estimated distributions' variability across model families, which we use to reduce our regression model's error (+1.5%).

To summarize, our contributions are as follows:

- We showcase the significance of the base model choice, which could improve upon the most common (*i.e.,* default) choice up to 14% in a size-controlled setting.
- We analyze the statistical relation between performances on standard benchmarks and reward modeling, showcasing strong correlations (Pearson ≥ 0.8) on many while illustrating their shortcoming in model selection (*i.e.,* small overlap on top models)
- We show a simple performance prediction regression model based on benchmarks' results, when employed for model selection, can achieve +18% overlap on average over the top 5-10, compared to the benchmark with the highest correlation.
- We showcase the positive impact of the post-training stages, especially SFT, achieving up to +15.5% gains on publicly available models. Moreover, we expose the negative impact of the standard post-SFT alignment steps, leading to a 3-5% performance drop.
- We exhibit the potential of using estimated data distributions, which improves our regression model's performance by +1.5%.

2 Related Work

Reward Modeling Recently, there has been a lot of effort in crafting better training datasets (Liu et al., 2024a; Wang et al., 2024c) and improving training architectures (Dorka, 2024; Lou et al., 2024; Zhang et al., 2024b; Wang et al., 2024a). However, the core objective for reward modeling still revolves around two main approaches: Bradley-Terry w/ Binary Preferences (Ziegler et al., 2019; Bradley and Terry, 1952) and Regression w/ Multi-Attribute Scores (Wang et al., 2024e) (see Section 3 for more details). For datasets, RMs are commonly trained on labeled preference datasets such as UltraFeedback (Cui et al., 2024), HelpSteer2 (Wang et al., 2024d), and Magpie (Xu et al., 2024).

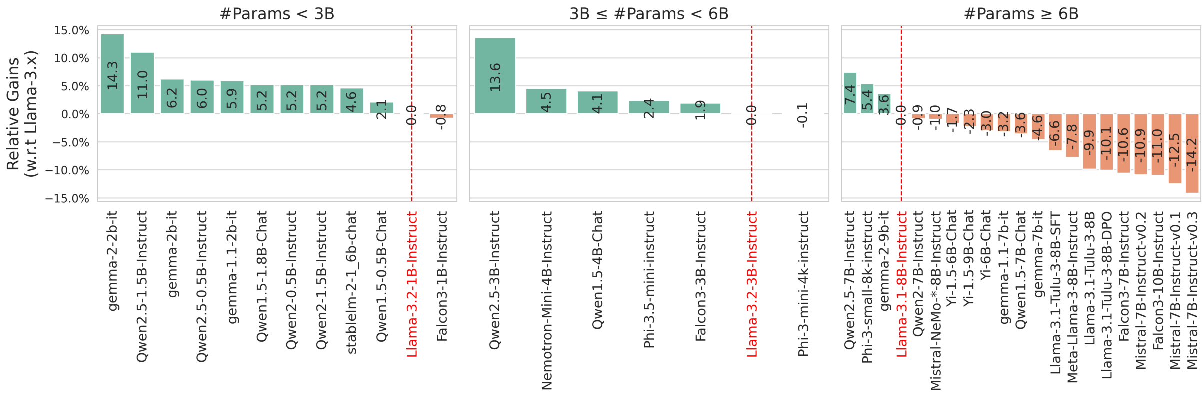
Reward Model Evaluation Until recently, one of the biggest challenges of training RMs has been evaluating the trained models in isolation. The lack of test sets in the released datasets made evaluation difficult without going through the highly costly policy training step. To overcome this issue, recent works (Lambert et al., 2024b; Liu et al., 2024c; Gureja et al., 2024) have introduced standardized benchmarks for evaluating these models. Among these benchmarks, RewardBench (Lambert et al., 2024b) is the most popular, with more than 150 entries at the time of writing this article.

3 Reward Modeling

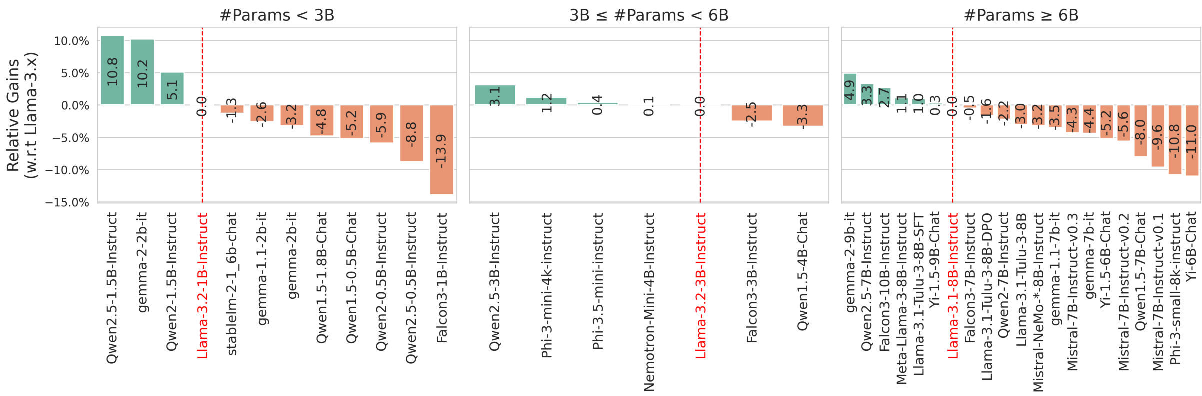
3.1 Training

Models. For our experiments, we use 40 different chat models from prominent publishers such as Microsoft, Google, and Meta, with sizes ranging from 494M to 10.30B (*i.e.,* the largest model we could train on our cluster). Appendix C provides more details on these models.

Bradley-Terry w/ Binary Preferences. The most popular choice for reward modeling is the



(a) **Bradley-Terry w/ Binary Preferences**



(b) **Regression w/ Multi-Attribute Scores**

Figure 2: **Reward Modeling Performance Gains.** Relative gains are illustrated concerning the Llama-3.x model (marked as red) within the same group.

Bradley-Terry (BT) (Bradley and Terry, 1952; Ziegler et al., 2019) model. The underlying assumption of BT is that for a pair of responses $\mathcal{Y} = (y_1, y_2)$, the human preference distribution ρ^* is generated from a latent reward function $r^*(x, y)$, which we only have indirect access to. This assumption can be formalized as

$$\rho^*(y_1 \succ y_2|x) = \frac{\exp(r^*(x, y_1))}{\sum_y^{\mathcal{Y}} \exp(r^*(x, y))} . \quad (1)$$

Then, framing BT as a binary classification task, we can parameterize the reward function and optimize a negative log-likelihood loss as

$$\mathcal{L}_{BT} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma(\zeta(x, y_w) - \zeta(x, y_l))\right] \quad (2)$$

where $\mathcal{D} = \{(x^i, y_w^i, y_k^i)\}_{i=1}^N \sim \rho^*$ is a binary preferences dataset and ζ is an LLM with a linear head that outputs a single scalar as the reward value.

To create a compatible dataset, first, an LLM ξ generates pairs of responses for samples from a given prompt dataset $\mathcal{D}_x$:

$$\mathcal{D}_\xi = \{(x, y_1, y_2)|\{y_1, y_2\} \sim \xi(x)\}_{x\sim\mathcal{D}_x} . \quad (3)$$

Then, the pairs are labeled by humans (or synthetically) to obtain the binary preferences:

$$\mathcal{D} = \{(x, y_w, y_l)|(y_w \succ y_l; x)\}_{(x,y_1,y_2)\sim\mathcal{D}_\xi} . \quad (4)$$

We follow a similar setup for training the reward models as Wang et al. (2024c). Specifically, each model is trained for one epoch on the HelpSteer2-Preference dataset, using a global batch size of 64, a constant learning rate, searched over $\{5, 6, 7, 8, 9\}e-7 \cup \{1, 2, 3, 4, 5\}e-6$ for each model separately, and an AdamW optimizer (Loshchilov and Hutter, 2019) with 20 warm-up steps. Each model is saved every 20 steps, and the final model is chosen based on the accuracy of the saved models on the validation set.

