

Risk-Averse Failure Modes of Jensen-Shannon Distance in RLVR Distributional Alignment

Sebastian Loftus^{1*}, Xinpeng Wang^{2,3}, Marie-Catherine de Marneffe^{1,4}, Barbara Plank^{2,3}

¹CENTAL, UCLouvain, Belgium, ²CIS, LMU Munich, Germany,

³Munich Center for Machine Learning, Germany, ⁴FNRS, Belgium

sebastian.loftus@uclouvain.be, xinpeng@cis.lmu.de, marie-catherine.demarneffe@uclouvain.be,
b.plank@lmu.de

Abstract

Jensen-Shannon Distance (JSD) appears well-suited as a reward signal for aligning model predictions with human annotations, particularly for subjective tasks where label variation reflects legitimate disagreement. We demonstrate that JSD’s mathematical properties create exploitable reward landscapes in reinforcement learning. Via a study on preference distribution prediction, we show that two JSD-based reward formulations both fail to achieve alignment and instead lead to models converging at conservative, near-uniform predictions that maximize expected reward without matching human behavior. We trace this failure to JSD’s non-uniform sensitivity: it penalizes errors near balanced distributions less than errors near extreme distributions, creating risk asymmetry that incentivizes safe predictions over accurate ones. Our findings reveal fundamental challenges in using bounded symmetric divergence metrics as scalar rewards for distributional alignment.

1 Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) emerged as a powerful LLM post-training strategy, capable of eliciting complex reasoning behavior on structured tasks, like math or coding (Shao et al., 2024; DeepSeek-AI, 2025). For these types of tasks, verifiable reward functions allow for an objective evaluation of model outputs, rewarding sound reasoning behavior. Extending the method to non-deterministic tasks with an inherent degree of human label variation (HLV) (Plank, 2022) poses a larger challenge, as traditional verifiers fail at capturing the inherent model uncertainty of the predicted label, necessary to account for human variation. To quantify models’ capabilities to accurately represent a pluralistic label space, a common

approach is to measure the similarity of the two distributions using distance metrics. We investigate using the Jensen-Shannon distance (JSD) to formulate a scalar reward based on the divergence between human and model label distributions, which we in turn use to fine-tune LLMs via RLOO (Ahmadian et al., 2024) and on the HelpSteer2 preference prediction dataset (Wang et al., 2025a), where the task is to predict human preferences between two model-generated outputs. Rather than improving human alignment, the fine-tuned models exhibited a severe failure mode, collapsing into predicting purely balanced distributions. An analysis of the possible values JSD can take in binary tasks reveals that JSD’s non-uniform sensitivity makes it highly susceptible to reward hacking, rendering it an unsuitable reward estimator for this task. We detail which distance metric properties lead to the observed behavior and what this implies for distributional reward function design.

2 Related Work

An increasing body of work explores distributional rewards to capture subjective variation. Siththaranjan et al. (2024) frame preference learning as inherently distributional and model the reward as a distribution. Zhang et al. (2025) train a mean-variance reward model and optimize it using the KL-Divergence loss on preference data. Zawar et al. (2024) compare using KL-Divergence and JSD for reward modeling in Multi-Agent Constrained Policy Optimization and find that the latter offers a more robust alternative to KL, due to its symmetry. The most similar to our approach is Lu et al. (2026), who align LLM uncertainty to human annotator variation, wherein they define an uncertainty calibration reward based on the mean absolute error between model predictions and human annotator distributions and use GRPO for optimization. In our work, we show that for a similar approach,

*Work done primarily while the author was a Master’s student at LMU.

certain distance metrics, like JSD lead to a failure mode and detail what general properties in distance metrics lead to this behavior.

3 Reward Functions

This section presents the formulation of the two JSD-based reward functions.

3.1 JSD-based Reward Estimation

Jensen-Shannon Distance For measuring the similarity between the model-predicted distribution and human annotator judgments, we employ JSD, i.e., the bidirectional extension of the KL-divergence. An advantage of using the JS Distance compared to JS Divergence is the fact that it satisfies the requirements for being a metric, making it a better-scaled distance and thus better for interpreting the results (Endres and Schindelin, 2003). JSD is defined as:

$$\text{JSD} = \sqrt{\frac{1}{2}D_{\text{KL}}(H \parallel M) + \frac{1}{2}D_{\text{KL}}(M \parallel H)}$$

Reward Functions The following details the design of the JSD-based reward functions. For each function, the input is comprised of two probability distributions, M and H , over the possible labels, where M is the model-predicted distribution and H is the distribution of human-annotated labels.

For the **Direct Approach**, we define the reward as $R_{\text{direct}} = 1 - \text{JSD}(H||M)$, so that predictions closely matching the human distribution yield rewards near 1, while more divergent predictions approach 0.

