

From Individual to Common: An Early Exploration of Consensus in Non-verifiable Data for Balanced Preference Optimization

Shangjian Yin and Zhouxing Shi

Department of Computer Science and Engineering
University of California, Riverside
{syin041, zhouxing.shi}@ucr.edu

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has demonstrated remarkable effectiveness in boosting the objective performance (e.g., reasoning) of Large Language Models (LLMs) through rule-based, on-policy self-improvement strategies. However, optimizing LLMs for subjective capabilities and alignment with human preferences remains challenging due to the non-verifiable nature. Most prior works use datasets comprising response pairs with substantial quality gaps labeled by a strong external judge. While effective for preference metrics, this paradigm often incurs an “alignment tax”, where the model’s objective performance on downstream tasks degrades as it overfits to subjective preferences. In this work, we introduce Donkey, a high-quality, non-verifiable dataset where response pairs differ only by subtle nuances. We find that LLMs optimized on Donkey via preference learning outperform those trained on data with explicit quality gaps, while simultaneously maintaining their objective capabilities. Furthermore, we observe that preference signals on Donkey can be decomposed into consensus preferences and individual preferences. Our analysis reveals that distilling consensus preferences provides a significantly more data-efficient signal for preference optimization. Our findings underscore the importance of leveraging nuanced preference signals and the consensus of multiple judges for advancing subjective LLM alignment. Our code and data will be available at <https://github.com/SJY8460/Donkey>.

1 Introduction

The landscape of Natural Language Processing (NLP) has been fundamentally transformed by the emergence of Large Language Models (LLMs), which exhibit exceptional proficiency across diverse domains ranging from complex mathematical reasoning and code synthesis to nuanced dialogue and general language comprehension (Cobbe et al., 2021; Wei et al., 2022; Bubeck et al., 2023; Chen et al., 2024b; Yin et al., 2025a,b). To develop helpful and safe LLM assistants, aligning model behavior with human preferences has become

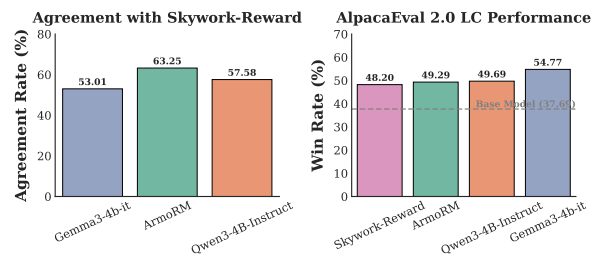
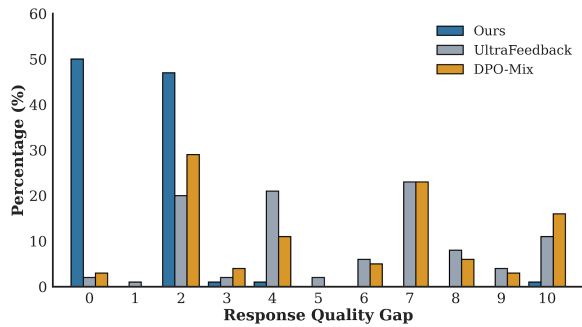


Figure 1: Agreement rates of different LLM-as-a-judge and reward models with Skywork-Reward, the top-ranked reward model on RewardBench, computed on our Donkey dataset, and their corresponding Length-Controlled Win Rate (LC) on AlpacaEval 2.

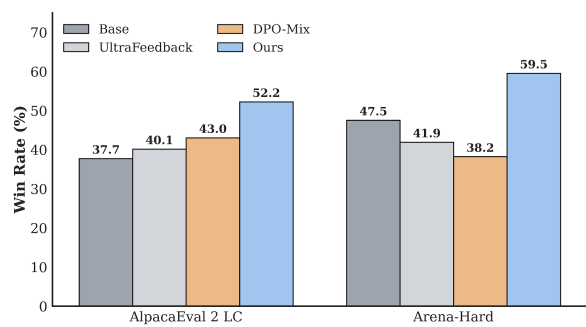
one of the central focuses of recent research (Mishra et al., 2021; Victor et al., 2022; Chung et al., 2022; Thop-pilan et al., 2022). LLM alignment typically involves an initial Supervised Fine-Tuning (SFT) on curated demonstration data (Ouyang et al., 2022; Tunstall et al., 2023; Yin et al., 2026), followed by preference optimization with Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Bai et al., 2022a) or AI Feedback (RLAIF) (Bai et al., 2022b; Lee et al., 2024).

Existing widely used preference optimization datasets, such as UltraFeedback (Cui et al., 2024) and DPO-Mix (Argilla, 2024), broadly include both verifiable data (e.g., math and code) and non-verifiable data (e.g., creative writing) that exhibit *distinct quality gaps*, as shown in Figure 2a. In this scenario, we find that different reward models tend to achieve around 80~90% agreement with each other. Consequently, usually a strong external model can be utilized to label preference signals (Wu et al., 2024; Meng et al., 2024). However, while previous works suggested that data with larger relative quality gaps as measured by explicit reward margins helps preference optimization and boosts subjective chat capabilities (Wang et al., 2024b; Huang et al., 2025), it often causes an “alignment tax” (Lin et al., 2024) and degrades performance on downstream tasks (Gao et al., 2023; Meng et al., 2024).

In contrast, we find that *absolute quality* is paramount. Unlike the large-margin approach which often degrades downstream capabilities, we observe that when both chosen and rejected responses possess high absolute quality, even when the relative quality gap is small, the



(a) Response Quality Gap Distribution



(b) Performance on Chat Benchmarks

Figure 2: Analysis of responses quality gap and model’s subjective chat performance. Specific dataset characteristics and quality gap evaluation template are provided in the Appendix A.1 and A.2.

data effectively improves alignment while maintaining or even enhancing downstream performance. Furthermore, regarding data composition, distinct from verifiable tasks that typically leverage rule-based verification, non-verifiable scenarios are inherently more complex and diverse due to their subjective nature. Crucially, we find that such *non-verifiable* instruction-following data yields significantly better preference optimization results compared to verifiable data that partially exist in datasets such as UltraFeedback. Since existing datasets contain only a small proportion of such high-quality, non-verifiable data (as shown in Figure 2a) and were generated using relatively dated LLMs, we introduce **Donkey**, a new *non-verifiable* synthetic dataset comprising 10k samples, where responses are designed to have *high absolute quality* with a *small relative quality gap*.

Empirical results demonstrate the efficacy of Donkey. Specifically, the model trained on our dataset labeled by the top reward model achieve 48.20% LC on AlpacaEval 2 (Dubois et al., 2025) and 64.6% WR on Arena-Hard (Li et al., 2024a). This represents a substantial improvement over the base model (37.7% and 47.5%), as well as models trained on UltraFeedback (40.1% and 41.9%) and DPO-Mix (43.0% and 38.2%). Crucially, Donkey maintains or even slightly improves objective performance across four benchmarks of downstream tasks, reaching 33.6% compared to the base model’s 32.7%. In stark contrast, models trained on UltraFeedback and DPO-Mix suffer significantly from the “alignment tax,” dropping to approximately 26%.

The intended small quality gap within the Donkey dataset reveals a surprising finding regarding the discernment of subtle quality differences. As illustrated in Figure 1, we observe a noticeable disagreement (> 30%) among different judges. Yet, distinct from this inconsistency, training on the individual signals from any single judge consistently yields promising improvements, with all achieving over 48% LC on AlpacaEval 2 compared to the base model’s 37.7%. Crucially, on our Donkey dataset, we define two distinct types of signals among multiple judges: **consensus signals** and **individual signals**. It is important to note that in this non-verifiable context, this distinction differentiates between strong,

shared agreements and weak, subjective nuances, rather than a binary distinction between “correct” and “wrong.” We find that consensus preference signals allow the model to achieve competitive or even superior performance compared to utilizing signals from a single strong judge, while requiring significantly less data. In contrast, highly individual judgments tend to have a limited impact on preference optimization, yielding results that are only marginally superior to random selection. Empirically, with the same data size, the model trained on consensus signals achieves an average of 55.86% across AlpacaEval 2 LC and Arena-Hard WR, outperforming single reward judges such as ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024a) (52.2% on average) and Skywork-Reward-V2-Llama-3.1-8B (Liu et al., 2025) (52.1% on average).

