Shallow Preference Signals: Large Language Model Aligns Even Better with Truncated Data?

Xuan Qi*[2], Jiahao Qiu*[1], Xinzhe Juan[3], Yue Wu†[1], Mengdi Wang†[1],
[1]AI Lab, Princeton University, [2]IIIS, Tsinghua University,
[3]Department of Computer Science & Engineering, University of Michigan,

Abstract

Aligning large language models (LLMs) with human preferences remains a key challenge in AI. Preference-based optimization methods, such as Reinforcement Learning with Human Feedback (RLHF) and Direct Preference Optimization (DPO), rely on human-annotated datasets to improve alignment. In this work, we identify a crucial property of the existing learning method: the distinguishing signal obtained in preferred responses is often concentrated in the early tokens. We refer to this as *shallow preference signals*.

To explore this property, we systematically truncate preference datasets at various points and train both reward models and DPO models on the truncated data. **Surprisingly**, models trained on truncated datasets, retaining only the first half or fewer tokens, achieve comparable or even superior performance to those trained on full datasets. For example, a reward model trained on the Skywork-Reward-Preference-80K-v0.2 dataset outperforms the full dataset when trained on a 40% truncated dataset. This pattern is consistent across multiple datasets, suggesting the widespread presence of *shallow preference signals*.

We further investigate the distribution of the reward signal through decoding strategies. We consider two simple decoding strategies motivated by the shallow reward signal observation, namely Length Control Decoding and KL Threshold Control Decoding, which leverage shallow preference signals to optimize the trade-off between alignment and computational efficiency. The performance is even better, which again validates our hypothesis.

The phenomenon of *shallow preference signals* highlights potential issues in LLM alignment: existing alignment methods often focus on aligning **only the initial tokens** of responses, rather than considering the full response. This could lead to discrepancies with real-world human preferences, resulting in suboptimal alignment performance.

1 Introduction

Aligning large language models (LLMs) with human preferences is a core challenge in artificial intelligence (AI) research (Wang et al., 2023a). Preference datasets (Liu et al., 2024a; Cui et al., 2023; Askell et al., 2021; Bai et al., 2022) have played a critical role in addressing this challenge by capturing human judgments of model outputs. These datasets enable the identification and prioritization of responses that are more aligned with human expectations. Preference-based optimization techniques, such as Reinforcement Learning with Human Feedback (RLHF) (Ouyang et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2023), rely on these datasets to refine the decision-making process of models.

Despite the promise of these methods, there are several challenges associated with them. Recent work (Zhang et al., 2024; Park et al., 2024a,b; Ethayarajh et al., 2024) has highlighted that reward models trained using RLHF may suffer from reward hacking. Factors such as response format, length, and even the inclusion of emojis can influence quality judgments, resulting in potential inaccuracies. In this paper, we introduce a previously underexplored aspect of preference data. Specifically, we observe that the signal indicating the superiority of the chosen response over the rejected one is not uniformly distributed across the entire response. In many cases, the relative quality of responses can be determined from only the early portion of the response—or even just a few tokens—rather than requiring an evaluation of the entire response. We refer to this phenomenon as **shallow preference signals**. This observation suggests that preference-based optimization methods may not need to rely

* Equal contribution.
† Correspondence to: `frankwupku@gmail.com`, `mengdiw@princeton.edu`.

Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM² 2025), pages 532–548
July 31 - August 1, 2025

on the full response to effectively capture the distinguishing features of higher-quality responses.

We hypothesize that focusing on the early portion of the response allows models to capture the most salient preference signals, resulting in more efficient training and potentially improved alignment performance. To test this hypothesis, we introduce a methodology where preference data is truncated at various positions, and models are trained on these truncated datasets. We analyze the distribution of preference signals in response pairs and conduct systematic experiments to validate the hypothesis that models trained on truncated preference data perform comparably to models trained on the full dataset. This is confirmed for both reward models and models fine-tuned with DPO. Our findings demonstrate that the distinguishing features between the chosen and rejected responses are concentrated in the early part of the response. In fact, models trained on truncated datasets—using only the first half or fewer tokens of each response—achieve similar, or even superior, performance compared to those trained on the full dataset. For instance, a reward model trained on the Skywork-Reward-Preference-80K-v0.2 (Liu et al., 2024a) dataset achieves an accuracy of only 75.85% on RewardBench (Lambert et al., 2024). However, when the dataset is truncated to 50% and 40%, the accuracy increases to 75.88% and 76.35%, respectively. Even with a truncation to 25%, the accuracy remains at 69.92%. Similarly, a reward model trained on the RLHFlow-pair-data-v2-80K-wsafetyRLHFlow-pair-data-v2-80K-wsafety[1] dataset achieves an accuracy of 65.96% on RewardBench. After truncating the dataset to 50% and 40%, the accuracy improves to 72.16% and 69.71%, respectively, with accuracy remaining at 62.44% for a 33% truncation.

Furthermore, our experiments suggest that the shallow preference signal phenomenon significantly impacts LLM content generation. Based on this observation, we find that simple strategies can perform well without needing complex decoding approaches. Recent work (Yang et al., 2024; Bergner et al., 2024; Hu et al., 2024b; Kavehzadeh et al., 2024) has proposed various decoding strategies, but our findings indicate that by focusing on the early portion of the response, we can achieve an optimal trade-off between reward and KL divergence. To test this, we explore two decoding strategies—Length Control Decoding and KL Threshold Control Decoding—to see if the early-token bias observed during training affects generation at inference time. Our results show that the differences between the DPO model trained on full preference data and the reference model are most noticeable in the early tokens of the generated response. As more of the response is generated, the difference decreases. This suggests that the reward signal in DPO training is concentrated in the early tokens, rather than being evenly distributed. (Lin et al., 2024) also explores token distribution differences between base LLMs and aligned models, though their method primarily focuses on in-context learning, avoiding parameter fine-tuning.

Meanwhile, the findings of this paper may shed light on existing problems in LLM alignment. Our experiments validates that current alignment methods often focus on aligning earlier tokens, rather than considering full sentences. The latter portions of answers generated by LLM tend to be generated through an auto-regressive mechanism, which does not exhibit significant quality variation through our decoding experiments. Through extensive experiments, we validate our hypothesis that focusing on the early portion of the response allows models to capture the most salient preference signals, resulting in more efficient training and potentially improved alignment performance. **However, alignment with truncated data is shallow alignment which only improves the performance on metrics but may keep further away from the real-world alignment with human values.** (Qi et al., 2024) proposes a related issue, but their work is confined to safety alignment and does not extend to the broader alignment challenges present in LLMs. Instead, our work validates the phenomenon more systemically and extensively.