Regression w/ Multi-Attribute Scores. While less explored compared to BT, Regression reward models (Wang et al., 2024e,a,d) have been posting impressive performance recently, topping the RewardBench at multiple points (*e.g.*, `ArmoRM-Llama3-8B-v0.1`[2] and

[2] RLHFlow/ArmoRM-Llama3-8B-v0.1

`Nemotron-4-340B-Reward`[3]). In contrast to the binary preferences, each sample is annotated with multiple values along different attributes (*e.g.,* Coherence, Correctness, Verbosity, etc.). Then, given an input x, an output score vector $y \in \mathbb{R}^n$, and an LLM ϕ, we optimize

$$\mathcal{L}_R = \text{MSE}(\phi(x)^{(-1)} W_\phi, y) \quad (5)$$

where $\phi(x)^{(-1)} \in \mathbb{R}^{\text{dim}(\phi)}$ is the last hidden state and $W_\phi \in \mathbb{R}^{\text{dim}(\phi) \times n}$ is a trainable linear projection (*i.e.,* a linear layer). This formulation leads to more flexible and interpretable reward models. To train the models, we follow a similar setup as Wang et al. (2024d). Specifically, each model is trained for two epochs on the HelpSteer2 dataset, using a global batch size of 64, a constant learning rate, searched over $\{1, 3, 5, 7, 9\}e - \{6, 7\}$ for each model separately, and an AdamW optimizer with 20 warm-up steps. Since RewardBench only supports BT models, for each model, we search for an optimal merge vector, w_m, as

$$\psi(x) = (\phi(x)^{(-1)} W_\phi))^T w \quad (6)$$

$$w_m = \underset{w \in S}{\text{argmax}} \sum_{x_c, x_r}^{D} \mathbb{1}\left(\psi(x_c) > \psi(x_r)\right) \quad (7)$$

where D is the validation set of HelpSteer2-Preference (Wang et al., 2024c), x_c and x_r are chosen and rejected responses, respectively, and $S = \{0.05k\}^4_{k=0,\ldots,20} \times \{-0.05k\}_{k=0,\ldots,20}$ ($\sim$4M combinations). We follow the approach in `Nemotron-4-340B-Reward` to assign positive weights for *Helpfulness*, *Correctness*, *Coherence*, *Complexity*, and a negative weight for *Verbosity*. Finally, we pick the model with the highest validation performance.

3.2 Evaluation

Following prior work (Wang et al., 2024d,c; Dorka, 2024; Lou et al., 2024; Zhang et al., 2024b; Wang et al., 2024a) and due to its popularity (*e.g.,* more than 150 entries), we evaluate our trained models using RewardBench (Lambert et al., 2024b), which contains $\sim$3k assorted tasks from 23 different datasets. Each task consists of a binary preference sample and is categorized into one of the following four categories: *Chat*, *Chat-Hard*, *Safety*, and *Reasoning*. We report the accuracy for each category and an overall score by averaging the accuracies.

[3] nvidia/Nemotron-4-340B-Reward

3.3 Experimental Results

To make a fairer comparison, we partition the models into three groups, each representing a range of roughly 3B parameters: $\{< 3\text{B}, (\geq 3\text{B}, < 6\text{B}), \geq 6\text{B}\}$. Then, we calculate the relative gains concerning the Llama-3.x model for each group (*i.e.,* the default choice) within the same group. Figure 2 present our results models trained using Bradley-Terry (w/ binary preferences) and Regression (w/ multi-attribute scores). While Llama-3.x models perform exceptionally well across our experiments, within each group, a few models post superior performances, with margins up to $\sim$14%. Specifically, looking at these top performances, models from the Qwen2.5 and Gemma-2 families consistently improve upon the results of their Llama-3.x counterpart, presenting reliable alternatives. Moreover, these experiments showcase the potentially high variances in performance within groups of models with similar sizes, which, in many cases, is the main limiting factor for model selection.

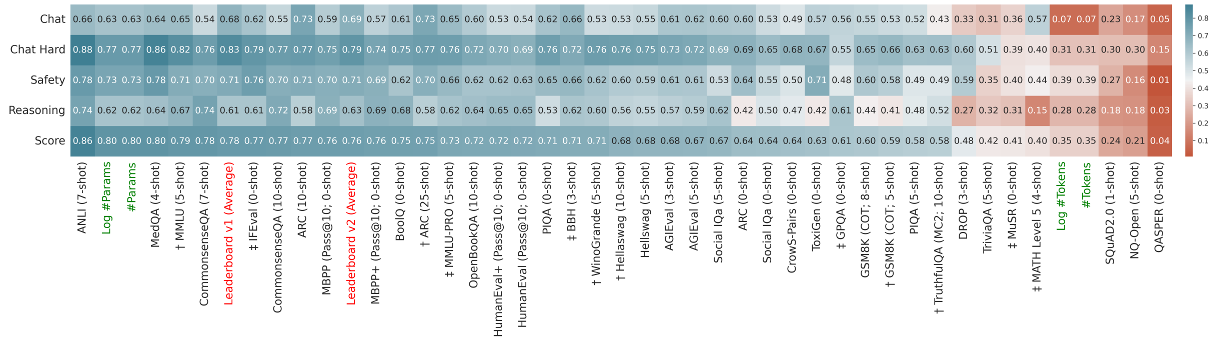
4 Benchmarks as Latent Skills Proxies

4.1 Statistical Correlation

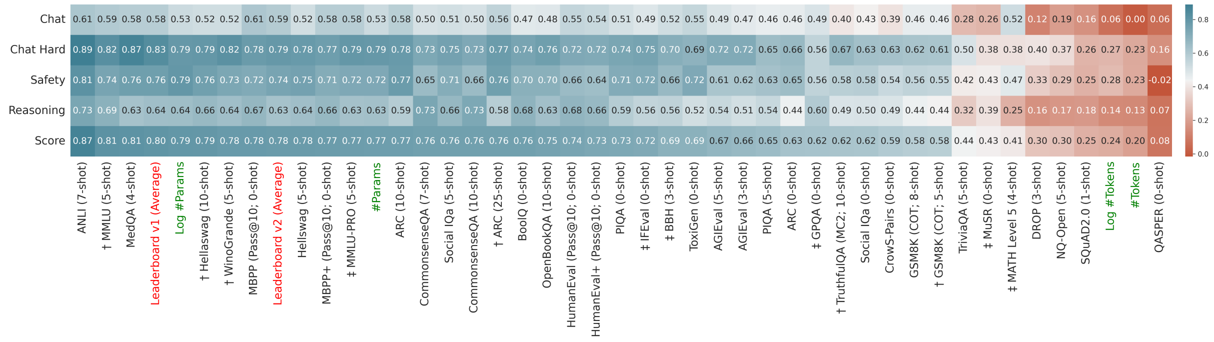
Setup. Practitioners often test their models on various benchmarks, covering many topics such as reasoning, coding, etc. These benchmarks, along with aggregate benchmarks such as Open LLM Leaderboard (Beeching et al., 2023; Fourrier et al., 2024) and HELM (Cecchini et al., 2024), act as a proxy measurement of the true capabilities of LLMs. Consequently, many of them are often used for model selection. For our analysis, we curate a list of 33 common benchmarks as reported in Llama-3.x (Dubey et al., 2024), Gemma-2 (Team et al., 2024), Phi-3.x (Abdin et al., 2024a), and Qwen2.5 (Yang et al., 2024) families (see Appendix B for more details). Besides these benchmarks, we also include training metrics such as the number of parameters and the number of training tokens, as they are commonly used in formulating scaling laws (Ruan et al., 2024; Polo et al., 2024).