Our contributions are summarized as follows: (1) We introduce Donkey, a high-quality synthetic dataset designed to explore fine-grained preference signals in non-verifiable domains, where response pairs are both of similarly high quality and distinguishable primarily by subjective preference. (2) We demonstrate that models trained on Donkey achieve significantly superior performance on standard chat benchmarks compared to those trained on existing datasets with distinct quality gaps, while effectively avoiding the “alignment tax” on downstream tasks. (3) We provide a detailed analysis of the impact of absolute quality versus relative quality gaps, revealing that high absolute quality is paramount even when the relative quality gap is small. (4) We further discover a critical distinction between consensus signals and individual signals, finding that distilling consensus signals provides a data-efficient optimization path that matches or exceeds strong single-judge baselines.

2 Related Work

2.1 Synthetic Data for LLMs

High-quality data serves as the cornerstone for the effective instruction tuning of LLM, having proven remarkably effective in enhancing performance across specialized domains such as code generation (Roziere et al., 2023) and mathematical reasoning (Yuan et al.,

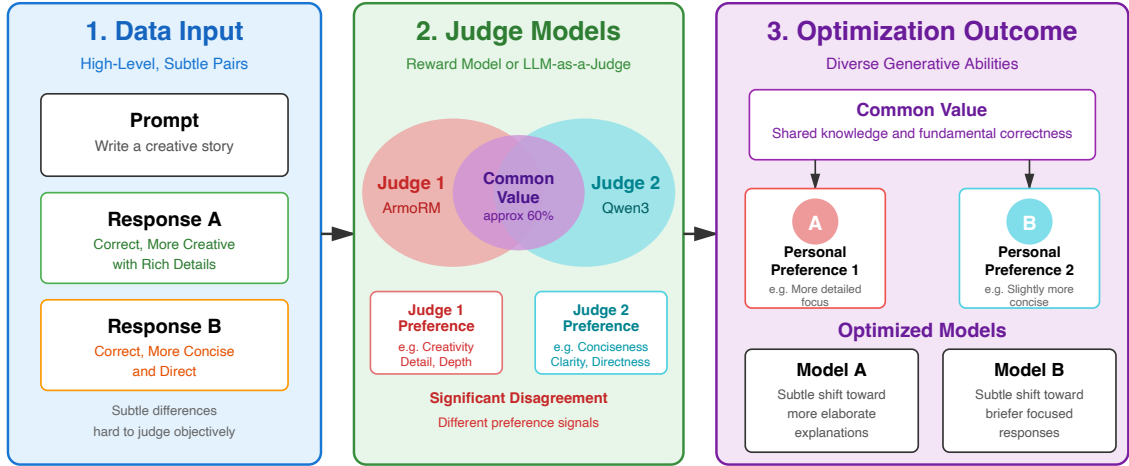


Figure 3: A brief illustration of preference signals in the Donkey dataset. The diagram demonstrates how subtle nuances in data pairs lead to disagreement among multiple judge models in non-verifiable scenarios.

2023; Luo et al., 2023; Zhu et al., 2025). However, while human-crafted data represents the gold standard for quality, acquiring it at scale is often prohibitively expensive. Consequently, the field has witnessed a paradigm shift towards the utilization of synthetic data as a cost-effective alternative, predominantly leveraging advanced proprietary models such as the GPT series (Radford et al., 2019; Brown et al., 2020; OpenAI, 2023) to synthesize high-quality training samples (Josifoski et al., 2023; Taori et al., 2023; Chiang et al., 2023; Li et al., 2023b). Recent methodologies have further refined this process by exploiting LLMs’ rephrasing abilities to diversify responses (Deng et al., 2023; Prasad et al., 2023) and augmenting datasets for improved fine-tuning efficiency (Yu et al., 2023; Liu et al., 2023). In the specific realm of preference optimization, widely adopted datasets such as UltraFeedback (Cui et al., 2024) typically distill data by sampling diverse model trajectories and constructing preference pairs based on scores assigned by a judge such as GPT-4. This method inherently produces pairs with noticeable quality gaps. In contrast, our research pivots to focus on datasets constructed purely on subjective preference signals, where response pairs are distinguishable only by subtle, non-verifiable nuances.

2.2 LLM-as-a-Judge and Reward Models

Leveraging LLM-as-a-Judge to evaluate language models has been a standard paradigm (Dubois et al., 2024; Li et al., 2023a; Fernandes et al., 2023; Bai et al., 2023; Saha et al., 2023; Chen et al., 2025; Wei et al., 2025a). Beyond evaluation, this technique is extensively employed for training reward models and curating data (Chen et al., 2024a; Li et al., 2024b; Lee et al., 2024; Wei et al., 2025b). Concurrently, specialized reward models (Wang et al., 2024a; Liu et al., 2025) represent another cost-effective and robust approach for annotating preference signals. Since these models are supervised on vast amounts of preference datasets de-

rived from human conversations and synthetic labels, they can effectively reflect human preferences, often outperforming vanilla LLMs on benchmarks such as RewardBench (Lambert et al., 2024; Malik et al., 2025). Notably, in scenarios where distinct quality gaps exist between responses, both advanced LLM-as-a-Judge systems and reward models tend to yield highly consistent judgments. Consequently, the practice in recent works is to adopt a single external judge for labeling preference data (Yuan et al., 2024; Wu et al., 2024; Meng et al., 2024). While frameworks utilizing multiple reward models have emerged, such as reward model ensembling to mitigate overoptimization (Coste et al., 2024) and reward model routing to enhance specialized alignment (Wu and Lu, 2025), their primary focus remains on rectifying hallucinated judgments or providing domain-specific expertise. In contrast, the role of multiple reward models for non-verifiable data involving open-ended generation and nuanced preference signals remains largely underexplored.

3 Method

In this section, we first detail the construction of our Donkey dataset, designed to capture fine-grained preference signals with high absolute quality. Subsequently, we introduce a consensus-driven optimization framework that integrates multiple perspectives to enhance label reliability and model robustness.

3.1 Donkey Dataset Construction

While verifiable domains like mathematics and code generation benefit from objective ground-truth verification, non-verifiable instruction-following tasks present a more formidable challenge due to the inherent ambiguity and subjectivity of their evaluation. Therefore, this work exclusively focuses on these open-ended domains where robust automated assessment remains an unsolved problem. Drawing inspiration from Ge et al.

(2024), which emphasizes the generation of diverse and rich non-verifiable content through persona-driven prompt engineering, we implement a weak-to-strong construction strategy. Specifically, we utilize a relatively smaller model, Gemma-2-9b-it (Team et al., 2024), to generate and filter diverse non-verifiable prompts, while reserving a significantly more capable model for response generation to ensure the outputs accurately reflect the prompt’s intent and quality.

In contrast to previous datasets like UltraFeedback (Cui et al., 2024), which aggregate responses from distinct models to artificially create large quality gaps, our data construction is guided by the insight that high absolute quality is paramount for downstream performance (detailed analyses are provided in Section 5.3). To ensure that both chosen and rejected responses maintain this high standard while exhibiting the subtle distinctions necessary for effective alignment, we utilize a single strong model, GPT-4o, to sample two independent responses for each prompt with a temperature of 0.7, as:

$$\mathcal{D}_{\text{Donkey}} = \{(x_i, y_{i,1}, y_{i,2}) \mid x_i \in \mathcal{P}, y_{i,1}, y_{i,2} \sim \pi_{\text{GPT-4o}}(x_i)\}_{i=1}^N, \quad (1)$$

where \mathcal{P} represents the set of non-verifiable subjective prompts, N denotes the total number of samples, and $y_{i,1}, y_{i,2}$ are the two independent high-quality responses generated by the model $\pi_{\text{GPT-4o}}$. Following the generation, we filter out identical responses and conduct a manual verification on a sampled subset to ensure the absence of factual errors.

3.2 Preference Optimization on Donkey

Single Judge Preference Optimization. Recent advancements in preference optimization have demonstrated significant potential in aligning LLMs with human preferences. In this study, we employ SimPO (Meng et al., 2024) as the default preference optimization algorithm, given its established prominence and efficacy in fine-tuning LLMs. SimPO introduces a length-normalized reward formulation that inherently aligns with the generation likelihood, scaled by a constant β . Additionally, it incorporates a target reward margin $\gamma > 0$ to enforce a minimum separation between the rewards of winning (y_w) and losing (y_l) responses.