In summary, the main contributions of our paper are as follows:

1. We introduce and systematically validate the phenomenon of **shallow preference signals**, demonstrating that the distinguishing features between high-quality and low-quality responses are often concentrated in the early portion of the response.

2. We show that training reward models and DPO models on truncated responses—using only the early portion—achieves performance

[1] https://huggingface.co/datasets/RLHFlow/pair_data_v2_80K_wsafety

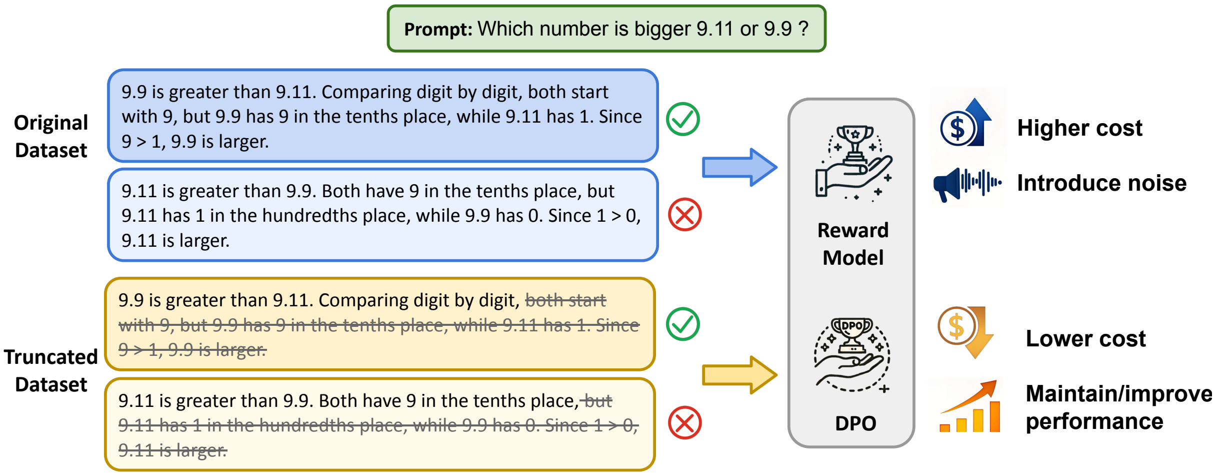


Figure 1: An example illustrating the phenomenon of shallow preference signals. It demonstrates how the relative quality of two responses can be determined from the early portion of the response, or even from the first sentence. Training with only the initial part allows the model to capture most of the preference signals while conserving resources.

comparable to or better than training on full responses. This finding holds across multiple datasets and supervision settings.

3. We provide a new perspective on the limitations of current alignment pipelines. Specifically, we suggest that current alignment methods face the limitation of shallow alignment, emphasizing that alignment should go beyond just aligning a few tokens and consider full sentences for more effective results.

2 Related Works

2.1 LLM Alignment with Human Preference

Aligning the outputs of large language models with human preferences is a crucial problem in the field of LLMs (Wang et al., 2023a). One of the most notable advancements in this area is Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022), which has led to the development of cutting-edge language models such as GPT-4o (Hurst et al., 2024), Gemini-2.0 (Anil et al., 2023), and Llama-3.1-70B-Instruct (Dubey et al., 2024). The traditional RLHF approach involves training a reward model to score the outputs of the language model, followed by fine-tuning using deep reinforcement learning algorithms like Proximal Policy Optimization (PPO) (Bai et al., 2022). However, PPO faces challenges in alignment tasks due to its complexity, instability, and inefficiency (Choshen et al., 2020; Engstrom et al., 2020). Several works have sought to improve the RLHF paradigm from various angles in order to better align LLMs with human preferences (Zhao et al., 2023; Azar et al., 2024; Tang et al., 2024). Among these, Direct Preference Optimization (DPO) (Rafailov et al., 2023) has gained significant attention, as it directly optimizes a policy using chosen and rejected pairs.

2.2 Reward Model

The reward model plays a critical role in RLHF (Christiano et al., 2017; Ouyang et al., 2022). Traditional reward models are often assumed to follow a Bradley-Terry model (Bradley and Terry, 1952a), which provides a score for an entire output to indicate its preference (Wang et al., 2023b; Christiano et al., 2017; Ouyang et al., 2022). However, the Bradley-Terry model has limitations, particularly its inability to handle complex or intransitive preferences (Munos et al., 2024; Swamy et al., 2024; Ye et al., 2024). Some works have addressed this issue by discarding the Bradley-Terry assumption and instead modeling the probability that one response is preferred over another (Jiang et al., 2023; Liu et al., 2024b; Dong et al., 2024a). Additionally, other approaches have explored the construction of multi-objective reward models to capture human preferences more comprehensively (Touvron et al., 2023; Wang et al., 2024b,a). Furthermore, some studies have proposed process reward models (Luo et al., 2023; Lightman et al., 2024; Li and Li, 2024) or stepwise reward models (Havrilla et al., 2024), which have shown promising results, especially in reasoning tasks.

2.3 Reward Hacking

Reward hacking refers to the situation in which an agent or model optimizes a proxy reward that deviates from the true objective, leading to suboptimal or even undesirable behavior (Skalse et al., 2022). This phenomenon has been widely studied across various environments such as grid-worlds, Atari games, and text generation tasks (Arjona-Medina et al., 2019; Pan et al., 2022; Xu et al., 2022). Prior research has focused on categorizing different forms of reward hacking and developing mitigation strategies, such as regularizing policy optimization (Laidlaw et al., 2024), imposing a KL divergence penalty (Miao et al., 2024), and applying model merging techniques to either the policy or reward model (Zhang et al., 2024). Despite these efforts, existing approaches have notable limitations. In response, recent studies have introduced new definitions and strategies for mitigating reward hacking, including the concept of "hackability" (Skalse et al., 2022) and the use of information-theoretic reward modeling (Miao et al., 2024). Furthermore, the application of reward hacking techniques to language models has been explored, particularly in improving the sample efficiency of preference learning (Zhu et al., 2024). In contrast to these prior approaches, our work mitigates a subset of reward hacking by truncating the model's responses and better aligning them with human preferences. This truncation process effectively reduces noise in the dataset, leading to improved accuracy. By removing certain noise components, our method can be seen as a novel approach to addressing reward hacking within the context of language models.

3 Methodology

In this section, we introduce the methodology used to investigate the structure and front-loaded nature of reward signals in large language models (LLMs) trained with preference data.

3.1 Formulation of Reward Signal Location

Consider a preference dataset containing pairs of responses, where one response is the *chosen response* and the other is the *rejected response*. The reward signal is defined as the inherent quality difference between these two responses. Let $r_{\text{cho}}(i)$ denote the *chosen response* for a given instance i, and $r_{\text{rej}}(i)$ denote the *rejected response*. The objective is to model the reward signal $R(i)$, which indicates the degree of preference for $r_{\text{cho}}(i)$ over $r_{\text{rej}}(i)$.