Results. Figure 3 presents our correlation analysis between these benchmarks/metrics and the final reward modeling performances[4]. As evident, some benchmarks showcase a very high (≥ 0.8) correlation, both on Pearson and Spearman, with

[4] On the *Chat* subcategory, all the models achieve s 90-95% performance, which makes them challenging to distinguish considering minor performance variances; hence, we observe relatively low correlations across benchmarks.



(a) **Spearman Correlation**



(b) **Pearson Correlation**

Figure 3: **Statistical Correlation w.r.t. Reward Modeling Performance.** The subset benchmarks of Open LLM Leaderboard v2 (v1) are denoted with an ‡ (†). *Text Colors:* Red → Aggregate benchmark, Green → Training metric.

ANLI (Williams et al., 2022) consistently beating other benchmarks across different subcategories.

Significance Test. We test the significance of the correlation coefficient with the following statistic:

$$t_c = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \quad (8)$$

where r is the sample correlation coefficient, and n is the sample size, which leads to a threshold t_c of 0.316 ($n = 40$) for p-value < 0.05. Using this threshold, we observe that most of the benchmarks' correlations have statistical significance.

Coverage Test. While a high correlation shows a strong statistical relationship between the two variables, we also care about the coverage at different points in their rankings. Given a benchmark β and reward bench ρ, we formally define the coverage at top-k as

$$\mathcal{C}(\beta, \rho, \mathcal{L}, k) = \frac{|\mathcal{T}_\beta(\mathcal{L}, k) \cap \mathcal{T}_\rho(\mathcal{L}, k)|}{k} \quad (9)$$

where $\mathcal{L}$ is a set of LLMs and $\mathcal{T}_x(y, z)$ is the top z LLMs in y on benchmark x. To simulate a real-world search where we need high coverage at higher ranks, we filter out any benchmark with less than 0.4 and 0.7 coverage at $k = 5$ and $k = 10$, respectively. Figure 4 illustrates the coverage values from $k = 5$ to $k = 30$ on the remaining benchmarks (see Appendix B for more details). Notably, all the benchmarks mostly follow a log-linear coverage pattern concerning k, with ANLI outperforming the other benchmarks. However, we

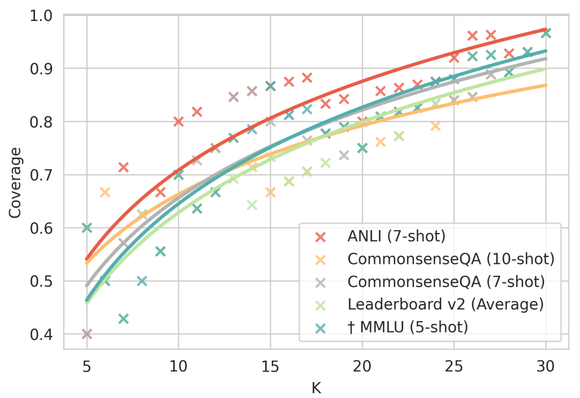


Figure 4: **Benchmark's Coverage.** We only retain benchmarks with at least 0.4 and 0.7 coverage at $k = 5$ and $k = 10$, respectively.

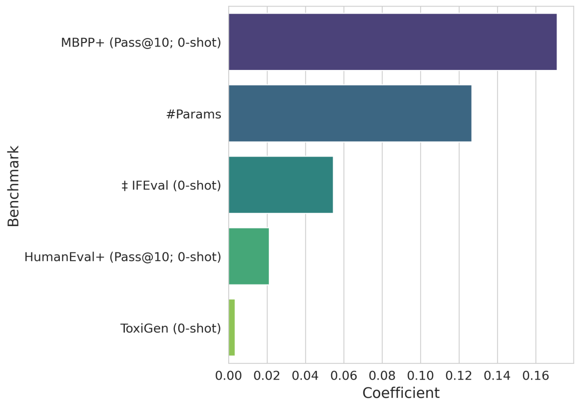


Figure 5: **Coefficients.** Only five benchmarks are assigned a non-zero weight by the trained model. The topics of these benchmarks are as follows: *Coding* $\rightarrow$ MBPP+ and HumanEval+, *Safety* $\rightarrow$ ToxiGen, *General* $\rightarrow$ IFEval, and *Training Metrics* $\rightarrow$ #Params (see Appendix B for more details).

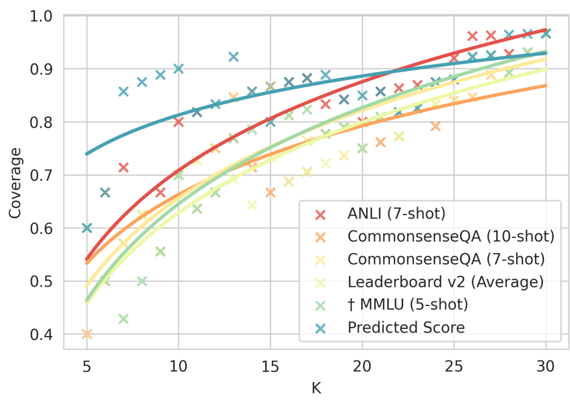


Figure 6: **Benchmarks vs. Predicted Score Coverage.** We only retain benchmarks with at least 0.4 and 0.7 coverage at $k = 5$ and $k = 10$, respectively.

also observe a relatively low coverage at higher ranks, which mitigates the effectiveness of using these benchmarks for model selection.

4.2 Regression Analysis

Setup. Considering the aforementioned low coverage in single-benchmark model selection, we hypothesize that combining the performances from a small set of benchmarks will yield much better predictive performance. To test this hypothesis, we run a 10-fold cross-validation experiment on an Elastic Net model, searching over the following hyperparameters: degree $\in \{1, 2, 3\}$, $\alpha \in \{0.1, 0.01, 0.001, 0.0001\}$, and l1_ratio $\in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. Then, we fit a model over all samples using the best hyperparameters.

Results. Figure 5 illustrates the benchmarks with a non-zero weight in the final model. Mapping back these five benchmarks to their main topics (see Appendix B for more details), we observe that they consist of two coding (MBPP+ (Liu et al., 2023) and HumanEval+ (Liu et al., 2023)), one safety (ToxiGen (Hartvigsen et al., 2022)), and one general (IFEval (Zhou et al., 2023)) benchmarks, along with one training metric (#Params). This combination closely follows the subcategories in RewarcBench: Coding $\approx$ Reasoning, Safety $=$ Safety, General + Training Metric $\approx$ Chat/Chat Hard. Moreover, in Figure 6, we compare the coverages of the fitted model to the standalone benchmarks. As evident, the trained model significantly improves the coverage in lower Ks, mitigating the critical problem of using standalone benchmarks. These results prove our hypothesis, showcasing the predictability of reward modeling performance from a low-dimensional vector of prior results.

5 Training Stages

5.1 Post-training

Setup. Traditionally, for training RMs, practitioners have used a base model that has undergone an SFT process (Stiennon et al., 2020). However, the recent advancements in LLMs have introduced more stages to the training process. In this section, we analyze the effect of these different stages on the RMs' performance using the publicly available models. While publishers don't regularly release the intermediate training checkpoints, recent efforts in open LLMs have made some of these intermediate models available for analysis. Specifically, for the `Llama-3.1-Tulu-3-8B`[5] model, Lambert et al. (2024a) have released three models from the end of each SFT, Direct Preference Optimization (DPO) (Rafailov et al., 2023), and Reinforcement Learning with Verifiable Rewards (RLVR) stages. Apart from the Tulu 3 model, we also include two other `Llama-3.1-8B`-based[6] models that have undergone the post-training phase, namely: `Llama-3.1-8B-Instruct`[7] and `Hermes-3-Llama-3.1-8B`[8] (Teknium et al., 2024).