Formally, given a prompt x and a pair of responses (y_1, y_2) , the winner y_w and loser y_l are determined based on the score assigned by the judge J :

$$(y_w, y_l) = \begin{cases} (y_1, y_2), & \text{if } J(x, y_1) > J(x, y_2) \\ (y_2, y_1), & \text{otherwise} \end{cases}. \quad (2)$$

The optimization objective is then defined as:

$$\mathcal{L}_{\text{SimPO}}(\pi_\theta) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\frac{\beta}{|y_w|} \log \pi_\theta(y_w|x) - \frac{\beta}{|y_l|} \log \pi_\theta(y_l|x) - \gamma \right) \right]. \quad (3)$$

Consensus and Individual Preference. As shown in Figure 3, given the noticeable disagreement observed among different judges in our initial experiments on our Donkey dataset, we further investigate the effectiveness of integrating multiple external judges together.

To rigorously address label ambiguity, we formalize the partition of our preference dataset based on inter-judge consistency. Let $\mathcal{J} = \{J_1, \dots, J_M\}$ denote the set of M judges. We divide the data into a Consensus Set (\mathcal{D}_c) and an Individual-Specific Set (\mathcal{D}_p) defined as:

$$\begin{aligned} \mathcal{D}_c &= \{(x, y_w, y_l) \mid \forall m, J_m(y_w) > J_m(y_l)\}, \\ \mathcal{D}_p &= \bigcup_{k=1}^M \mathcal{D}_p^{(k)}, \quad \text{where} \\ \mathcal{D}_p^{(k)} &= \{(x, y_w^{(k)}, y_l^{(k)}) \mid J_k(y_w^{(k)}) > J_k(y_l^{(k)}) \\ &\quad \wedge \exists m \in [M], J_m(y_w^{(k)}) < J_m(y_l^{(k)})\}. \end{aligned} \quad (4)$$

Here, \mathcal{D}_c represents the *high-agreement* regime, providing robust, noise-free signals where all judges align with each other. In contrast, \mathcal{D}_p handles *conflicting preferences* by aggregating the specific viewpoints of individual judges. Specifically, within each subset $\mathcal{D}_p^{(k)}$, we anchor the label $(y_w^{(k)}, y_l^{(k)})$ to the perspective of judge J_k when a global consensus is absent. This allows the model to learn from diverse, subjective viewpoints even without a universally agreed-upon preference.

Accordingly, for each specific judge k , we can rewrite the optimization objective as:

$$\begin{aligned} \mathcal{L}_k(\pi_\theta) &= -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}_c} \left[\log \sigma \left(\frac{\beta}{|y_w|} \log \pi_\theta(y_w|x) \right. \right. \\ &\quad \left. \left. - \frac{\beta}{|y_l|} \log \pi_\theta(y_l|x) - \gamma \right) \right] \\ &\quad - \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}_p^{(k)}} \left[\log \sigma \left(\frac{\beta}{|y_w|} \log \pi_\theta(y_w|x) \right. \right. \\ &\quad \left. \left. - \frac{\beta}{|y_l|} \log \pi_\theta(y_l|x) - \gamma \right) \right] \end{aligned}, \quad (5)$$

where the first term aligns the model with general consensus (\mathcal{D}_c), while the second term incorporates the specific individual perspectives of judge k ($\mathcal{D}_p^{(k)}$).

4 Experiments

4.1 Experimental Settings

We employ the lightweight Gemma-2-2B-it (Team et al., 2024) as the default backbone model due to its computational efficiency and strong baseline performance. For preference optimization, we perform a grid search over the hyperparameters, setting $\beta \in \{2.5, 10\}$ and $\gamma \in \{1.0, 1.5, 3\}$. All models are trained with a global batch size of 128 distributed across four GPUs. We use a learning rate of 1×10^{-6} , following a cosine decay scheduler with a warmup ratio of 0.1, and each training run is conducted for 6 epochs.

Table 1: Overall performance (%) comparison across different benchmarks. "General Chat" includes AlpacaEval 2 (AE2 for short, with LC as the metric) and Arena-Hard (AH for short, with WR as the metric), while "Downstream Tasks" cover Math500, MMLU-Redux, and PopQA. Best scores are **bolded**, and second-best are underlined. We also provide a further study on data quality and preference gaps in Section 5.3.

Method	Size	General Chat			Downstream Tasks				Overall
		AE2 LC	AH WR	Avg.	Math500	MMLU-R	PopQA	Avg.	
<i>Baselines</i>									
Base	-	37.69	47.5	42.60	19.6	56.1	<u>22.4</u>	32.70	36.68
UltraFeedback	60k	40.13	41.9	41.02	6.6	55.8	14.6	25.67	31.81
DPO-Mix	7k	43.01	38.2	40.61	7.2	52.4	21.0	26.87	32.36
<i>Donkey Variants (Ours)</i>									
RandomSFT	10k	42.37	50.6	46.49	25.4	40.6	21.4	29.13	36.07
RandomSimPO	10k	43.01	49.7	46.36	18.0	54.1	23.0	31.70	37.56
Skywork-Reward	10k	48.20	64.6	56.40	21.4	57.6	21.8	<u>33.60</u>	42.72
ArmoRM	10k	<u>49.29</u>	<u>60.0</u>	54.65	<u>21.8</u>	57.4	21.8	33.67	42.05
ConsensusRM	6k	52.21	<u>59.5</u>	<u>55.86</u>	<u>21.4</u>	57.6	20.8	33.27	<u>42.30</u>

4.2 Evaluation Metrics

Our experimental evaluation employs a comprehensive set of metrics to assess model performance across various dimensions. For subjective benchmarks, we primarily focus on **AlpacaEval 2** (Dubois et al., 2025), an LLM-based automatic evaluation benchmark utilizing prompts from AlpacaFarm (Dubois et al., 2024) where model responses are compared against GPT-4-Turbo references and a length-controlled win rate (LC) is used as the metric, and **Arena-Hard** (Li et al., 2024a), which features 500 challenging user queries, where model responses are compared against GPT-4-0314 references and a standard win rate (WR) is used as the metric. Complementing these, we assess objective performance using **MMLU-Redux** (Gema et al., 2025) and **PopQA** (Mallen et al., 2023) to test general and long-tail knowledge, respectively, while also including **Math500** (Hendrycks et al., 2021) to specifically evaluate mathematical reasoning capabilities.

4.3 Main Baselines

In our comparisons, we include a diverse set of baselines trained on both widely-used public datasets and variations of our constructed **Donkey** dataset. Specifically, we evaluate: (1) **UltraFeedback** (Cui et al., 2024): a large-scale (60k), widely-used dataset containing diverse instructions; (2) **DPO-Mix** (Argilla, 2024) a small (7k), high-quality preference dataset constructed by filtering only top-rated chosen responses from open-source collections, including UltraFeedback; (3) **RandomSFT**: a model trained via supervised fine-tuning on our Donkey dataset using randomly selected prompt-response pairs, serving as a non-preference baseline; (4) **RandomSimPO**: a model trained on our Donkey dataset with randomly assigned preference signals to establish a lower bound for optimization effectiveness; (5) **Skywork-Reward** (Liu et al., 2025): a model optimized on our Donkey dataset using preference signals labeled exclusively by the Skywork-Reward-V2-Llama-3.1 reward model; (6) **ArmoRM** (Wang et al., 2024a): a

model trained using preference signals provided exclusively by ArmoRM-Llama3-8B-v0.1; and (7) **ConsensusRM**: Our proposed high-confidence variant, trained on a subset of our Donkey dataset where preference labels reflect a consensus between both ArmoRM and Skywork-Reward.

5 Main Results

5.1 Performance Comparison

As shown in Table 1, we observe that models trained on our Donkey dataset variants consistently outperform both the base model and strong external baselines. In terms of subjective benchmarks, our methods achieve substantial gains over existing baselines. Specifically, ConsensusRM achieves an AlpacaEval 2 score of 52.21%, significantly surpassing UltraFeedback (40.13%) and DPO-Mix (43.01%). Similarly, on the challenging Arena-Hard benchmark, the Skywork-Reward model reaches a score of 64.6%, outperforming UltraFeedback (41.9%) and DPO-Mix (38.2%) by a large margin. These results indicate that our constructed Donkey dataset is superior to existing datasets, enabling the effective optimization of advanced LLMs.