We hypothesize that the reward signal is concentrated in the early part of the response. To formalize this, let $r_{\text{cho}}(i) = [y_1, y_2, \ldots, y_T]$ and $r_{\text{rej}}(i) = [z_1, z_2, \ldots, z_T]$ represent the token sequences for the chosen and rejected responses, respectively, where T is the total number of tokens in each response. We define the reward signal at each token position t as the difference in the model's log-probability for the chosen and rejected responses at that position:

$$R_t(i) = \log p(y_t \mid x, y_{1:t-1}) - \log p(z_t \mid x, z_{1:t-1}),$$

where x represents the input context, and $\log p(y_t \mid x, y_{1:t-1})$ is the log-probability of the token y_t in the chosen response at position t, conditioned on the context x and the preceding tokens $y_{1:t-1}$. Similarly, $\log p(z_t \mid x, z_{1:t-1})$ is the log-probability of the token z_t in the rejected response at the same position.

We argue that the total reward signal $R(i)$ can be approximated as the cumulative sum of the reward signals up to a truncation point t_k:

$$R(i) = \sum_{t=1}^{t_k} R_t(i) = \log p(y_{1:t_k} \mid x) - \log p(z_{1:t_k} \mid x),$$

where t_k represents the truncation point, beyond which the reward signal becomes less informative or introduces noise. This leads to the hypothesis that truncated responses up to position t_k preserve most of the reward signal, enabling the training of effective reward models and DPO models without requiring the full response.

To further validate our hypothesis, we investigate the effects of truncating the responses in preference datasets on training the reward model and DPO, where a formal statement can be found in Appendix B.

3.2 Mixing Strategy and Decoding Policies

To further investigate the impact of early-token preference signals during decoding, we utilize a mixing strategy and two decoding policies. The mixing strategy combines the DPO policy with the corresponding reference model policy to enhance the reward-KL divergence tradeoff.

3.2.1 Mixing Strategy

The mixing strategy involves combining the probability distributions from the DPO model π_{DPO}

and the reference model π_{ref} in a weighted manner. Specifically, we define a mixing policy π_{mix} as:

$$\pi_{\text{mix}} = \text{softmax}\left(a \cdot \log \frac{\pi_{\text{DPO}}}{\pi_{\text{ref}}} + \log \pi_{\text{ref}}\right)$$

where a is a mixing coefficient controlling the trade-off between the DPO and reference model. This strategy allows for fine-tuning the balance between the reward signal captured by the DPO policy and the stability provided by the reference model.

3.2.2 Decoding Strategies

We explore two decoding strategies that prioritize the early part of the response or manage the KL divergence between the DPO and reference models.

Length Control Decoding: In this strategy, the first t tokens are generated by sampling from the DPO policy, while the remaining tokens are generated by sampling from the reference model. The goal is to focus on the part of the response where the reward signal is concentrated. The strategy is parameterized by the truncation length t, which controls the point at which the decoding switches between the two models.

$$y_k = \begin{cases} \text{sample from } \pi_{\text{DPO}} & \text{if } k \leq t \\ \text{sample from } \pi_{\text{ref}} & \text{if } k > t \end{cases}$$

KL Threshold Control Decoding: In this strategy, we compute the KL divergence between the DPO model and the reference model at each token generation step. If the KL divergence exceeds a predefined threshold b, we sample from the DPO policy; otherwise, we sample from the reference model. This dynamic approach allows the model to maintain flexibility in adjusting to the relative importance of reward signal versus stability during the response generation process.

$$y_t = \begin{cases} \text{sample from } \pi_{\text{DPO}} & \text{if } \text{KL}(\pi_{\text{DPO}} \parallel \pi_{\text{ref}}) > b \\ \text{sample from } \pi_{\text{ref}} & \text{if } \text{KL}(\pi_{\text{DPO}} \parallel \pi_{\text{ref}}) \leq b \end{cases}$$

where $y_t^{(i)}$ denotes the i-th sampled token from the DPO model at the t-th position.

The KL divergence $\text{KL}(\pi_{\text{DPO}} \parallel \pi_{\text{ref}})$ is computed at each token position as:

$$\text{KL}(\pi_{\text{DPO}} \parallel \pi_{\text{ref}}) = \mathbb{E}_{y_t \sim \pi_{\text{DPO}}}\left[\log \frac{\pi_{\text{DPO}}(y_t|x, y_{<t})}{\pi_{\text{ref}}(y_t|x, y_{<t})}\right]$$

This expectation is estimated using Monte Carlo sampling. Specifically, we sample $K = 1,000$ tokens from the DPO model at each token position, and the KL divergence is computed as:

$$\hat{\text{KL}}(\pi_{\text{DPO}} \parallel \pi_{\text{ref}}) = \frac{1}{K}\sum_{i=1}^{K} \log \frac{\pi_{\text{DPO}}(y_t^{(i)}|x, y_{<t})}{\pi_{\text{ref}}(y_t^{(i)}|x, y_{<t})}$$

Both of these strategies are used to examine how early-token reward signals influence inference-time behavior, while maintaining acceptable KL divergence during decoding.

4 Experiment: Truncation Effects on Reward Models and DPO

4.1 Experiment Setting

In this experiment, we investigate the effect of truncating response sequences at different positions within preference datasets Skywork-Reward-Preference-80K-v0.2 (Liu et al., 2024a), ultrafeedback-binarized (Cui et al., 2023), and RLHFlow-pair-data-v2-80K-wsafety[2], which are commonly used in the context of large language models. Specifically, we apply truncation to the response sections (including both chosen and rejected responses) at varying positions. The truncation process retains only the initial portion of the response tokens, while the remaining tokens are discarded, resulting in the creation of multiple truncated datasets. We then train reward models and use Direct Preference Optimization (DPO) to fine-tune models on these truncated datasets and compare their performance with models trained on the original, untruncated datasets. We also investigate the use of DPO implicit reward (Rafailov et al., 2023) to assess the quality of two responses on datasets with different truncation ratios, and compare the accuracy of this evaluation with the actual quality judgments.

We utilize Google's gemma-2b-it[3] model as the base for training the reward model, following the methodology outlined in RLHFlow (Dong et al., 2024b) to train a standard Bradley-Terry reward model (Bradley and Terry, 1952b). For the DPO training, we use the Llama-3.1-8B-Instruct (Patterson et al., 2022) as the base model, following the DPO methodology outlined in OpenRLHF (Hu et al., 2024a) to fine-tune the model. In the experiment using DPO implicit reward to assess accuracy, we use the LLaMA3-iterative-DPO-final model (Xiong et al., 2024; Dong et al., 2024b) as the DPO policy model and its supervised fine-tuning (SFT) checkpoint, LLaMA3-SFT, trained from Llama-3-8B, as the reference policy model.