Results. Table 1 presents our experimental results comparing different post-training stages to the base model. From these results, we can observe

[5] allenai/Llama-3.1-Tulu-3-8B
[6] meta-llama/Llama-3.1-8B
[7] meta-llama/Llama-3.1-8B-Instruct
[8] NousResearch/Hermes-3-Llama-3.1-8B; SFT + DPO.

Model	Chat	Δ	Chat Hard	Δ	Safety	Δ	Reasoning	Δ	Score	Δ
`Llama-3.1-8B`	93.9	-	53.7	-	64.7	-	79.1	-	72.9	-
`Llama-3.1-8B-Instruct`	95.3	1.5%	68.2	27.0%	84.6	30.8%	84.7	7.1%	83.2	14.1%
`Hermes-3-Llama-3.1-8B`	95.5	1.7%	71.3	32.8%	83.8	29.5%	74.0	-6.4%	81.1	11.2%
`Llama-3.1-Tulu-3-8B-SFT`	95.3	1.5%	70.8	31.8%	84.9	31.2%	85.8	8.5%	84.2	15.5%
`Llama-3.1-Tulu-3-8B-DPO`	94.7	0.9%	69.1	28.7%	82.3	27.2%	80.1	1.3%	81.6	11.9%
`Llama-3.1-Tulu-3-8B`	93.3	-0.6%	65.6	22.2%	83.5	29.1%	78.5	-0.8%	80.2	10.0%

Table 1: **Post-training Performances.** The Δ columns showcase the relative change to the base model's performance for each category.

that the post-training procedure significantly improves the overall performance of RMs. However, the extra steps after the SFT phase decrease the models' performance across all categories. This phenomenon could be due to the focus of these stages on human alignment, which slightly degrades other capabilities (Korbak et al., 2022). Looking at the subcategories, we note that the *Chat Hard* and *Safety* consistently get significant performance boosts (between 22-32%) after the post-training procedure. We believe this is due to dense exposure to related samples that focus on improving the models' safety and complex conversational capabilities. Moreover, the performances on *Chat* category remain primarily unchanged ($<2\%$), persistent with our previous observations in Section 4 where even the worst models posted high performances. Finally, in the *Reasoning* category, while the initial SFT stage moderately ($\sim$8.5%) improves the performance, the following stages reverse most of the gains. Given the focus of the RLVR stage on improving math capabilities, these results are somewhat surprising. This phenomenon might be explained by the fact that only 31% of reasoning samples in RewardBench are math-related, compared to 69% targeting coding correctness. However, given a potential co-dependence of math and coding capabilities, further investigation is needed on this phenomenon, which we leave to future works.

5.2 Pre-training

Setup. Prior works have examined the relation between eventual model capabilities and many LLMs' attributes, ranging from compute (Hoffmann et al., 2022) to downstream (Ruan et al., 2024) metrics. However, pre-training data distribution has remained a significant underexplored factor among these attributes, mainly due to its confidential, proprietary nature. Efforts in open LLM training (Liu et al., 2024d; OLMo et al., 2024) present an opportunity to study this factor. Recent studies (Shi et al., 2024; Zhang et al., 2024a; Zhang and Wu, 2024; Kim et al., 2024) have developed pre-training data detection techniques by viewing it as a membership inference attack (MIA) task. However, the curated MIA datasets lack the scale and coverage needed for a comprehensive analysis of the pre-training data distribution, as they have less than 10k samples. To address this issue, we curate a large-scale dataset by sampling 200k examples from each of the *Github*, *Book*, *ArXiv*, *Wikipedia*, and *StackExchange* subsets in SlimPajama (Soboleva et al., 2023), resulting in a 1M sample dataset[9]. Moreover, to detect the presence of a document in an LLM, we use a truncated version (*i.e.,* the first 2048 tokens) of the length-normalized sequence probability (Malinin and Gales, 2021). The truncation helps reduce the cost of running such analysis at scale, as some books have more than 170k tokens, and mitigates the noise from later tokens, as LLMs have shown to have a problem making robust use of tokens in the middle of long documents (Liu et al., 2024b; Hsieh et al., 2024).

Given a document $D = [t_i]_{i=1,\ldots,m}$, an LLM ϕ, and a tokens limit N, we calculate a presence score $\mathcal{S}_\phi$ as

$$\mathcal{S}_\phi(D, N) = \frac{1}{N}\sum_{i=1}^{N} \log p_\phi(t_i|t_{1:i-1}) \,. \qquad (10)$$

We use `Crystal`[10] (Liu et al., 2024d) as our ground truth LLM, as all of the SlimPajama dataset has been used in its pre-training stage. Finally, for each model, we reuse the extracted distribution from the largest member of its family if and only if they've been trained on the same amount of data, assuming the same data was used for the pre-training stage (see Appendix B for more details).

[9] 1.25% of all the documents in the original categories.
[10] LLM360/Crystal

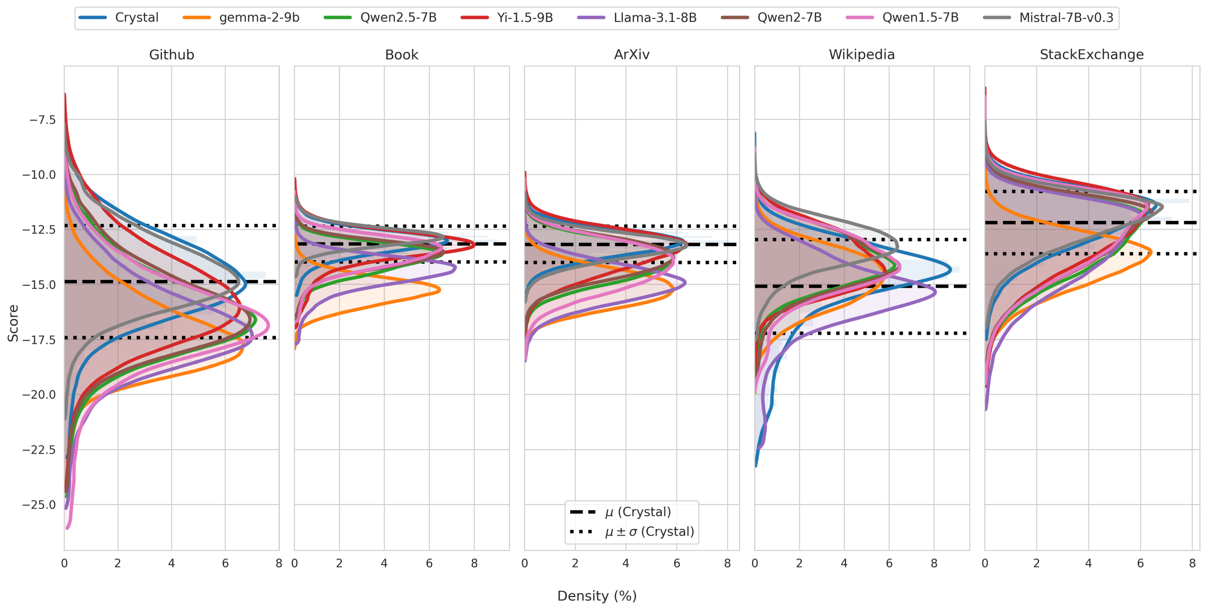


Figure 7: **Estimated Pre-training Data Distributions.** `Crystal` (Liu et al., 2024d) represents our ground truth, as it has seen the entire SlimPajama dataset in the pre-training phase exactly once.