Regarding objective benchmarks, our models demonstrate robustness on downstream tasks without suffering from the "alignment tax" often observed in other methods. For instance, UltraFeedback shows a significant regression in mathematical reasoning, dropping to a Math500 score of 6.6% compared to the Base model's 19.6%. In contrast, our Donkey variants maintain or even improve performance on these tasks: RandomSFT achieves 25.4% on Math500, and the preference-optimized models (Skywork and ConsensusRM) maintain competitive scores (around 21.4%) while slightly improving on general knowledge (MMLU-Redux) compared to the base model. This suggests that our approach effectively balances conversational style alignment without compromising the model's fundamental reasoning and factual capabilities.

Finally, we analyze the impact of different preference

Table 2: Ablation study of ConsensusRM. We analyze the impact of signal validity, compare consensus-based filtration against single-judge downsampling, and evaluate the necessity of strict agreement versus random selection.

Method	AE2 LC	AH WR	Avg.
ConsensusRM	52.21	59.5	55.86
ConsensusRM-Reverse	36.63	36.6	36.62
Skywork-Reward-Downsample	47.15	57.1	52.13
ArmoRM-Downsample	49.80	54.1	51.95
ConsensusRM+Random	51.23	58.0	54.62
Skywork-Reward	48.20	64.6	56.40
ArmoRM	49.29	60.0	54.65

signals and data efficiency. We find that models trained with different state-of-the-art reward models, such as Skywork-Reward and ArmoRM, yield comparable improvements, demonstrating the robustness of training on our Donkey. Notably, ConsensusRM, which utilizes only the subset of data where judges agree with each other (around 6k samples), achieves competitive overall performance (42.30%). This is comparable to models trained on the full Donkey dataset and significantly higher than baselines trained on much larger data, highlighting that leveraging consensus agreement allows for effective training with significantly less data.

5.2 Ablation Study on Consensus Mechanism

To further validate the effectiveness of our proposed consensus strategy, we conducted a comprehensive ablation study as detailed in Table 2 and an example in Figure 11. First, to verify the validity of our preference signals, we trained the ConsensusRM-Reverse model by deliberately reversing the chosen and rejected labels derived from the consensus preferences. This reversal resulted in a catastrophic performance collapse, with the AlpacaEval 2 score plummeting from 52.21% to 36.63% and the overall average score dropping to 36.62%; this sharp decline confirms that our performance gains stem fundamentally from the accurate direction of the preference signals rather than merely the distribution of the prompts. Next, to disentangle the benefits of data filtration from the quality of the consensus mechanism itself, we compared our method against Skywork-Reward-Downsample and ArmoRM-Downsample, where training data was reduced to the same 6k scale using rankings from single reward models. Our ConsensusRM (Avg. 55.86%) consistently outperforms these single-judge baselines, which achieved averages of 52.13% and 51.95%, respectively. This performance gap indicates that while individual reward models are strong, relying on a single judge introduces idiosyncratic noise, whereas the intersectional agreement in our method effectively reduces such biases to retain only the highest-fidelity samples. Finally, we experiment with ConsensusRM+Random, where the model is trained on the entire dataset but assigns random preference labels specifically to the Individual-Specific Set in Eq. (4), while keeping the labels in the Consensus Set intact. This variant yields a score of 54.62%, which is lower than the

Table 3: Ablation study on data quality using different data sources. “UF” is short for UltraFeedback and “1B” is short for Llama-3.2-1B-Instruct. Quality and Gap represent scores on a scale of 0–10.

Method	Quality	Gap	AE2 LC	Math
Base	-	-	37.70	19.6
UF (Non-Verifiable)	8.64	5.51	53.20	12.8
UF (Verifiable)	8.44	5.43	43.85	14.0
1B (Weak Pairs)	7.04	4.80	43.29	16.6
GPT-4o & 1B	-	5.48	52.40	16.8
Default (Ours)	9.33	1.20	52.20	21.4

valid single-judge baselines (e.g., Skywork-Reward at 56.40%). This performance gap confirms that real individual preferences contain valid, distinguishable signals rather than random noise. However, the marginal gain from these individual perspectives is limited, indicating that their contribution to the final model capability is weaker than that of the consensus preference signal, which serves as the primary driver for robust alignment.

5.3 Further Study on Quality Gap of Data

To investigate the influence of the preference data quality gap, we constructed several baselines. First, we randomly sampled 10k verifiable and 10k non-verifiable examples from UltraFeedback (UF). Additionally, utilizing the prompts from our Donkey dataset, we generated response pairs solely using Llama-3.2-1B-Instruct (1B for short) (Grattafiori et al., 2024), and constructed a high-gap configuration by pairing our original GPT-4o responses with these weak Llama-3.2-1B outputs (denoted as GPT-4o & 1B). As shown in Table 3, simply enforcing a large preference gap is insufficient. The 1B setting yields poor results due to low absolute quality, while the UF subsets and the GPT-4o & 1B configuration, despite having large quality gap and achieving high AlpacaEval 2 scores, lead to significant degradation on Math500. This suggests that relying on clear distinctions between strong and weak responses enhances general instruction-following capabilities but incurs a high cost on downstream reasoning tasks. In contrast, our Default setting achieves the highest absolute quality (9.33) with a much smaller gap (1.20), resulting in the best reasoning accuracy on Math500 (21.4%) while maintaining competitive chat performance on AlpacaE-

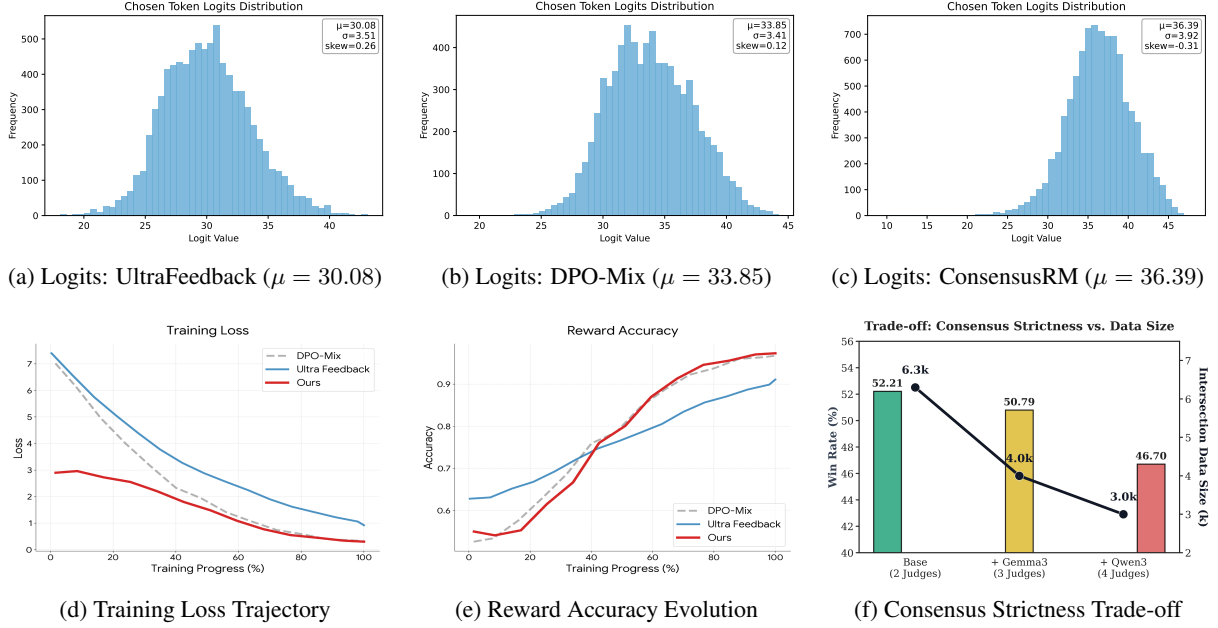


Figure 4: (a-c) Logit shift of responses on AlpacaEval 2 from models trained on different datasets. (d-e) Training loss and reward accuracy trajectories. (f) Trade-off between consensus strictness and data size.

val2. This indicates that high absolute quality combined with fine-grained relative gaps effectively mitigates the alignment tax on reasoning capabilities.