[2]https://huggingface.co/datasets/RLHFlow/pair_data_v2_80K_wsafety

[3]https://huggingface.co/google/gemma-2b-it

4.1.1 Metrics

The performance of the models is evaluated using two metrics:

Test Accuracy. This metric measures the proportion of instances where the reward model assigns a higher score to the chosen response compared to the rejected response.

GPT4o Win Rate. This metric is computed using the AlpacaEval 2.0 (Li et al., 2023) standard test set and the default baseline model with GPT4o acting as the judge.

4.2 Results

4.2.1 Evaluation of Reward Models on RewardBench

We evaluate the performance of the trained reward models on the core RewardBench evaluation set. For each dataset, we train the reward models on the training set using truncated versions of the responses with truncation ratios of 50%, 40%, 33% and 25%. The results are presented in Table 1.

Truncating the response in the preference data to 50% or 40% of tokens had minimal impact on the performance of the trained reward model across all three datasets. In fact, for certain metrics and datasets, models trained on truncated data outperformed those trained on full responses. However, truncating the response to 33% or 25% of its original length leads to a slight reduction in performance. Despite this, the performance drop remains small, and the models continue to exhibit the majority of the performance seen with the original, untruncated datasets.

4.2.2 Evaluation of Reward Models on Each Task of UltraFeedback

We train reward models on the ultrafeedback-binarized dataset, separately for each task: Helpfulness, Honesty, Instruction Following, and Truthfulness. For each task, we train the reward models on the training set using truncated versions of the responses with truncation ratios of 50%, 40%, 30%, 20% and 10%. Results are shown in Table 2.

The results show that truncating the responses to 50% or 40% of their original length had a negligible effect on test accuracy for each task. In some tasks, models trained on truncated data even perform better than those trained on full responses. However, when the responses are truncated to shorter lengths (e.g., 30%, 20%, or 10%), a slight decrease in test accuracy is observed. Nonetheless, the models retain a substantial portion of their original performance, indicating that truncation did not result in a significant loss of accuracy.

4.2.3 Evaluation of DPO-trained Models on AlpacaEval 2.0

In addition to training reward models, we investigate the effect of response truncation in the preference dataset by Direct Preference Optimization (DPO). For this experiment, we use the Skywork-Reward-Preference-80K-v0.2 dataset (Liu et al., 2024a). The dataset responses are truncated at various ratios of 50%, 40%, 33% and 25%. Results are shown in Table 3.

The results indicate that truncating the responses in the preference data had a minimal effect on the performance of models trained with DPO. While the impact increased with the truncation ratio, truncating the response to 50% or 40% of its original length does not significantly degrade the performance of the DPO-trained models. This suggests that, in the context of DPO training, the majority of the signals used to evaluate response quality are concentrated in the earlier segments of the response.

4.2.4 Implicit Reward Accuracy on Truncated Responses

In this experiment, we truncate the responses in the Skywork-Reward-Preference-80K-v0.2 (Liu et al., 2024a) dataset at various proportions and compute the DPO implicit reward for each response pair. We then compare the preferences derived from the implicit rewards with the actual human-annotated preferences to assess the consistency. The results are presented in Figure 2.

The results indicate that as the length of the response considered increases, the preferences derived from the DPO implicit reward align more closely with human-annotated preferences. Interestingly, even when only the initial portion of the response is considered, the preferences derived from the DPO implicit reward show a high degree of consistency with human preferences. This suggests that, in preference datasets, evaluating only the early tokens of a response is sufficient to accurately assess the relative quality of two responses, without the need to examine the entire response.

Dataset	Dimension	Original Dataset	50%	40%	33%	25%
Skywork-Preference	Chat	0.8073	0.7318	0.7039	0.5866	0.5978
	Chat-Hard	0.7039	0.7105	0.6974	0.6776	0.6732
	Safety	0.8216	0.8068	0.7946	0.8162	0.8030
	Reasoning	0.7043	0.7769	0.8101	0.7064	0.7450
	Total	0.7585	0.7588	0.7635	0.7000	0.6992
UltraFeedback	Chat	0.7946	0.8098	0.8073	0.7844	0.7644
	Chat-Hard	0.6029	0.6425	0.6342	0.5983	0.5946
	Safety	0.7416	0.7632	0.7848	0.7384	0.6756
	Reasoning	0.7056	0.6904	0.6682	0.6886	0.5646
	Total	0.7391	0.7327	0.7194	0.7018	0.6355
RLHFlow-Preference	Chat	0.9553	0.9302	0.9287	0.8574	0.8291
	Chat-Hard	0.4517	0.4561	0.4506	0.4323	0.4127
	Safety	0.6730	0.6621	0.6438	0.5985	0.6081
	Reasoning	0.5984	0.8374	0.7894	0.6247	0.5723
	Total	0.6596	0.7216	0.6971	0.6244	0.5562

Table 1: Performance of reward models trained on different truncation ratios for various datasets. The table presents the evaluation scores across multiple dimensions from the RewardBench core set: **Chat**, **Chat-Hard**, **Safety** and **Reasoning**. **Total** is the final score on the RewardBench core set. **Skywork-Preference** refers to Skywork-Reward-Preference-80K-v0.2 dataset, **UltraFeedback** refers to ultrafeedback-binarized dataset, **RLHFlow-Preference** refers to RLHFlow-pair-data-v2-80K-wsafety dataset. **Original Dataset** refers to the model trained on the full dataset without truncation; **50%**, **40%**, **33%**, and **25%** refer to truncated datasets with corresponding ratios. The highest score in each row is highlighted with darker blue, and the second-highest score with lighter blue.

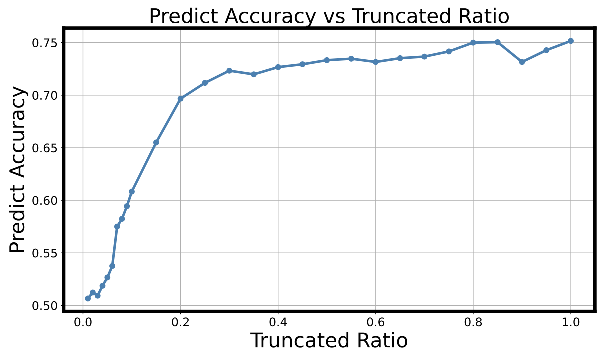


Figure 2: The x-axis represents the response truncation ratio, while the y-axis shows the accuracy of DPO implicit reward in predicting the relative quality of responses based on truncated datasets.