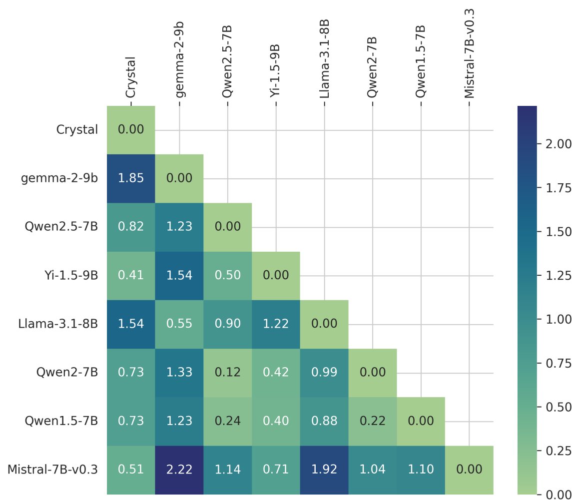


Figure 8: **Jensen-Shannon Distance.** The values are based on the scores from the entire dataset.

Results. Figure 7 illustrates the score distributions across different subsets of SlimPajama for seven models from different families. Notably, we observe a difference between the score ranges across the categories, even for the ground truth model that has seen everything once. We believe this is due to the potential occurrence of similar documents in the excluded *CommonCrawl* and *C4* categories. Figure 7 showcases the Jnsen-Shannon Distance (JSD) between different models over the scores of the entire 1M samples. As evident, some model pairs showcase significantly higher distances than others, showcasing a variability across models that can be utilized for downstream predictions. We also notice that the Qwen$\{1.5, 2, 2.5\}$ models have the lowest non-zero distances, which suggests that different generations of models released by a publisher potentially have significant overlaps in their pre-training data. Moreover, we expand our regression analysis (see Section 4.2) by adding the average scores of the categories to the already established five features (see Figure 5). Our experiments show that compared to adding these features improves the mean absolute error by $+1.5\%$ (from 3.2% to 1.7%), compared to only using the original five features, which showcases the untapped potential of the pre-training data distributions.

6 Conclusion

In this paper, we presented a systematic analysis of the effect of base model selection on the reward modeling performance. First, we showcased the significant variability of final performance by only changing the base model. Then, we analyzed the possibility of knowing a model's performance apriori, leading to a simple model with high coverage across the range of models, using commonly disclosed metrics and performances. Finally, we investigate different training stages, showcasing 1) the positive and negative effects of certain steps in post-training and 2) illustrating the untapped potential of using estimated pre-training data distributions.

Limitations

Training Regimen. While our experiments are designed to remove the effect of reward modeling training data (*i.e.,* using the same small dataset for all models), using larger datasets might reveal unknown behaviors for some models. However, given our computational resource constraints, we leave these experiments to future works, as the current cost of our experiments is ~4500 GPU/hours.

Post-training. In our analysis, we observed an interesting and unintuitive phenomenon where RLHF and preference optimization hurt the models' performance in the reasoning category of RewardBench. However, we only had access to a limited number of publicly available models; further investigation is needed to exhibit the main reason for this phenomenon.

Pre-training. Given our limited resources, we could only run our data distribution estimation experiments on a subset of models. Extending our model set in future works will boost our understanding of the effect of data distributions. Moreover, we relied on a relatively simple score to scale to the number of samples we had; further experiments with other methods at scale could help gain more insights.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024a. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. 2024b. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand. Association for Computational Linguistics.

Anthropic. 2024. Meet claude. https://www.anthropic.com/claude. Accessed: 2024-11-25.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

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*.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. MARS: Meaning-aware response scoring for uncertainty estimation in generative LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7752–7767, Bangkok, Thailand. Association for Computational Linguistics.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard (2023-2024). https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

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.

David Cecchini, Arshaan Nazir, Kalyan Chakravarthy, and Veysel Kocaman. 2024. Holistic evaluation of large language models: Assessing robustness, accuracy, and toxicity for real-world applications. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 109–117, Mexico City, Mexico. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N.

Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

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*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Ultrafeedback: boosting language models with scaled ai feedback. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Nicolai Dorka. 2024. Quantile regression for distributional reward models in rlhf. *arXiv preprint arXiv:2409.10164*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

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*.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`.

Leo Gao, John Schulman, and Jacob Hilton. 2023a. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Pengzhi Gao, Liwen Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2023b. Learning multilingual sentence representations with cross-lingual consistency regularization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 243–262, Singapore. Association for Computational Linguistics.

Gemini Team. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Srishti Gureja, Lester James V Miranda, Shayekh Bin Islam, Rishabh Maheshwary, Drishti Sharma, Gusti Winata, Nathan Lambert, Sebastian Ruder, Sara Hooker, and Marzieh Fadaee. 2024. M-rewardbench: Evaluating reward models in multilingual settings. *arXiv preprint arXiv:2410.15522*.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric

Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. 2022. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030. Curran Associates, Inc.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Gyuwan Kim, Yang Li, Evangelia Spiliopoulou, Jie Ma, Miguel Ballesteros, and William Yang Wang. 2024. Detecting training data of large language models via expectation maximization. *arXiv preprint arXiv:2410.07582*.

Tomasz Korbak, Ethan Perez, and Christopher Buckley. 2022. RL with KL penalties is better viewed as Bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1083–1091, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. 2024a. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. 2024b. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024a. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and LINGMING ZHANG. 2023. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2024c. Rm-bench: Benchmarking reward models of language models with subtlety and style. *arXiv preprint arXiv:2410.16184*.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Yonghao Zhuang, Guowei He, Haonan Li, Fajri Koto, Liping Tang, Nikhil Ranjan, Zhiqiang Shen, Roberto Iriondo, Cun Mu, Zhiting Hu, Mark Schulze, Preslav Nakov, Timothy Baldwin, and Eric P. Xing. 2024d. LLM360: Towards fully transparent open-source LLMs. In *First Conference on Language Modeling*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Xingzhou Lou, Dong Yan, Wei Shen, Yuzi Yan, Jian Xie, and Junge Zhang. 2024. Uncertainty-aware reward model: Teaching reward models to know what is unknown. *arXiv preprint arXiv:2410.00847*.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2024. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*.

OpenAI. 2024. Introducing openai o1. `https://openai.com/o1`. Accessed: 2024-11-25.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Felipe Maia Polo, Seamus Somerstep, Leshem Choshen, Yuekai Sun, and Mikhail Yurochkin. 2024. Sloth: scaling laws for llm skills to predict multi-benchmark performance across families. *arXiv preprint arXiv:2412.06540*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. 2024. Observational scaling laws and the predictability of langauge model performance. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

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

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. Slimpajama: A 627b token cleaned and deduplicated version of redpajama.

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. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Ryan Teknium, Jeffrey Quesnelle, and Chen Guang. 2024. Hermes 3 technical report. *arXiv preprint arXiv:2408.11857*.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024a. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10582–10592, Miami, Florida, USA. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024b. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2024c. Helpsteer2-preference: Complementing ratings with preferences. *arXiv preprint arXiv:2410.01257*.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024d. Helpsteer2: Open-source dataset for training top-performing reward models. *arXiv preprint arXiv:2406.08673*.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. 2024e. HelpSteer: Multi-attribute helpfulness dataset for SteerLM. 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 3371–3384, Mexico City, Mexico. Association for Computational Linguistics.

Adina Williams, Tristan Thrush, and Douwe Kiela. 2022. ANLIzing the adversarial natural language inference dataset. In *Proceedings of the Society for Computation in Linguistics 2022*, pages 23–54, online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*.

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*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Anqi Zhang and Chaofeng Wu. 2024. Adaptive pretraining data detection for large language models via surprising tokens. *arXiv preprint arXiv:2407.21248*.

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024a. Pretraining data detection for large language models: A divergence-based calibration method. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5263–5274, Miami, Florida, USA. Association for Computational Linguistics.