5.4 Logit Shift Analysis

To quantify the underlying factors contributing to the model’s performance improvement on AlpacaEval 2, we conduct a logit shift analysis. This method allows us to examine the distributional changes in the model’s output probabilities, providing a metric to measure how the alignment process alters the model’s confidence and decision sharpness on unseen test data. We randomly sampled 20 prompts from the dataset and generated responses using the different models. For each generated response, we extracted the unnormalized logits $z_i \in \mathbb{R}$ for the chosen tokens before the softmax operation. To perform a global statistical analysis, we flatten all collected logits across the generated sequences into a single one-dimensional vector:

$$\mathbf{z}_{\text{flat}} = \{z_1^{(1)}, z_2^{(1)}, \dots, z_{m_k}^{(k)}\}, \quad (6)$$

where for the i -th sample among k samples ($i \leq k$), $z_j^{(i)}$ denotes the logit for the j -th token and m_i denotes the number of tokens. Subsequently, we quantify the asymmetry of the flattened logit distribution by computing the skewness:

$$\text{Skewness} = \frac{1}{n} \sum_{i=1}^n \left(\frac{z_i - \mu}{\sigma} \right)^3. \quad (7)$$

As visualized in the distribution plots in Figure 4(a-c), the model trained on ConsensusRM exhibits a clear distributional shift relative to the UltraFeedback and DPO-Mix baselines. Specifically, the UltraFeedback-trained model has a lower mean chosen-token logit of

$\mu = 30.08$ and a slightly positive skewness of 0.26, indicating a mildly right-skewed distribution with a longer tail toward higher values. In contrast, the ConsensusRM model produces a substantial rightward shift, increasing the mean logit to $\mu = 36.39$. Its skewness also changes to -0.31 , indicating a mildly left-skewed distribution in which a larger share of logits is concentrated in the higher-value region, with a longer tail toward lower values. Overall, this distributional change suggests that our consensus-based alignment strategy encourages the model to assign higher scores to chosen tokens, resulting in sharper token-level preferences and more decisive generations.

5.5 Training Dynamics and Stability

To analyze the optimization process, we examine the training trajectories visualized in Figures 4(d) and (e). As shown in Figure 4(d), the model trained on Donkey (red line) exhibits the lowest training loss throughout the process compared to DPO-Mix and UltraFeedback. Furthermore, Figure 4(e) shows that while UltraFeedback begins with higher initial accuracy, the model trained on Donkey rapidly improves, eventually surpassing both baselines to reach the highest final reward accuracy. These results suggest that our Donkey facilitates stable convergence and provides learnable preference signals that enable effective learning.

5.6 Consensus Strictness vs. Data Size

In this subsection, we investigate the trade-off between the strictness of consensus filtering and the resulting dataset size. As shown in Figure 4(f), we observe a clear tension between filtration strictness and data availability. Specifically, the baseline configuration requiring

agreement from two judges yields 6.3k samples and achieves the highest LC of 52.21% in AlpacaEval 2. However, enforcing stricter consensus by incorporating a third and fourth judge drastically reduces the dataset size to 4.0k and 3.0k, respectively. This reduction correlates with a decline in performance, as the LC falls to 50.79% and further to 46.70%. These results indicate that while a stricter filtration tends to improve label quality, excessive strictness induces severe data scarcity that hampers model generalization. Consequently, the two-judge consensus emerges as the optimal configuration, striking a balance between maintaining high preference quality and preserving sufficient data volume. Future work could explore scaling up the initial pool of candidate responses or adopting consensus-based weighting mechanisms to maintain sufficient dataset sizes even under stricter consensus protocols, thereby maximizing both data quality and quantity.

Table 4: General chat performance (%) comparison using Qwen2.5-3B-Instruct as the backbone model.

Method	AE2 LC	AH WR	Avg.
Base	24.21	54.6	39.41
DPO-Mix	20.40	58.1	39.25
ConsensusRM	27.04	58.0	42.52

5.7 Generalizability Analysis

To further verify the generalizability of our approach across different model series, we applied our training strategy to Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct (Qwen et al., 2025). As shown in Tables 4 and 5, the improvement trend remains consistent across both backbones. On Qwen2.5-3B-Instruct, DPO-Mix does not always guarantee performance improvements; specifically, it shows a regression in the average score (39.25%) compared to the base model (39.41%), with a notable drop in AlpacaEval 2 performance. In contrast, ConsensusRM demonstrates consistent effectiveness, achieving the highest average score of 42.52%. Similarly, on the larger Qwen2.5-7B-Instruct backbone, ConsensusRM again outperforms both the base model and DPO-Mix, substantially improving AlpacaEval2.0 while also improving Math500. These results further confirm that the high-quality preference signals in the Donkey dataset are model-agnostic and can effectively enhance the capabilities of different LLM backbones across both general chat and reasoning settings.

Table 5: Results on the Qwen2.5-7B-Instruct backbone.

Method	AE2 LC	Math500	Avg.
Base	36.06	72.0	54.03
DPO-Mix	34.45	69.6	52.03
ConsensusRM	43.04	75.0	59.02

5.8 Multi-Judge Results

Our default judge for subjective evaluation is Qwen2-72B-Instruct (Yang et al., 2024), which is both strong and cost-effective. In our measurements, Qwen2-72B-Instruct achieves a 67.9% agreement rate with human judgments, compared with 65.7% for a single human annotator reported in AlpacaEval 2.0 (Dubois et al., 2025). We also observe highly similar trends when using GPT-4-Turbo as the judge (Yin et al., 2025c). To further validate the robustness of our evaluation, we additionally report results with Llama3.1-70B-Instruct (Grattafiori et al., 2024), which achieves a human agreement rate of 69.1%. As shown in Table 6, the overall improvement trend remains consistent across both judges. In particular, compared with the base model, all reward-model variants achieve clear gains, while ConsensusRM maintains competitive performance across both judges. At the same time, its average response length remains moderate and does not show undesirable overlong behavior.

Table 6: Multi-judge results in AlpacaEval 2.

Method	Llama (LC)	Qwen (LC)	Avg. Len
Base	39.35	37.69	2079
ArmoRM	50.95	49.29	1863
Skywork-Reward	52.80	48.20	2792
ConsensusRM	51.41	52.21	2427

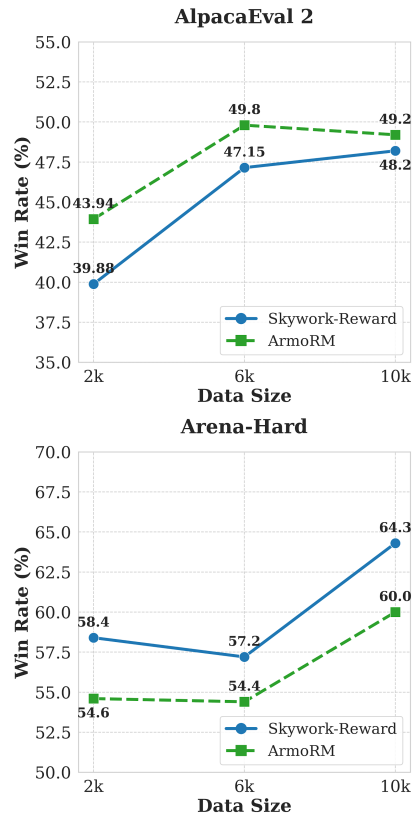


Figure 5: Data scaling law of different reward models.

5.9 Scaling Law Analysis of Data Size

To investigate the impact of dataset size on model performance, we analyze the training trends across different data scales (2k, 6k, 10k) using both Skywork-Reward and ArmoRM signals, as illustrated in Figure 5. On AlpacaEval 2, we observe a rapid performance growth when increasing the data size from 2k to 6k, where the Skywork-Reward model improves significantly from 39.88% to 47.15% and ArmoRM peaks at 49.8%; however, these gains tend to saturate or plateau beyond the 6k point. In contrast, the trends on the more challenging Arena-Hard benchmark differ, showing a substantial jump primarily when scaling to 10k, where the Skywork-Reward model reaches its peak WR of 64.3%. Consequently, regarding the average performance across both benchmarks, we observe a consistent upward trend distinct from the fluctuations of individual metrics. The Skywork-Reward model’s average score rises steadily from 49.14% at 2k to 52.18% at 6k, finally reaching 56.25% at 10k, while ArmoRM similarly demonstrates continuous overall growth. This indicates that while performance on AlpacaEval 2 tends to saturate at moderate data scales, the more challenging Arena-Hard benchmark continues to benefit significantly from increased data volume, requiring larger datasets to realize substantial performance gains.