5 Experiment: KL Divergence and Reward-KL Tradeoff for Evaluating Response Quality

This section presents a set of experiments that examine the relationship between the Kullback-Leibler (KL) divergence between the DPO model and the reference model, and the reward-KL tradeoff during response generation. These experiments aim to validate the hypothesis that the reward signal in preference datasets is primarily concentrated in the early part of the response, highlighting the phenomenon of shallow preference signals.

5.1 Experiment Setup

To investigate this hypothesis, we perform two key experiments. In the first experiment, we compute the KL divergence between the DPO model and the reference model at each token generation step. This experiment allows us to observe how the KL divergence evolves as the response is generated and whether the early tokens exhibit a higher divergence compared to later ones. In the second experiment, we explore the reward-KL tradeoff during generation. Based on our observation of shallow preference signals, we adjust the sampling strategy according to the behavior of the DPO and reference models to further confirm that the reward signal is concentrated in the early part of the response. We use a simple baseline decoding strategy, described in subsubsection 3.2.1, and test different decoding strategies to explore how well the early preference signal can be captured.

For both experiments, we use the LLaMA3-iterative-DPO-final model (Xiong et al., 2024; Dong et al., 2024b) as the DPO policy model and its supervised fine-tuning (SFT) checkpoint, LLaMA3-SFT, trained from Llama-3-8B, as the reference policy model. The corresponding reward is measured using the reward model FsfairX-LLaMA3-RM-v0.1 (Dong et al., 2024b). We ran-

Task	Original Dataset	50%	40%	30%	20%	10%
Helpfulness	0.89	0.90	0.90	0.87	0.82	0.73
Honesty	0.87	0.88	0.87	0.84	0.79	0.76
Instruction Following	0.91	0.91	0.86	0.87	0.74	0.69
Truthfulness	0.85	0.84	0.84	0.83	0.81	0.64
Average	0.88	0.8825	0.87	0.855	0.795	0.705

Table 2: UltraFeedback test accuracy across different tasks with various truncation ratios. The table presents the test accuracy for each task in the UltraFeedback dataset, with different truncation ratios: **Original Dataset** refers to the model evaluated on the full, unmodified UltraFeedback dataset; **50%**, **40%**, **30%**, **20%**, and **10%** refer to models evaluated using truncated versions of the dataset. The tasks listed include: **Helpfulness**, **Honesty**, **Instruction Following**, and **Truthfulness**. **Average** represents the mean accuracy across all tasks. The highest score in each row is highlighted with darker blue, and the second-highest score with lighter blue.

Metric	Llama3.1 8B	Original Dataset	50%	40%	33%	25%
LCWR	21.45	24.90	25.19	24.85	23.51	21.13
WR	22.37	23.92	24.15	23.57	23.43	20.96

Table 3: Performance of DPO models with different truncation ratios. The table presents the evaluation metrics for both the original model and the DPO models trained on truncated datasets: **Llama3.1 8B** refers to the original Llama-3.1-8B-Instruct model; **Original Dataset** refers to the Llama-3.1-8B-Instruct model fine-tuned using the full Skywork-Reward-Preference-80K-v0.2 dataset with the DPO algorithm; **50%**, **40%**, **33%**, and **25%** refer to models fine-tuned using truncated versions of the dataset. **LCWR** refers to Length-controlled Win Rate and **WR** refers to Win Rate. The highest score in each row is highlighted with darker blue, and the second-highest score with lighter blue.

domly selected 1000 instructions from the training sets of Alpaca (Taori et al., 2023) and UltraFeedback (Cui et al., 2023) to form the instruction sets for these two experiments. The KL divergence between the two policies at each token is computed as described in subsubsection 3.2.2, and the KL divergence between the two policies for the whole response generation is accumulated across all token generation steps. The final KL divergence is computed as:

$$\hat{\mathrm{KL}}(\pi_{\text{mix}} \parallel \pi_{\text{ref}}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\log\frac{\pi_{\text{mix}}(y_t^{(i)}|x_i, y_{<t})}{\pi_{\text{ref}}(y_t^{(i)}|x_i, y_{<t})}$$

where N represents the size of the instruction set, T denotes the total number of tokens in the response, x_i is the instruction, $y_t^{(i)}$ refers to the generated token at position t, and $y_{<t}$ refers to the tokens generated prior to token t.

5.2 Results

5.2.1 KL Divergence Analysis Across Token Positions

In the first experiment, we analyze the KL divergence between the DPO model and the reference model at each token generation step. The KL divergence is computed for each token y_t by comparing the conditional probability distributions of the DPO model $\pi_{\text{DPO}}(y_t|x, y_{<t})$ and the reference model $\pi_{\text{ref}}(y_t|x, y_{<t})$, where x is the instruction, and $y_{<t}$ represents previously generated tokens. As shown in Figure 3, the KL divergence is high in the early tokens, indicating significant differences between the DPO and reference models. However, the divergence diminishes significantly as token generation progresses, suggesting that the primary divergence occurs in the initial phase of response generation.

This observation supports the hypothesis that the reward signal in preference datasets is mostly concentrated in the first part of the response, with minimal divergence in the later tokens, where the DPO model relies on the tokens generated earlier.

5.2.2 Reward-KL Tradeoff for Length Control and KL Threshold Control Decoding

The second experiment explores the reward-KL tradeoff during response generation, based on the observation of shallow preference signals. We focus on two simple decoding strategies: Length Control Decoding and KL Threshold Control Decoding, which are based on the idea that the reward signal is concentrated in the early portion of the response.

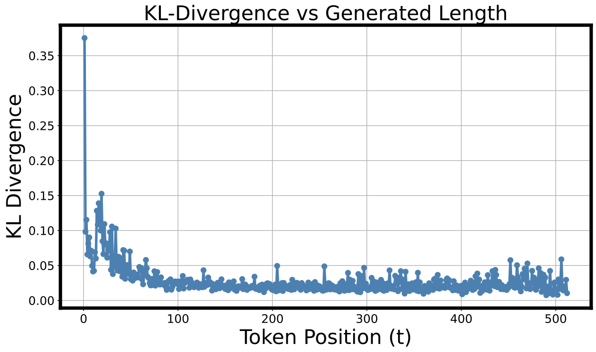


Figure 3: KL Divergence between the DPO model and the reference model at each token position. The plot shows that the divergence is higher for early tokens and decreases as generation progresses.

Length Control Decoding In Length Control Decoding, we sample from the DPO policy for the first t tokens and from the reference policy for the remaining tokens. We evaluate this strategy for various values of t and compute the average reward and KL divergence for each configuration.

KL Threshold Control Decoding In KL Threshold Control Decoding, we compute the KL divergence $\mathrm{KL}(\pi_{\mathrm{DPO}} \,\|\, \pi_{\mathrm{ref}})$ at each token position. If the divergence exceeds a threshold b, we sample from the DPO policy; otherwise, we sample from the reference policy. We test several values of b and record the average reward and KL divergence.