Yifan Zhang, Ge Zhang, Yue Wu, Kangping Xu, and Quanquan Gu. 2024b. General preference modeling with preference representations for aligning language models. *arXiv preprint arXiv:2410.02197*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. AGIEval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2299–2314, Mexico City, Mexico. Association for Computational Linguistics.

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*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

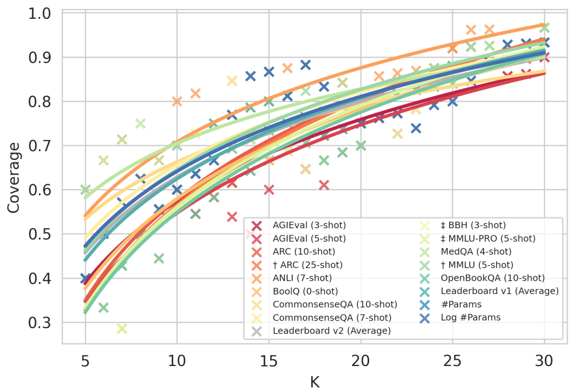


Figure 9: **Benchmark's Coverage.** We only retain benchmarks with at least 0.4 and 0.6 coverage at $k = 5$ and $k = 10$, respectively.

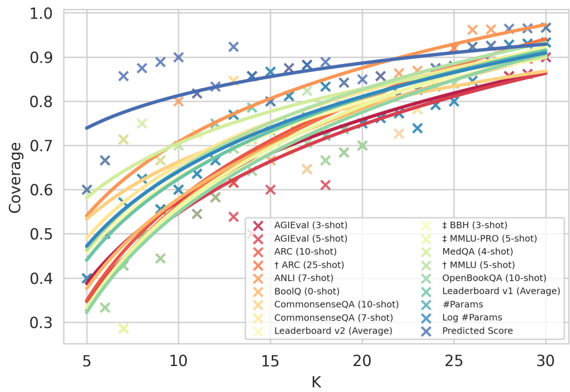


Figure 10: **Benchmarks vs. Predicted Score Coverage.** We only retain benchmarks with at least 0.4 and 0.6 coverage at $k = 5$ and $k = 10$, respectively.

A RewardBench as Ground Truth

Given the heavy reliance of our work on RewardBench, we conduct an independent verification of the preferences. Specifically, we sample 50 tasks from the tasks that our top 10 models got wrong the most. Then, we gather 3 annotations from different annotators and use a majority vote to determine the final preference. All annotators were senior Computer Science PhD students specializing in NLP with extensive experience working with and evaluating LLMs. Our results show an agreement of 98%, establishing the quality of RewardBench.

B Benchmarks

Table 2 showcases all the 32 benchmarks used in our experiments. Moreover, Figure 9 illustrates the coverage using an expanded set of benchmarks with at least 0.4 and 0.6 coverage at k = 5 and k = 10, respectively.

C Models

Table 3 showcases all the 40 models used in our experiments.

D Full Results

Table 5 and Table 4 present the full results using the Bradley-Terry and Regression methods, respectively.

E Bradley-Terry vs. Regression

Setup. The training method is one of the early design choices for reward modeling, significantly influencing the costly data curation process, as the data format is often not easily transferable. While

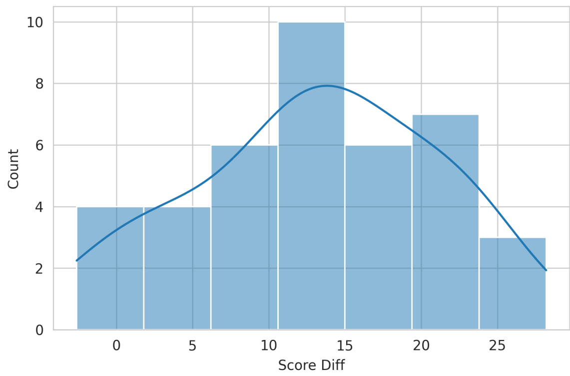


Figure 11: **Bradley-Terry vs. Regression Performance Difference.** A positive value indicates a better performance on the Regression method.

previous works have briefly compared Bradley-Terry vs. Regression training (Wang et al., 2024c), finding their similar performances on ~70B models, our understanding of their differences is somewhat limited. In our experiments, we use the HelpSteer2 and HelpSteer2-Preference datasets, which have the same underlying samples with different annotation styles[11]. This setup presents an opportunity to compare these two approaches fairly.

Results. Figure 11 illustrates the performance difference between Bradley-Terry and Regression methods across our model pool. As evident, the Regression models outperform their Bradley-Terry counterparts by a large margin. We also observe that the gap is much less with stronger models (*e.g.,* `Qwen2.5-7B-Instruct` and `gemma-2-9b-it`), which could lead to a perfor-

[11] HelpSteer2-Preference excludes indistinguishable responses (denoted by human annotators), which Bradley-Terry w/ Binary Preferences can not model.

Framework	Dataset	Topic	#Shots	Models
lm_eval (Gao et al., 2024)	leaderboard_ifeval (Zhou et al., 2023)	General	0	LGPQ
	winogrande (Sakaguchi et al., 2021)		5	LGP
	hellaswag (Zellers et al., 2019)	Reading Comprehension	5,10	GP
	openbookqa (Mihaylov et al., 2018)		10	P
	triviaqa (Joshi et al., 2017)		5	LGP
	squadv2 (Rajpurkar et al., 2018)		1	L
	drop (Dua et al., 2019)		3	L
	boolq (Clark et al., 2019)		0	LGP
	anli (Zhong et al., 2024)	Adversarial	7	P
	truthfulqa_mc2 (Lin et al., 2022)		10	GP
	commonsense_qa (Talmor et al., 2019)	Commonsense Reasoning	7,10	LP
	piqa (Bisk et al., 2020)		0,5	GP
	social_iqa (Sap et al., 2019)		0,5	GP
	nq_open (Kwiatkowski et al., 2019)		5	G
	agieval_en (Zhong et al., 2024)	Expert Reasoning	3,5	LGP
	ai2_arc (Clark et al., 2018)		0,10,25	LGP
	leaderboard_bbh (Suzgun et al., 2023)		3	LGPQ
	leaderboard_gpqa (Rein et al., 2024)		0	LGPQ
	leaderboard_mmlu_pro (Wang et al., 2024b)		5	LGPQ
	leaderboard_musr (Gao et al., 2023b)		0	LGPQ
	medqa_4options (Jin et al., 2021)		2	P
	mmlu (Hendrycks et al., 2021)		5	LGP
	gsm8k_cot_llama (Cobbe et al., 2021)	Math	5,8	LGPQ
	leaderboard_math (Hendrycks et al., 2021)		4	LGPQ
	crows_pairs_english (Nangia et al., 2020)	Safety	0	G
	toxigen (Hartvigsen et al., 2022)		0	G
	qasper (Dasigi et al., 2021)	Long-context	0	P
	leaderboard v1 (Beeching et al., 2023)	Aggregate	-	LGPQ
	leaderboard v2 (Fourrier et al., 2024)		-	LGPQ
evalplus (Liu et al., 2023)	HumanEval (Chen et al., 2021)	Coding	0	LGPQ
	HumanEval+ (Liu et al., 2023)		0	LGPQ
	MBPP (Austin et al., 2021)		0	LGPQ
	MBPP+ (Liu et al., 2023)		0	LGPQ

Table 2: **Benchmarks.** We gather a comprehensive list of 33 common benchmarks from the technical reports of well-known models. **Legened:** L → Llama-3.x, G → Gemma-2, P → Phi-3.5, and Q → Qwen2.5.