6 Conclusion

In this paper, we introduce Donkey, a high-quality dataset capturing nuanced preference signals where responses are both of high quality and differ only subjectively. Experiments demonstrate that models trained on Donkey significantly outperform those trained on widely-used datasets, effectively enhancing subjective alignment while mitigating the “alignment tax” on objective reasoning tasks. Crucially, we reveal that distilling consensus preferences offers a highly data-efficient optimization path that matches or exceeds strong single-judge baselines. These findings underscore the importance of leveraging nuanced preference signals and the consensus of multiple judges for advancing subjective LLM alignment and downstream capabilities.

Limitations

Despite the promising results, our study may have two limitations. First, our experiments were conducted on a specific range of model sizes; future research is needed to investigate whether the data efficiency and consensus benefits we observed scale effectively to significantly larger foundation models (e.g., 70B+). Second, while we successfully mitigated the alignment tax on standard benchmarks, our current exploration of reasoning domains is not exhaustive; future work could further dive into more complex and verifiable areas, such as advanced mathematical problem-solving and repository-level code generation, to fully assess the potential of consensus-driven alignment.

References

- Argilla. 2024. Dpo mix 7k dataset. <https://huggingface.co/datasets/argilla/dpo-mix-7k>.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.
- Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. 2023. Benchmarking foundation models with language-model-as-an-examiner. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.
- Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Sriniwasan, Tianyi Zhou, Heng Huang, et al. 2024a. AlpacaGasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*.
- Wei-Lin Chen, Zhepei Wei, Xinyu Zhu, Shi Feng, and Yu Meng. 2025. Do LLM evaluators prefer themselves for a reason? *arXiv preprint arXiv:2504.03846*.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024b. Self-play fine-tuning converts weak language models to strong language models.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90%* chatgpt quality.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martić, Shane Legg, and Dario Amodei. 2017. Deep

- reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv e-prints*, pages arXiv–2210.
- 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*.
- Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2024. [Reward model ensembles help mitigate overoptimization](#).
- 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.
- Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. 2023. Rephrase and respond: Let large language models ask better questions for themselves. *arXiv preprint arXiv:2311.04205*.
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2025. [Length-controlled alpacaeval: A simple way to debias automatic evaluators](#).
- Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2024. Alpaca-farm: A simulation framework for methods that learn from human feedback.
- Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André Martins, Graham Neubig, Ankush Garg, Jonathan Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1066–1083, Singapore. Association for Computational Linguistics.
- Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.
- Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.
- Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. 2025. [Are we done with mmlu?](#)
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Al-lonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esioibu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yearly, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Rapparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal

Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, DingKang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Sweet, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov,

Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihalescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiao Cheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. [The llama 3 herd of models](#).

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the math dataset](#).

Kexin Huang, Junkang Wu, Ziqian Chen, Xue Wang, Jinyang Gao, Bolin Ding, Jiancan Wu, Xiangnan He, and Xiang Wang. 2025. [Larger or smaller reward margins to select preferences for LLM alignment?](#) In *Forty-second International Conference on Machine Learning*.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: Synthie and the case of information extraction. *arXiv preprint arXiv:2303.04132*.

Nathan Lambert, Valentina Pyatkin, Jacob Daniel Morrison, Lester James Validad Miranda, Bill Yuchen

- Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hanna Hajishirzi. 2024. RewardBench: Evaluating reward models for language modeling. *ArXiv*, abs/2403.13787.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback.
- Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024a. From live data to high-quality benchmarks: The Arena-Hard pipeline.
- Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. 2024b. Self-alignment with instruction back-translation. In *The Twelfth International Conference on Learning Representations*.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023a. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.
- Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023b. Textbooks are all you need ii: phi-1.5 technical report.
- Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. 2024. Mitigating the alignment tax of RLHF. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.
- Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. 2023. Tinygsm: achieving > 80% on gsm8k with small language models. *arXiv preprint arXiv:2312.09241*.
- Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiakai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. 2025. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*.
- Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.
- Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. 2025. Rewardbench 2: Advancing reward model evaluation. <https://huggingface.co/spaces/allenai/reward-bench>.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [When not to trust language models: Investigating effectiveness of parametric and non-parametric memories](#).
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward.
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.
- OpenAI. 2023. [GPT-4 technical report](#).
- 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.
- Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2023. Rephrase, augment, reason: Visual grounding of questions for vision-language models. *arXiv preprint arXiv:2310.05861*.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. [Qwen2.5 technical report](#).
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.
- Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2023. Branch-solve-merge improves large language model evaluation and generation. *arXiv preprint arXiv:2310.15123*.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

- 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.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshov, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonnell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iversen, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshtir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. [Gemma 2: Improving open language models at a practical size.](#)
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.
- Sanh Victor, Webson Albert, Raffel Colin, Bach Stephen, Sutawika Lintang, Alyafeai Zaid, Chaffin Antoine, Stiegler Arnaud, Raja Arun, Dey Manan, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.
- Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024a. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.
- Shiqi Wang, Zhengze Zhang, Rui Zhao, Fei Tan, and Nguyen Cam-Tu. 2024b. Reward difference optimization for sample reweighting in offline rlhf. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2109–2123.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2025a. InstructRAG: Instructing retrieval-augmented generation via self-synthesized rationales. In *The Thirteenth International Conference on Learning Representations*.
- Zhepei Wei, Xiao Yang, Kai Sun, Jiaqi Wang, Rulin Shao, Sean Chen, Mohammad Kachuee, Teja Golapudi, Tony Liao, Nicolas Scheffer, et al. 2025b. TruthRL: Incentivizing truthful LLMs via reinforcement learning. *arXiv preprint arXiv:2509.25760*.
- Xinle Wu and Yao Lu. 2025. [Reward model routing in alignment.](#)
- Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2024. Self-play preference optimization for language model alignment.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang,

Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. [Qwen2 technical report](#).

Shangjian Yin, Peijie Huang, JiaTian Chen, Haojing Huang, and Yuhong Xu. 2025a. ECLM: Entity level language model for spoken language understanding with chain of intent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Shangjian Yin, Peijie Huang, and Yuhong Xu. 2025b. MIDLM: Multi-intent detection with bidirectional large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*.

Shangjian Yin, Shining Liang, Wenbiao Ding, Yuli Qian, Zhouxing Shi, Hongzhi Li, and Yutao Xie. 2026. [Pika: Expert-level synthetic datasets for post-training alignment from scratch](#).

Shangjian Yin, Zhepei Wei, Xinyu Zhu, Wei-Lin Chen, and Yu Meng. 2025c. [Aligning large language models via fully self-synthetic data](#).

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *ArXiv preprint, abs/2308.01825*.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. 2025. The surprising effectiveness of negative reinforcement in LLM reasoning. *arXiv preprint arXiv:2506.01347*.

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*.

A Appendix

A.1 Dataset Statistics

We analyze the token length distributions of our proposed Donkey dataset compared to existing popular datasets, UltraFeedback and DPO-Mix. As shown in Table 7, while Donkey utilizes relatively concise prompts (Avg. 55.03), its responses are significantly longer (Avg. ~ 884 tokens) than those in UltraFeedback (~ 370) and DPO-Mix (~ 296). This substantial increase in response length reflects the dense information content and detailed reasoning structures characteristic of the Donkey dataset.

Table 7: Average token lengths for Prompts and Responses across Donkey, UltraFeedback, and DPO-Mix.

Dataset	Prompt	Response 1	Response 2
Donkey (Ours)	55.03	884.07	884.45
UltraFeedback	106.73	418.37	327.18
DPO-Mix	117.83	292.65	300.49

A.2 Prompt Templates for Quality Assessment

In our experiments, we employ Qwen2-72B-Instruct as the LLM-as-a-Judge. We use two distinct prompts: one to evaluate the relative quality difference (gap) between responses (Figure 6), and another to assess the absolute quality of individual responses (Figure 7).