The results of both strategies, shown in Figure 4, demonstrate that simple strategies, based on the observed concentration of reward signals in the early tokens, improve the reward-KL tradeoff compared to the baseline. These findings confirm that adjusting the decoding strategy in a simple manner—by focusing on the early tokens—can lead to better alignment between reward and KL divergence, further supporting the idea that the reward signal is concentrated in the early part of the response.

6 Conclusion

We introduce shallow preference signals, where key distinguishing features between preferred and non-preferred responses are concentrated in early response tokens. Our experiments show that models trained on truncated data—retaining 40% to 50% of tokens—perform similarly or better in reward modeling and Direct Preference Optimization (DPO) than those trained on full-length data. Additionally, we highlight the limitation of current methods that focus mainly on initial tokens, suggesting the need for strategies that consider entire

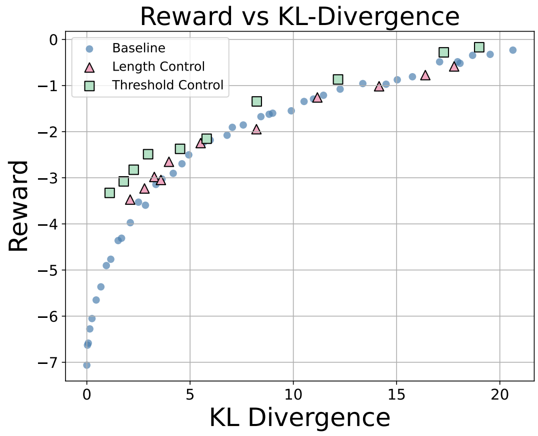


Figure 4: Reward and corresponding KL Divergence for the baseline and two different control strategies. The blue dots represent data from the baseline, while the red triangles and green squares represent the Length Control and KL Threshold Control strategies, respectively.

responses for more accurate alignment with human preferences.

References

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. 2019. RUDDER: return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13544–13555.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Benjamin Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. A general language assistant as a laboratory for alignment. *CoRR*, abs/2112.00861.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pages 4447–4455. PMLR.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862.

Benjamin Bergner, Andrii Skliar, Amelie Royer, Tijmen Blankevoort, Yuki M. Asano, and Babak Ehteshami Bejnordi. 2024. Think big, generate quick: Llm-to-slm for fast autoregressive decoding. *CoRR*, abs/2402.16844.

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

Ralph Allan Bradley and Milton E. Terry. 1952b. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2020. On the weaknesses of reinforcement learning for neural machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307.

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

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024a. RLHF workflow: From reward modeling to online RLHF. *CoRR*, abs/2405.07863.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024b. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, 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 Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, 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, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. 2020. Implementation matters in deep policy gradients: A case study on PPO and TRPO. *CoRR*, abs/2005.12729.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: model alignment as prospect theoretic optimization. *CoRR*, abs/2402.01306.

Alexander Havrilla, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Raileanu. 2024. Glore: When, where, and how to improve LLM reasoning via global and local refinements. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Weixun Wang, Dehao Zhang, and Yu Cao. 2024a. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*.

Yuxuan Hu, Ke Wang, Xiaokang Zhang, Fanjin Zhang, Cuiping Li, Hong Chen, and Jing Zhang. 2024b. SAM decoding: Speculative decoding via suffix automaton. *CoRR*, abs/2411.10666.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. 2024. Gpt-4o system card. *CoRR*, abs/2410.21276.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14165–14178. Association for Computational Linguistics.

Parsa Kavehzadeh, Mohammadreza Pourreza, Mojtaba Valipour, Tinashu Zhu, Haoli Bai, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. 2024. S2D: sorted speculative decoding for more efficient deployment of nested large language models. *CoRR*, abs/2407.01955.

Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. 2024. Correlated proxies: A new definition and improved mitigation for reward hacking. *Preprint*, arXiv:2403.03185.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. *CoRR*, abs/2403.13787.

Wendi Li and Yixuan Li. 2024. Process reward model with q-value rankings. *CoRR*, abs/2410.11287.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Raghavi Chandu, Chandra Bhagavatula, and Yejin Choi. 2024. The unlocking spell on base llms: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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. *CoRR*, abs/2410.18451.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. 2024b. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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. *CoRR*, abs/2308.09583.

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. 2024. Inform: Mitigating reward hacking in RLHF via information-theoretic reward modeling. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Côme Fiegel, Andrea Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J. Mankowitz, Doina Precup, and Bilal Piot. 2024. Nash learning from human feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. 2024a. Offsetbias: Leveraging debiased data for tuning evaluators. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 1043–1067. Association for Computational Linguistics.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024b. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 4998–5017. Association for Computational Linguistics.

David A. Patterson, Joseph Gonzalez, Urs Hölzle, Quoc V. Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. 2022. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 55(7):18–28.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2024. Safety alignment should be made more than just a few tokens deep. *CoRR*, abs/2406.05946.

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 *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward hacking. *CoRR*, abs/2209.13085.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Steven Wu, and Alekh Agarwal. 2024. A minimaximalist approach to reinforcement learning from human feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. 2024. Generalized preference optimization: A unified approach to offline alignment. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca:

An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

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, Miami, Florida, USA, November 12-16, 2024*, pages 10582–10592. Association for Computational Linguistics.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023a. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. 2024b. 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), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 3371–3384. Association for Computational Linguistics.

Ziqi Wang, Le Hou, Tianjian Lu, Yuexin Wu, Yunxuan Li, Hongkun Yu, and Heng Ji. 2023b. Enable language models to implicitly learn self-improvement from data. *CoRR*, abs/2310.00898.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. 2024. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. *Preprint*, arXiv:2312.11456.

Yingchen Xu, Jack Parker-Holder, Aldo Pacchiano, Philip J. Ball, Oleh Rybkin, Stephen Roberts, Tim Rocktäschel, and Edward Grefenstette. 2022. Learning general world models in a handful of reward-free deployments. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Seongjun Yang, Gibbeum Lee, Jaewoong Cho, Dimitris Papailiopoulos, and Kangwook Lee. 2024. Predictive pipelined decoding: A compute-latency trade-off for exact LLM decoding. *Trans. Mach. Learn. Res.*, 2024.

Chenlu Ye, Wei Xiong, Yuheng Zhang, Nan Jiang, and Tong Zhang. 2024. A theoretical analysis of nash learning from human feedback under general kl-regularized preference. *CoRR*, abs/2402.07314.

Xuanchang Zhang, Wei Xiong, Lichang Chen, Tianyi Zhou, Heng Huang, and Tong Zhang. 2024. From lists to emojis: How format bias affects model alignment. *CoRR*, abs/2409.11704.