Publisher	Model	Release Date (First Commit)	#Params (B)	#Downloads (Feb 2025)	#Likes	#Pre-training Tokens (T)
Microsoft	Phi-3.5-mini-instruct	08/2024	3.82	1.143M	776	3.4
	Phi-3-small-8k-instruct	05/2024	7.38	25.1k	160	4.8
	Phi-3-mini-4k-instruct	04/2024	3.82	900k	1122	3.3
Google	gemma-2-9b-it	06/2024	9.24	393.4k	639	8.0
	gemma-2-2b-it	07/2024	2.61	437.6k	915	2.0
	gemma-1.1-7b-it	03/2024	8.54	20.7k	270	6.0
	gemma-1.1-2b-it	03/2024	2.51	93.3k	154	6.0
	gemma-7b-it	02/2024	8.54	62.1k	1151	6.0
	gemma-2b-it	02/2024	2.51	105.8k	701	6.0
Meta	Llama-3.2-3B-Instruct	09/2024	3.21	1.497M	939	9.0
	Llama-3.2-1B-Instruct	09/2024	1.24	1.523M	738	9.0
	Llama-3.1-8B-Instruct	07/2024	8.03	5.669M	3546	15.0
	Meta-Llama-3-8B-Instruct	04/2024	8.03	2.101M	3788	15.0
01.ai	Yi-1.5-9B-Chat	05/2024	8.83	20.9k	139	3.6
	Yi-1.5-6B-Chat	05/2024	6.06	19.6k	41	3.6
	Yi-6B-Chat	11/2023	6.06	9.3k	65	3.0
Alibaba	Qwen2.5-7B-Instruct	09/2024	7.62	1.275M	459	18.0
	Qwen2.5-3B-Instruct	09/2024	3.09	326.5k	158	18.0
	Qwen2.5-1.5B-Instruct	09/2024	1.54	592.5k	299	18.0
	Qwen2.5-0.5B-Instruct	09/2024	0.49	696.2k	198	18.0
	Qwen2-7B-Instruct	06/2024	7.62	821.4k	611	7.0
	Qwen2-1.5B-Instruct	06/2024	1.54	187.9k	134	7.0
	Qwen2-0.5B-Instruct	06/2024	0.49	170.3k	174	12.0
	Qwen1.5-7B-Chat	01/2024	7.72	25.5k	165	4.0
	Qwen1.5-4B-Chat	01/2024	3.95	5.6k	38	2.4
	Qwen1.5-1.8B-Chat	01/2024	1.84	11.2k	48	2.4
	Qwen1.5-0.5B-Chat	01/2024	0.62	556.2k	76	2.4
Mistral AI	Mistral-7B-Instruct-v0.3	05/2024	7.25	1.755M	1293	8.0
	Mistral-7B-Instruct-v0.2	12/2023	7.24	3.586M	2634	8.0
	Mistral-7B-Instruct-v0.1	09/2023	7.24	1.332M	1547	8.0
Stability AI	stablelm-2-1_6b-chat	04/2024	1.64	4.4k	32	2.0
Nvidia	Mistral-NeMo-Minitron-8B-Instruct	10/2024	8.41	3.1k	71	15.0
	Nemotron-Mini-4B-Instruct	09/2024	4.20	0.1k	147	8.0
Ai2	Llama-3.1-Tulu-3-8B-SFT	11/2024	8.03	23.4k	21	15.0
	Llama-3.1-Tulu-3-8B-DPO	11/2024	8.03	28.5k	22	15.0
	Llama-3.1-Tulu-3-8B	11/2024	8.03	12.7k	139	15.0
TII	Falcon3-10B-Instruct	12/2024	10.30	37,9k	87	16.0
	Falcon3-7B-Instruct	12/2024	7.46	45.2k	49	14.0
	Falcon3-3B-Instruct	12/2024	3.23	30.5k	23	14.1
	Falcon3-1B-Instruct	12/2024	1.67	31.4k	32	14.1

Table 3: **Models.** We curate an inclusive list of 40 models from prominent model providers.

Publisher	Model	Chat	Chat Hard	Safety	Reasoning	Score
Microsoft	Phi-3.5-mini-instruct	96.1	62.3	77.2	76.9	78.1
	Phi-3-small-8k-instruct	89.7	66.7	76.4	57.0	72.4
	Phi-3-mini-4k-instruct	96.4	58.6	77.2	83.6	78.9
Google	gemma-2-9b-it	95.8	74.1	88.4	94.3	88.1
	gemma-2-2b-it	94.7	56.8	79.9	80.7	78.0
	gemma-1.1-7b-it	97.2	61.0	81.1	79.5	79.7
	gemma-1.1-2b-it	89.4	46.3	74.6	50.5	65.2
	gemma-7b-it	93.3	60.5	83.4	78.1	78.8
	gemma-2b-it	92.2	42.5	67.0	56.7	64.6
Meta	Llama-3.2-3B-Instruct	95.3	68.6	87.7	59.3	77.7
	Llama-3.2-1B-Instruct	93.3	42.3	65.4	70.2	67.8
	Llama-3.1-8B-Instruct	95.3	68.2	84.6	84.7	83.2
	Meta-Llama-3-8B-Instruct	93.9	75.4	86.6	81.2	84.3
01.AI	Yi-1.5-9B-Chat	95.8	69.5	80.1	88.7	83.5
	Yi-1.5-6B-Chat	93.3	63.4	77.2	78.3	78.0
	Yi-6B-Chat	93.3	56.4	71.5	67.4	72.2
Alibaba	Qwen2.5-7B-Instruct	94.7	72.8	87.8	90.7	86.5
	Qwen2.5-3B-Instruct	92.7	63.4	82.0	85.3	80.8
	Qwen2.5-1.5B-Instruct	92.7	56.4	80.7	84.8	78.6
	Qwen2.5-0.5B-Instruct	89.9	45.6	51.9	48.4	59.0
	Qwen2-7B-Instruct	95.3	66.4	78.4	84.0	81.0
	Qwen2-1.5B-Instruct	92.7	47.8	72.0	79.0	72.9
	Qwen2-0.5B-Instruct	92.2	39.9	54.7	60.7	61.9
	Qwen1.5-7B-Chat	93.3	51.8	74.6	81.3	75.2
	Qwen1.5-4B-Chat	91.1	50.9	78.0	77.6	74.4
	Qwen1.5-1.8B-Chat	90.8	40.1	56.4	64.8	63.0
	Qwen1.5-0.5B-Chat	91.3	43.2	58.0	58.0	62.6
Mistral AI	Mistral-7B-Instruct-v0.3	94.1	62.3	75.1	84.1	78.9
	Mistral-7B-Instruct-v0.2	93.0	59.9	78.2	79.5	77.6
	Mistral-7B-Instruct-v0.1	92.7	58.8	71.1	71.8	73.6
Stability AI	stablelm-2-1_6b-chat	90.5	47.4	59.3	69.0	66.5
Nvidia	Mistral-NeMo-Minitron-8B-Instruct	93.6	61.0	82.6	82.9	80.0
	Nemotron-Mini-4B-Instruct	93.0	61.4	75.0	82.0	77.8
Ai2	Llama-3.1-Tulu-3-8B-SFT	95.3	70.8	84.9	85.8	84.2
	Llama-3.1-Tulu-3-8B-DPO	94.7	69.1	82.3	80.1	81.6
	Llama-3.1-Tulu-3-8B	93.3	65.6	83.5	78.5	80.2
TII	Falcon3-7B-Instruct	96.6	64.0	89.7	80.4	82.7
	Falcon3-3B-Instruct	95.0	53.9	78.1	73.9	75.2
	Falcon3-1B-Instruct	84.6	31.6	53.2	46.2	53.9
	Falcon3-10B-Instruct	95.5	67.3	89.5	91.1	85.9