Next, the **Binned Approach** estimates the reward by placing $\text{JSD}(H||M)$ into reward bins. By mapping JSD to specified rewards, we can provide larger reward signals for more accurate predictions, thereby rewarding exploitation. For the case that no parsable probability distribution is generated, the model is penalized with a score of -5 to encourage valid distributions. A detailed overview of the bins can be found in Table 1. Formally, this is defined as a piecewise function:

$$R_{\text{binned}} = \begin{cases} B(\text{JSD}), & \text{if output is valid} \\ -5, & \text{if output is unparsable} \end{cases}$$

where $B(\cdot)$ represents the mapping function that assigns the JSD score to its corresponding reward bin.

JSD	0.0	< 0.1	< 0.2	< 1.0	undefined
Reward	10	5	2	0	-5

Table 1: Bins for the Binned approach, where JSD is $\text{JSD}(H||M)$.

4 Experiment

The following section details the experimental setup based on Chen et al. (2024)’s setup for employing LLM-as-a-judge for distribution prediction for NLI.

4.1 Setup

Data We use the HelpSteer2-Preference (Wang et al., 2025b) dataset, which contains 7,118 prompts and preference pairs. To avoid overfitting to the most common distribution in the training data in the fine-tuning experiments, we prune the dataset to keep a more balanced set of distributions. For each unique distribution, we keep up to 120 samples, yielding 666 in total. These were split 70/15/15 into 466 training samples and 100 each for testing and validation; see Appendix A.8 for an example.

Models We use two models from different model families for the experiments to account for variance in model architecture. In an effort to reduce computational costs and to keep the necessary compute requirements to a minimum, smaller-scale models were chosen. The two models Qwen2.5-3B-Instruct,¹ and Llama-3.2-3B-Instruct² are henceforth referred to as Qwen and Llama.

Algorithm We use the REINFORCE Leave-one-out (RLOO) algorithm for reward-based fine-tuning. RLOO treats each complete generation as a single action and optimizes the reward for the entire sequence. It reduces variance by generating k samples per prompt and computing each sample’s baseline as the average reward of the other $k-1$ samples. The policy gradient is estimated by averaging these k individual gradient estimates (Williams, 1992; Ahmadian et al., 2024; Kool et al., 2019).

Implementation To extract the model’s probability distribution, we employ the verbalized probability expression approach, where the prediction is extracted directly from the LLM’s generated output.

¹<https://huggingface.co/Qwen/Qwen2.5-3B-Instruct>

²<https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct>

Compared to other expression methods, verbalization has been found to be the closest calibrated to human judgment (Meister et al., 2025). The final distribution was extracted from the generation using regular expressions. The detailed prompt can be found in Appendix A.1. To calculate the JSD, we used the jensenshannon method of the scipy.entropy library³ with the base e, bounding the possible values in $[0, \sqrt{\ln 2}] \approx [0, 0.833]$. To determine whether the failure mode is unique to the verbalization approach, we do additional tests using a logit-based approach, but after fine-tuning the model’s behavior did not change significantly and as such, we focused on the verbalized approach (see Appendix A.7 for further details on the logit approach).

4.2 Evaluation Metrics

To evaluate the method, we utilize distance correlation (Székely et al., 2007) and average Jensen-Shannon Divergence (JSD) to assess overall model performance across the test set. Specifically, distance correlation captures global trends, while average JSD evaluates performance at the instance level. Additionally, we calculate average entropy to analyze the distributional shape of the predictions both before and after fine-tuning.

Dist.-Corr. captures global dependence between model and human predictions (0 = independent, 1 = perfectly dependent), detecting both linear and nonlinear relationships, and is computed across the test set.

Average JSD is computed per instance and averaged over the test set and shows whether training was able to minimize the objective, but does not give information on the actual divergence of the distributions.

Average Entropy of the predictions before and after fine-tuning is reported to gain an understanding of the general shape of the model-predicted distributions, where values close to 1.0 indicate balanced distributions and 0.0 indicate imbalanced distributions.

4.3 Results

The following results demonstrate that the models converged to output balanced predictions (see Table 3). We focus on the change of the predicted

distributions before and after fine-tuning, as this is the main point of interest for the later discussion.

Distributions Analyzing the mean entropy shows that across both models and reward functions, the entropy rises above 0.94, indicating highly balanced predictions. Notably, Llama-base initially already had high average entropy, and as such, the difference before and after fine-tuning is not as large. Averaging the predictions shows that both base models show a pronounced position bias toward Response A - approximately [0.6/0.4] for Qwen and [0.7/0.3] for Llama, see Figure 3 in Appendix A.5. This bias, a well-documented issue in LLM evaluation (Wang et al., 2024), does not affect our validation study, as our goal is to demonstrate mode collapse, which remains evident even under the bias. This bias persists after fine-tuning in Qwen-direct, Llama-direct, and Llama-bins ([0.6/0.4]) and in Qwen-bins ([0.4/0.6]), which might be attributed to the inherent position bias of both models.