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

Yuchen Zhu, Daniel Augusto de Souza, Zhengyan Shi, Mengyue Yang, Pasquale Minervini, Alexander D'Amour, and Matt J. Kusner. 2024. When can proxies improve the sample complexity of preference learning? *CoRR*, abs/2412.16475.

A Preliminaries

A.1 Autoregressive Language Model And Token-Level Markov Decision Process

Autoregressive language models (ARLMs) are designed to generate token sequences $y_1, y_2, \ldots, y_T$ conditioned on the preceding tokens in a given context. Formally, for a provided input prompt x, the model generates the token sequence $y = (y_1, y_2, \ldots, y_T)$ by factorizing the joint distribution of the sequence using the chain rule of probability:

$$p(y|x) = \prod_{t=1}^{T} p(y_t|y_1, y_2, \ldots, y_{t-1}, x),$$

where $p(y_t|y_1, y_2, \ldots, y_{t-1}, x)$ represents the conditional probability of generating token y_t, given all previous tokens $y_1, y_2, \ldots, y_{t-1}$ and the input prompt x.

This process is typically framed as a token-level Markov Decision Process (MDP), where each state at time step t, denoted s_t, represents the sequence of tokens generated up to that point:

$$s_t = (x, y_1, y_2, \ldots, y_{t-1}),$$

and the action a_t corresponds to the generation of the next token y_t. The transitions between states are deterministic and are given by:

$$s_{t+1} = (x, y_1, y_2, \ldots, y_t),$$

as each subsequent state is determined solely by the previous state and the action of generating the next token.

This token-level MDP formulation is useful for various applications, such as in training RL-based models where the language model needs to learn to generate tokens that not only fit the linguistic context but also satisfy some predefined quality criteria. Moreover, recent advancements in reinforcement learning from human feedback (RLHF) have sought to fine-tune such models to align with human preferences, making this framework essential for ensuring that ARLMs produce high-quality, aligned outputs.

In the context of reinforcement learning (RL), the task is framed as a Max-Entropy RL problem, where the reward is a combination of a task-specific reward function and a regularization term. The objective is to maximize the expected sum of the rewards, along with the entropy of the policy to promote exploration:

$$\mathbb{E}_{x\sim X, y\sim\pi(\cdot|x)} \left[r(y|x) + \beta \log \pi_{\text{ref}}(y|x)\right] + \beta \mathbb{E}_{x\sim X}[H(\pi(\cdot|x))]$$

where $r(y|x)$ represents the reward for generating a sequence y given the input prompt x, $\pi_{\text{ref}}(y|x)$ is a reference policy that can be used to encourage alignment with desired behaviors, and $H(\pi(\cdot|x))$ is the entropy of the policy at time t, promoting exploration by discouraging deterministic behaviors.

At the token level, the RL objective can be rewritten as:

$$\mathbb{E}_{s_0\sim X, a_t\sim\pi(\cdot|s_t)} \left[\sum_{t=1}^{T} r'(s_t, a_t)\right] + \beta \mathbb{E}_{s_0\sim X}[H(\pi(\cdot|s_0))],$$

where $r'(s_t, a_t)$ is the token-level reward, defined as:

$$r'(s_t, a_t) = \begin{cases} \beta \log \pi_{\text{ref}}(a_t|s_t), & \text{if } s_{t+1} \text{ is not terminal,} \\ r(y|x) + \beta \log \pi_{\text{ref}}(a_t|s_t), & \text{otherwise.} \end{cases}$$

In this formulation, the reward function $r(y|x)$ typically measures how well the generated sequence aligns with the desired outcome, while the entropy term $\beta \log \pi_{\text{ref}}(a_t|s_t)$ encourages diversity in the generated tokens.

The objective in reinforcement learning is to find an optimal policy π^* that maximizes the expected cumulative reward. This is done by solving for the optimal Q-function $Q^*(s_t, a_t)$, which provides the expected future reward for taking action a_t from state s_t:

$$Q^*(s_t, a_t) = r'(s_t, a_t) + V^*(s_{t+1}),$$

where $V^*(s_t)$ is the optimal state-value function, representing the expected reward from state s_t. The optimal policy π^* satisfies the following equation:

$$\beta \log \frac{\pi^*(a_t|s_t)}{\pi_{\text{ref}}(a_t|s_t)} = Q^*(s_t, a_t) - V^*(s_t).$$

When $t < T$, the optimal policy maximizes the difference between the state-value function of the next state and the current state, encouraging the model to generate the sequence that leads to the highest cumulative reward.

A.2 RLHF with Reward Models

Reinforcement learning from human feedback (RLHF) is an approach where a reward model is used to guide the training of the language model. The reward model $r(y|x)$ evaluates the quality of a generated response y given a prompt x. The goal is to maximize the expected reward by adjusting the model's parameters using a policy optimization algorithm such as Proximal Policy Optimization (PPO)

Initially, (Christiano et al., 2017) proposed learning a reward model using the Bradley-Terry model to assign a score to each response. For a pair of responses y and y', the Bradley-Terry model defines the probability that y is preferred over y' as:

$$P(y \succ y'|x) = \frac{\exp(r(y;x))}{\exp(r(y;x)) + \exp(r(y';x))},$$

The reward function is learned by maximizing the log-likelihood of preference predictions.

For a triplet (x, y_w, y_l), where y_w is the winner and y_l is the loser, the Direct Preference Optimization (DPO) loss is derived as follows:

$$\ell_{\text{DPO}}(x, y_w, y_l; \theta; \pi_{\text{ref}}) :=$$
$$-\log\sigma\left(\beta\left[\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right]\right)$$

where $\sigma(\cdot)$ is the logistic function, $\sigma(z) = \frac{1}{1+\exp(-z)}$ and β is a hyperparameter that controls the importance of the preference signal in the optimization process. This DPO method provides a more efficient and stable solution compared to traditional methods that require separate reward modeling and policy optimization.

B Training Reward Models and DPO Models with Truncated Preference Data

In this work, we investigate the effects of truncating the responses in preference datasets at various positions. Let $r_{\text{cho}}(i)^{\text{trunc}}$ and $r_{\text{rej}}(i)^{\text{trunc}}$ denote the truncated chosen and rejected responses, respectively, where truncation is applied to retain only the first t_k tokens of each response:

$$r_{\text{cho}}(i)^{\text{trunc}} = [y_1, y_2, \dots, y_{t_k}],$$
$$r_{\text{rej}}(i)^{\text{trunc}} = [z_1, z_2, \dots, z_{t_k}]$$

We train reward models on these truncated preference datasets. The reward model aims to predict the relative quality of responses given the truncated input. Specifically, we model the reward using the following formula:

$$P(y \succ y' \mid x) = \frac{\exp(r(y;x))}{\exp(r(y;x)) + \exp(r(y';x))},$$

where $r(y;x)$ represents the reward function for response y given the context x, and $P(y \succ y' \mid x)$ is the probability that response y is preferred over y'. Although the reward model is trained on truncated responses, it is still able to assess the quality of full responses effectively by leveraging the reward function learned from the truncated portions.