A.3 Dataset Example and Reward Model Disagreement

In this section, we present a representative case from our dataset involving an analysis of construction plans regarding OSHA safety violations. Notably, this instance exhibits a disagreement between reward models: Skywork-Reward favors Response 1 ($R_1 > R_2$), whereas ArmoRM prefers Response 2 ($R_2 > R_1$). The prompt and candidate responses for this case are detailed in Figures 8, 9, and 10, while Figure 11 provides another qualitative comparison illustrating the differences between ConsensusRM-Reverse and ConsensusRM.

```
Quality Gap Assessment Prompt

You are an expert evaluator assessing the quality gap (absolute difference) between two
responses to the same prompt.

**Original Instruction:**
{instruction}

**Response 1:**
{response_1}

**Response 2:**
{response_2}

Please evaluate the ABSOLUTE quality gap between the chosen and rejected responses on a
scale of 0 to 10, where:
- 0: Both responses are identical or nearly identical in quality (no meaningful difference)
- 1-3: Small quality gap, minor differences
- 4-6: Medium quality gap, noticeable differences
- 7-9: Large quality gap, one is significantly better than the other
- 10: Huge quality gap, one is vastly superior to the other

Consider the following criteria:
1. Accuracy and correctness
2. Completeness and thoroughness
3. Clarity and coherence
4. Relevance to the prompt
5. Helpfulness and usefulness

Respond with ONLY a single number between 0 and 10 representing the absolute difference.
Do not include any explanation, just the number.
```

Figure 6: The prompt template used for calculating the relative Quality Gap using Qwen2-72B-Instruct.

```
Absolute Quality Assessment Prompt

You are an expert evaluator assessing the absolute quality of a response.

**Instruction:**
{instruction}

**Response:**
{response}

Rate the quality on a scale of 1-10:
- 1-2: Very poor
- 3-4: Poor
- 5-6: Moderate
- 7-8: Good
- 9-10: Excellent

Respond with ONLY a single number (1-10), nothing else.
```

Figure 7: The prompt template used for assessing Absolute Quality using Qwen2-72B-Instruct.

```
Dataset Example Part 1: Synthesized Prompt

"Analyze the following construction plans and blueprints for potential safety violations,
specifically focusing on OSHA regulations regarding scaffolding and fall protection. Please
provide a detailed report outlining any discrepancies or areas of concern, including relevant
code citations and potential legal ramifications for the construction company."
```

Figure 8: The synthesized prompt for the OSHA safety analysis example.

Dataset Example Part 2: Response 1 (Preferred by Skywork-Reward)

Certainly! However, I don't have the ability to analyze specific construction plans or blueprints directly, as I cannot view or interpret files or images. However, I can guide you on how to analyze them yourself in light of OSHA regulations, and provide you with a framework for identifying potential safety violations related to scaffolding and fall protection.

Below is an outline you can follow to analyze the provided construction plans and blueprints for compliance with OSHA standards:

-

****Framework for Analyzing Construction Plans for OSHA Compliance****

****1. Scaffolding Safety****

Scaffolding is one of the most common areas where OSHA violations occur. Review the plans with the following in mind:

- **OSHA Requirements (29 CFR 1926 Subpart L):**
 - **Proper Design and Load Capacity:** Scaffolds must be designed to support at least four times the maximum intended load. Check the load ratings and ensure the plans account for potential weight, including workers, tools, and materials.
 - **Access to Scaffolding:** Safe access, such as ladders or stair towers, must be provided. Evaluate whether the plans include these access points.
 - **Guardrails and Fall Protection:** Guardrails must be installed on scaffolds more than 10 feet above a lower level. Verify that the blueprints specify guardrails that meet OSHA height and strength requirements.
 - **Planking and Platforms:** Fully planked platforms are required, with no more than a 1-inch gap between planks. Inspect the construction plans for proper platform specifications.
 - **Stability and Anchorage:** Scaffolding must be secured to prevent tipping. Ensure the blueprints show proper anchorage points and stabilization methods.
- **Common Violations:**
 - Missing or inadequate guardrails.
 - Scaffolds not properly secured to the structure.
 - Insufficient access or improper use of ladders.
 - Overloading scaffolds beyond their capacity.

****2. Fall Protection****

Falls are a leading cause of injuries and fatalities in construction. Carefully review the plans for compliance with OSHA fall protection standards.

- **OSHA Requirements (29 CFR 1926 Subpart M):**
 - **Fall Protection Threshold:** Fall protection must be provided at elevations of 6 feet or more in the construction industry. Check whether the plans include fall protection systems for all areas meeting this threshold.
 - **Guardrails, Safety Nets, and Personal Fall Arrest Systems (PFAS):** Ensure the plans specify one or more of these systems in all areas where workers will be exposed to fall hazards.
 - **Anchor Points:** PFAS must be anchored to structural members capable of supporting at least 5,000 pounds per worker. Look for detailed specifications on anchor points in the plans.
 - **Openings and Edges:** Holes in floors, walls, or roofs must be covered or protected by guardrails. Verify that the plans address this requirement.
- **Common Violations:**
 - Lack of guardrails on elevated work surfaces or around openings.
 - Insufficient or improperly designed anchor points for PFAS.
 - Missing fall protection on leading edges or roof work.

(Continued in text...)

Dataset Example Part 2: Response 1 (Continued)

****3. General Site Safety****

While scaffolding and fall protection are the focus, also consider general site safety provisions.

- **OSHA Requirements:** - **Training:** Workers must be trained to recognize and mitigate fall hazards (29 CFR 1926.503). - **Inspections:** Scaffolds and fall protection systems must be inspected before each shift (29 CFR 1926.451(f)(3)). - **Overhead Protection:** Look for plans that include protection from falling objects, such as toeboards, screens, or debris nets.
- **Common Violations:** - Lack of worker training or certification. - Failure to include provisions for overhead protection. - Inadequate inspection protocols.

****Key Steps for Your Analysis****

1. **Review Blueprint Details:** Look for specific notes, dimensions, and specifications related to scaffolds, guardrails, anchor points, and fall protection systems. 2. **Compare Against OSHA Standards:** Use OSHA's construction safety standards (29 CFR Part 1926) as a reference to identify any discrepancies. 3. **Document Violations:** For each area of concern, note the specific OSHA regulation that is being violated and describe the issue. 4. **Assess Legal Ramifications:** Highlight the potential consequences of non-compliance, such as fines, project delays, or increased liability in the event of worker injuries.

****Example Report Excerpt****

****Issue #1: Missing Guardrails on Scaffolding**** - **Regulation:** OSHA 29 CFR 1926.451(g)(1) - **Description:** The blueprints for scaffolding on the west elevation do not include guardrails on platforms above 10 feet. This is a violation of OSHA's requirement for fall protection on scaffolds. - **Potential Consequences:** Failure to install guardrails could result in fines of up to \$15,625 per violation and increased liability in the event of worker injuries.

****Issue #2: Inadequate Anchor Points for PFAS**** - **Regulation:** OSHA 29 CFR 1926.502(d)(15) - **Description:** The plans indicate anchor points for personal fall arrest systems, but no specifications are provided to confirm that they meet the 5,000-pound strength requirement. This creates a potential safety hazard for workers using PFAS. - **Potential Consequences:** Non-compliance could lead to serious worker injuries or fatalities, resulting in legal action and penalties.

If you'd like, I can help you create a more detailed analysis template or explain specific OSHA regulations further. Let me know!

Figure 9: Response 1. This response provides a structured framework and examples, preferred by Skywork-Reward.

Dataset Example Part 3: Response 2 (Preferred by ArmoRM)

Certainly, I can provide guidance on how to analyze construction plans and blueprints for potential safety violations regarding scaffolding and fall protection under OSHA regulations. However, since you didn't attach or describe the specific plans, I will outline a general approach to conducting this type of analysis. If you provide further details or specific elements from the plans, I can tailor the guidance accordingly.