Table 4: **Regression Performance.**

Publisher	Model	Chat	Chat Hard	Safety	Reasoning	Score
Microsoft	Phi-3.5-mini-instruct	61.5	51.5	63.1	61.1	59.3
	Phi-3-small-8k-instruct	83.5	55.3	81.9	75.8	74.1
	Phi-3-mini-4k-instruct	64.8	46.1	56.6	59.7	56.8
Google	gemma-2-9b-it	83.8	51.1	70.8	83.6	72.3
	gemma-2-2b-it	84.1	46.5	67.6	81.3	69.9
	gemma-1.1-7b-it	76.3	45.4	65.4	75.1	65.5
	gemma-1.1-2b-it	74.0	41.9	67.0	63.2	61.5
	gemma-7b-it	77.1	43.0	63.8	72.5	64.1
	gemma-2b-it	79.6	39.0	65.0	63.7	61.8
Meta	Llama-3.2-3B-Instruct	70.4	47.4	50.8	58.9	56.9
	Llama-3.2-1B-Instruct	57.0	51.3	58.0	56.0	55.6
	Llama-3.1-8B-Instruct	78.2	62.1	69.5	65.1	68.7
	Meta-Llama-3-8B-Instruct	73.2	53.9	57.2	59.1	60.9
01.AI	Yi-1.5-9B-Chat	80.7	54.8	62.8	67.4	66.4
	Yi-1.5-6B-Chat	76.5	50.2	59.9	81.3	67.0
	Yi-6B-Chat	71.5	52.9	67.0	71.6	65.7
Alibaba	Qwen2.5-7B-Instruct	90.5	61.8	78.1	74.1	76.1
	Qwen2.5-3B-Instruct	74.0	57.0	75.1	75.8	70.5
	Qwen2.5-1.5B-Instruct	80.2	49.6	58.4	78.1	66.6
	Qwen2.5-0.5B-Instruct	79.1	42.5	55.3	69.5	61.6
	Qwen2-7B-Instruct	85.5	51.1	57.8	76.7	67.8
	Qwen2-1.5B-Instruct	70.7	47.4	56.1	69.0	60.8
	Qwen2-0.5B-Instruct	70.4	48.0	57.0	67.8	60.8
	Qwen1.5-7B-Chat	77.7	51.3	62.3	69.3	65.1
	Qwen1.5-4B-Chat	75.4	48.9	53.0	66.6	61.0
	Qwen1.5-1.8B-Chat	79.9	40.4	59.9	62.9	60.8
	Qwen1.5-0.5B-Chat	71.5	44.1	60.3	54.7	57.7
Mistral AI	Mistral-7B-Instruct-v0.3	56.7	53.1	58.2	50.0	54.5
	Mistral-7B-Instruct-v0.2	80.7	38.2	54.1	58.1	57.8
	Mistral-7B-Instruct-v0.1	56.7	52.6	58.4	57.2	56.2
Stability AI	stablelm-2-1_6b-chat	71.2	49.3	60.5	59.9	60.2
Nvidia	Mistral-NeMo-Minitron-8B-Instruct	86.3	50.2	56.9	77.4	67.7
	Nemotron-Mini-4B-Instruct	81.6	49.8	63.2	50.9	61.4
Ai2	Llama-3.1-Tulu-3-8B-SFT	65.4	53.9	59.9	69.1	62.1
	Llama-3.1-Tulu-3-8B-DPO	76.5	41.9	58.5	57.5	58.6
	Llama-3.1-Tulu-3-8B	78.5	38.6	58.2	59.7	58.8
TII	Falcon3-7B-Instruct	50.6	57.0	50.5	74.2	58.1
	Falcon3-3B-Instruct	70.4	52.4	57.2	55.3	58.8
	Falcon3-1B-Instruct	65.4	44.3	50.4	59.3	54.8
	Falcon3-10B-Instruct	53.1	51.5	57.4	68.8	57.7

Table 5: **Bradley-Terry Performance.**

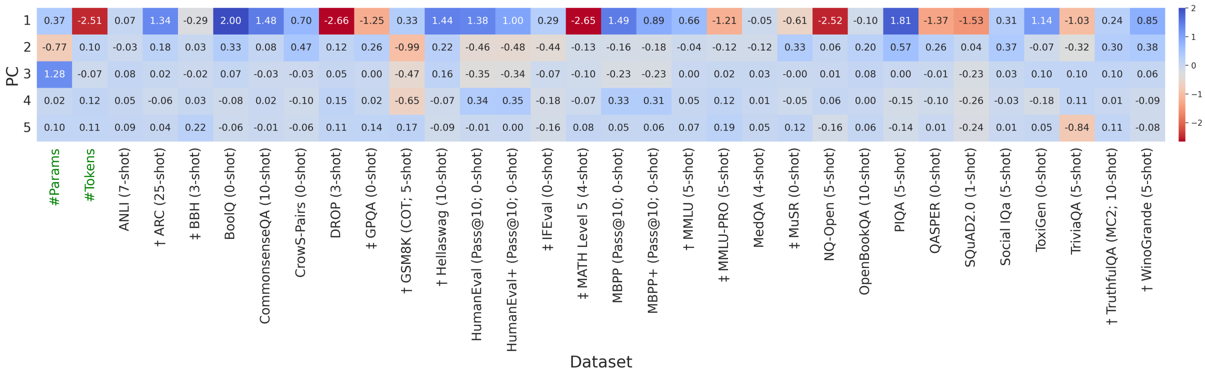


Figure 12: **Principal Component's Weights.**

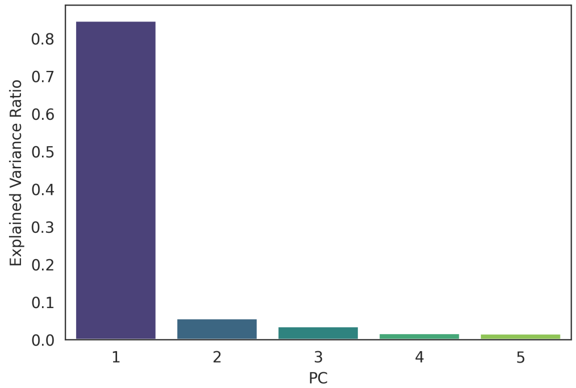


Figure 13: **PCA Explained Variance.** We find that the top 5 PCs explain ∼96.8% of the variance; hence, the benchmark-model matrix is low-dimensional.

mance match on 70B scale models, consistent with previous findings (see Appendix D for more details). This observation suggests that the Regression method is less reliant on the quality of the base model, making it a better overall choice when possible. Moreover, we note much more overfitting and instability when training with the Bradley-Terry method, making obtaining high-quality RMs more challenging.

F Low-dimensional Capabilities

Setup. Prior works (Ruan et al., 2024; Polo et al., 2024) have found the LLMs' capabilities to be low-dimensional, meaning that most of the variance over the standard benchmarks can be explained by a few principal components (PCs). Since our experiments use an expanded set of benchmarks (5 vs. 32), we replicate their analysis at a larger scale. Moreover, Ruan et al. (2024) find that the PCs are explainable, meaning specific topics, such as reasoning or coding, can explain each of them.

Results. Figure 13 illustrates the explained variance by the first five PCs (∼97%), which verifies that the benchmark-model matrix is low-dimensional. Moreover, Figure 12 replicates their analysis over the expanded set of benchmarks. While some PCs showcase a strong connection to specific topics (*e.g.,* PC4 ≈ Math + Coding), we can not assign clear-cut topics to them, in contrast to prior findings.

G Implementation Details

All our experiments are carried out on a server with 8 × RTX A6000 GPUs with 48GB VRAM, 500GB RAM, and 64 CPU cores. Moreover, we implemented our code using Hugging Face Transformers (Wolf et al., 2020) and PyTorch (Paszke et al., 2019) libraries.