Similarly, for Direct Preference Optimization (DPO), we fine-tune a base model on the truncated preference datas. The DPO objective seeks to maximize the likelihood of the chosen response over the rejected response by minimizing the following loss:

$$\ell_{\text{DPO}}(x, y_w^{\text{trunc}}, y_l^{\text{trunc}}; \theta; \pi_{\text{ref}}) :=$$
$$-\log\sigma\left(\beta\left[\log\frac{\pi_\theta(y_w^{\text{trunc}} \mid x)}{\pi_{\text{ref}}(y_w^{\text{trunc}} \mid x)} - \log\frac{\pi_\theta(y_l^{\text{trunc}} \mid x)}{\pi_{\text{ref}}(y_l^{\text{trunc}} \mid x)}\right]\right),$$

where π_θ is the probability distribution generated by the model, π_{ref} is the reference model's distribution, y_w^{trunc} and y_l^{trunc} represent the truncated winning and losing responses, and σ is the sigmoid function. In our approach, we train the DPO model on truncated responses, but it is still capable of generating full responses and performing in regular dialogues. The truncation helps to focus on the most relevant tokens early in the response, reducing noise from irrelevant parts of the response.

C Investigating the Autoregressive Influence on Preference Signals

In previous experiments, we observed that the preference signal appears to be concentrated in the initial portion of the response sequence. This could potentially be an artifact of the autoregressive nature of the data generation process. Given that the datasets used in earlier experiments were synthesized using autoregressive language models, we hypothesize that this phenomenon might be influenced by the autoregressive paradigm itself.

To validate this hypothesis, we conducted a series of experiments using human-generated responses and preference labels. Specifically, we employed the SHP dataset (Ethayarajh et al., 2022), which consists of responses and preference annotations generated by humans, to repeat the experiments outlined in subsubsection 4.2.1 and subsubsection 4.2.4.

C.1 Results

C.1.1 Performance on RewardBench

We trained reward models on the human-generated SHP dataset using both original and truncated versions of the responses. The evaluation was conducted on the RewardBench core set. The results, shown in Table 4, demonstrate that the shallow preference signal phenomenon persists even when using human-generated data.

C.1.2 DPO Implicit Reward Accuracy on Human-Generated Data

We also applied the DPO implicit reward approach to the truncated human-generated responses, as described in subsubsection 4.2.4, to predict the relative quality of response pairs. The accuracy of these predictions was then compared to human-annotated preferences. The results, shown in Figure 5, confirm that the shallow preference signal phenomenon persists even with human-generated data. As the truncation ratio decreases, the alignment between DPO implicit reward predictions and human-annotated preferences remains high, demonstrating that even truncated responses are sufficient for accurately predicting relative quality.

C.2 Conclusion

The results from the human-generated data experiments provide strong evidence that the observed shallow preference signal is not solely a byproduct of autoregressive data generation. Even when the data is generated by humans, the preference signal remains concentrated in the early portions of the response. This indicates that the phenomenon is likely inherent in the structure of the response itself, rather than an artifact of the autoregressive generation process.

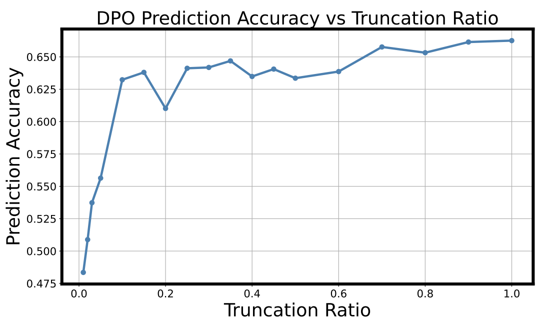
D Limitations

One limitation of this work is that the observed phenomena may have alternative explanations beyond the shallow preference signal we propose. Although our experiments support the hypothesis from multiple angles, some experimental outcomes might be influenced by other factors. For instance, in the experiment where the DPO model was trained on a truncated dataset, while our hypothesis accounts for the observed results, it is also possible that the DPO algorithm's inherent limitations could affect its performance, restricting its learning ability and hindering its capacity to fully capture human preferences beyond the initial token positions.

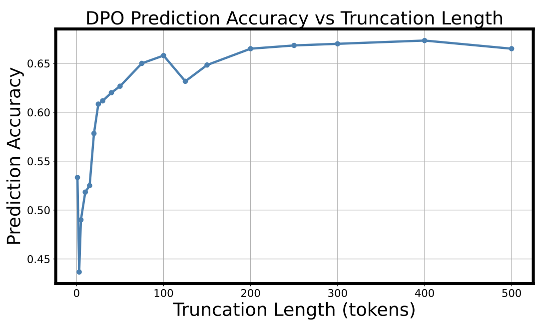
Another limitation is the absence of a strong theoretical foundation for the proposed phenomenon. Although our empirical results are compelling, a comprehensive theoretical explanation of the specific parts of a response that contribute to human preferences remains elusive. Future research could explore this aspect in more depth to establish a more robust theoretical framework.

Dataset	Dimension	Original Dataset	50%	40%	33%	25%
SHP-Preference	Chat	0.8198	0.8071	0.8139	0.7874	0.7709
	Chat-Hard	0.6039	0.6352	0.5759	0.5155	0.5274
	Safety	0.7906	0.8049	0.7825	0.7698	0.7589
	Reasoning	0.5624	0.5532	0.5439	0.5592	0.5451
	Total	0.7008	0.7056	0.6989	0.6882	0.6712

Table 4: Performance of reward models trained on the human-generated SHP dataset with different truncation ratios. The results show the evaluation scores across multiple dimensions: **Chat**, **Chat-Hard**, **Safety**, **Reasoning**, and **Total**. **Original Dataset** refers to the model trained on the full dataset without truncation; **50%**, **40%**, **33%**, and **25%** refer to datasets where the responses are truncated to retain 50%, 40%, 33%, and 25% of the original token length, respectively. The highest score in each row is highlighted with darker blue, and the second-highest score with lighter blue.



(a) Predict Accuracy vs Truncated Ratio



(b) Predict Accuracy vs Truncated Length

Figure 5: Accuracy of DPO implicit reward in predicting the relative quality of responses on the human-generated SHP dataset with truncated responses. The x-axis represents the truncation ratio and length, and the y-axis shows the accuracy of DPO implicit reward predictions compared to human annotations.