****General Approach to Analyzing Construction Plans for Safety Violations****

1. **Understand OSHA Standards for Scaffolding and Fall Protection** - Review OSHA's scaffolding standards (29 CFR 1926 Subpart L) and fall protection standards (29 CFR 1926 Subpart M). - Key scaffolding requirements include: - Proper design and construction to support intended loads (1926.451(a)). - Guardrails, midrails, and toeboards to prevent falls (1926.451(g)). - Safe access to scaffolds (ladders, stair towers, or equivalent) (1926.451(e)). - Regular inspections by a competent person (1926.451(f)(3)). - Key fall protection requirements include: - Fall protection for workers at heights of 6 feet or more in construction (1926.501(b)). - Personal fall arrest systems, guardrails, or safety nets where required (1926.502). - Anchor points capable of supporting 5,000 pounds per worker (1926.502(d)(15)). - Ensure that workers are trained in scaffold use and fall protection (1926.454).
2. **Review the Construction Plans and Blueprints - Scaffolding:** - Look for indications of scaffold placement, type, and dimensions. Ensure scaffolds are designed to handle the loads specified in the plans. - Verify whether guardrails, midrails, and toeboards are included in the design. - Check for adequate access points (e.g., ladders or stairs) and verify that these comply with OSHA standards. - **Fall Protection:** - Identify areas where workers will be exposed to fall hazards (e.g., open edges, leading edges, rooflines, or elevated work surfaces). - Confirm that guardrails, safety nets, or personal fall arrest systems are specified in these areas. - Verify anchor point locations and ensure they meet strength requirements. - **Other Considerations:** - Check for the inclusion of tie-offs for fall arrest systems. - Look for any potential obstructions or design elements that could interfere with fall protection systems.
3. **Identify and Document Potential Violations** - Compare the plans and blueprints against OSHA standards. - Note any discrepancies or missing safety features, such as: - Missing or inadequate fall protection systems. - Scaffold designs that lack guardrails, toeboards, or proper access. - Anchor points not meeting strength requirements. - Lack of compliance with load-bearing requirements for scaffolds. - Document the location and nature of each potential violation.
4. **Cite Relevant OSHA Regulations** - For each identified discrepancy, cite the specific OSHA standard that applies. For example: - Missing guardrails on a scaffold: 1926.451(g)(1). - Absence of fall protection at an unprotected edge: 1926.501(b)(1). - Insufficient strength of anchor points: 1926.502(d)(15).
5. **Assess Potential Legal Ramifications** - Non-compliance with OSHA standards can result in: - Fines and penalties for the construction company. - Stop-work orders or project delays. - Increased liability in the event of worker injuries or fatalities. - Damage to the company's reputation and potential legal action from injured workers or their families.
6. **Provide Recommendations** - Suggest corrective actions to address the identified violations. - Recommend a review of the plans by a certified safety professional or engineer to ensure compliance. - Emphasize the importance of worker training and regular safety inspections.

****Sample Report Outline****

- **1. Executive Summary**** - Brief overview of findings. - Summary of key violations and areas of concern.
- **2. Analysis of Scaffolding Design**** - Description of scaffolding plans. - Identified violations with OSHA citations. - Recommendations for corrective actions.
- **3. Analysis of Fall Protection Measures**** - Description of fall hazard areas. - Identified violations with OSHA citations. - Recommendations for corrective actions.
- **4. Potential Legal and Financial Ramifications**** - Overview of risks associated with non-compliance.
- **5. Conclusion**** - Final summary of findings and recommendations.
- **6. Appendices**** - Copies of relevant OSHA standards. - Annotated blueprints highlighting areas of concern.

****Next Steps****

If you can share more specific details about the construction plans or blueprints (e.g., scaffold type, worksite height, or fall hazard areas), I can provide a more detailed analysis and tailored recommendations.

Figure 10: Response 2. This response provides a general approach and checklist, preferred by ArmoRM.

How do I wrap a present neatly?

Response 1:

Score: 83 / 100

Here's a step-by-step guide to wrapping presents neatly: **Materials:** * Wrapping paper * Scissors * Tape (clear or colored) * Ruler or straight edge * Ribbon (optional) * Gift tag (optional) **Steps:** 1. **Prepare the Gift:** * Remove any price tags or stickers. * If necessary, flatten the gift box to ensure a smooth surface for wrapping. 2. **Measure and Cut:** * Lay out your wrapping paper and place the gift centered on top. * Measure and cut the paper generously wider than the gift on all sides, leaving enough extra length for folding and sealing. 3. **Start Wrapping:** * Fold the top flap of the paper down, aligning the edges carefully. * Bring the left side of the paper over the right side, aligning the edges smoothly. Use tape to secure the join. Repeat on the other side, creating a neat rectangle. 4. **Secure the Sides:** * Fold the bottom edge of the paper up and secure it with tape. Repeat this process for the other side, creating a neat, closed edge. 5. **Fold and Tuck:** * Fold the top flap down again, creating a clean and smooth edge. Tuck any excess paper neatly underneath. Use tape to secure the folded edge. 6. **Finishing Touches:** * Gently smooth out any wrinkles or creases in the wrapping paper. * Cut a length of ribbon and tie a bow or secure it with tape for a decorative finish. * **Add a Gift Tag:** Write a personalized message and attach it to the gift. **Tips for Extra Neatness:** * **Use a ruler:** This helps achieve straight lines when folding and cutting the paper. * **Practice:** Wrapping presents neatly takes practice! Don't get discouraged if your first few attempts aren't perfect. * **Dampen the paper:** Lightly misting the paper with water can help it conform more easily to curves. * **Use double-sided tape:** This can create a more professional finish and prevent the tape from showing through the paper.

Response 2:

Score: 92 / 100

Here's a step-by-step guide to wrapping presents like a pro: **Preparation is Key** 1. **Gather Your Supplies:** * **Wrapping paper:** Choose paper that complements the gift and is the right size. * **Scissors:** Sharp scissors make clean cuts. * **Tape:** Masking tape for secure adhesion, clear tape for a more decorative look (optional). * **Ruler:** Helpful for straight lines and measuring. * **Box (optional):** For larger gifts, a sturdy box helps prevent wrinkles. * **Gift tag:** Don't forget the personal touch! * **Decoration (optional):** Ribbon, bows, twine, ornaments, etc. for extra flair. 2. **Prepare the Gift:** * **Neatly Fluff:** If the gift has stuffing, smooth it out to avoid bunching. * **Box it Up (optional):** If using a box, wrap the box itself with paper (double-wrap for extra protection). **Wrapping Techniques** * **The Basic Wrap:** 1. **Lay It Out:** Place the gift on a flat surface with some room to work. 2. **Paper Measurement:** * **Standard Size:** Paper should extend about 6 inches (15 cm) beyond both sides of the gift. * **Wider Gifts:** Add more paper on all sides. 3. **Center & Smooth:** Position the paper, aligning one short edge with the top/bottom of the gift. Smooth down firmly, removing wrinkles. 4. **Fold & Tuck:** Fold the top flap down, tucking the excess paper under the gift. Repeat with the sides, creasing well to secure. 5. **Seal It Up:** Apply tape along the entire edge of each flap, pressing firmly for a strong bond. 6. **Finishing Touch:** Fold the bottom edge up and tape neatly. * **Creative Extras:** * **Diagonal Wrap:** For a modern look, fold the paper diagonally to create a triangle at the corners, then tuck and tape. * **Double-Wrap:** For extra protection or a luxe feel, wrap the gift in two layers of paper, overlapping the edges. * **Slim Gift Hacks:** Use a long strip of paper, wrapping it around the gift like a scarf, then taping at the ends. **Finishing Touches** 1. **Cut the Tape:** Use a straight, clean cut to avoid jagged edges. 2. **Add a Ribbon (optional):** Tie a bow, wrap loosely around, or create a knot. 3. **Gift Tag Time:** Attach your tag securely with tape or a glue stick. 4. **Decoration Fun:** Get creative! Add sprigs of greenery, ornaments, or patterned ribbon for personality. **Tips for Perfection:** * **Practice Makes Perfect:** Wrap a few scraps of paper first to get the hang of it. * **Sharp Scissors:** They make clean cuts, reducing paper puckering. * **Don't Be Afraid to Experiment:** Find what wrapping styles you like best! * **Think About the Recipient:** A simple wrap is perfect for kids, while elegant paper and ribbon suit adults. * **Have Fun!** Wrapping should be enjoyable, not stressful. Let me know if you want help with a specific wrapping style or have a particular gift in mind! 🎁

Figure 11: Qualitative comparison of generated responses. Response 1 (ConsensusRM-Reverse) suffers from a dense, unstructured layout, resulting in a lower score (83/100). In contrast, Response 2 (ConsensusRM) exhibits superior organization with clear headings, achieving a higher score (92/100) as rated by GPT-5.2.