Model	Version	Dist.-Corr.↑	Avg. JSD↓	Avg. Entropy
Qwen	Base	0.243	0.276	0.604
	Direct	0.208	0.206	0.943
	Binned	0.095	0.210	0.940
Llama	Base	0.187	0.218	0.869
	Direct	0.201	0.205	0.944
	Binned	0.278	0.210	0.963

Table 2: Results of fine-tuning experiments. **Green**: improvement from base, **Red**: deterioration from base. Fine-tuning was able to minimize the target JSD in all settings, Dist.-Corr. improved using Llama and deteriorated with Qwen.

Performance Fine-tuning successfully reduced the average JSD across all models and functions, validating the effectiveness of the algorithm. Even though the predicted distributions collapsed, a minor correlation persists post-fine-tuning, with Dist.-Corr. values ranging from 0.095 to 0.278—likely a result of balanced instances appearing in the test set. Interestingly, while Dist.-Corr. decreased for Qwen, it unexpectedly improved over the baseline for Llama, a shift potentially explained by the exceptionally low baseline correlation Llama exhibited initially.

5 Discussion

To identify reasons for the previous results, we analyze the possible rewards the function can output, revealing risk-averse behavior due to the non-

³<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.entropy.html>

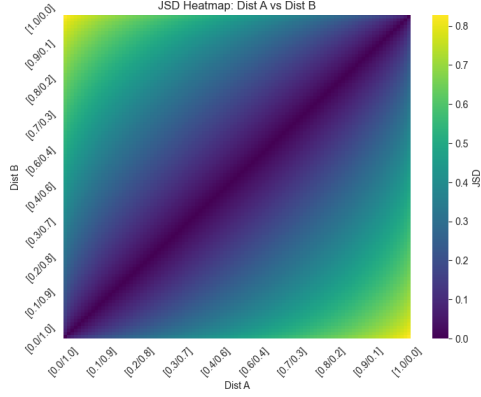


Figure 1: Values JSD can take with two-element distributions using base e . A large part of the surface is covered by values below 0.4.

uniform reward landscape of the reward functions and discuss other factors that might have contributed to the behavior.

5.1 Reward Landscape

To identify potential reasons for the failure mode, we analyzed the reward landscape, i.e., the values the rewards can take, using JSD as an optimization goal. Figure 1 shows the possible values $JSD(A||B)$ can take in a two-event case, where $A, B \in \mathcal{P}(R^2)$ are two-dimensional distributions, which is directly applicable to our binary preference task. The heatmap reveals non-uniform sensitivity: predictions near $[0.5/0.5]$ achieve $JSD < 0.4$, while extreme predictions face JSD ranging from 0.0 to 0.83 depending on the target distribution. In our case, if a model predicts highly-divergent distributions, it can either achieve very low JSD or very high JSD based on the human distribution, but if it predicts balanced predictions, no matter the target distribution, it gets a relatively good reward. This creates a risk asymmetry where conservative predictions are low-risk (bounded penalty) and extreme predictions are high-risk (variable penalty). Given this structure, RL optimization converges to near-uniform predictions that minimize expected penalty rather than accurately matching diverse human distributions.

5.2 Reasons for Failure

As previously discussed, the main issue of the method is the non-uniform sensitivity of the metric, as predictions close to the corners of the graph run the risk of producing larger errors, than those closer to the corners of the graph in Figure 1. This leads to strategies focusing on the exploitation of

the balanced and safe distributions without incentivizing exploration of more divergent predictions with a chance of receiving a smaller reward. A further issue might be RLOO’s leave-one-out baseline subtraction. When models generate diverse predictions, reward variance creates large advantages that destabilize training. Conservative predictions near $[0.5/0.5]$ yield consistent rewards and stable gradients, implicitly penalizing the exploration needed for accurate distribution matching.

5.3 Implications

Having identified what properties lead to this failure mode, we can identify other metrics and algorithms that share these characteristics and, as such, can also not be leveraged for reward design. In general, any bounded metric that creates a “safe zone” in the reward landscape, where moderate rewards can be earned across diverse targets without severe penalties, is prone to similar exploitation. Here, we focus on symmetric metrics. Appendix A.6 shows that KL-divergence, as an inherently asymmetric metric, exhibits comparable behavior. This risk-reward asymmetry makes conservative predictions a rational strategy from the model’s optimization perspective, even when they fail to achieve the intended alignment objective.

The observed behavior exemplifies reward hacking, i.e., models exploiting oversights in reward design to achieve high rewards through unintended solutions (Skalse et al., 2022). While reward hacking is well-documented in single-objective RL (Wang et al., 2023), we reveal a distinct failure mode: bounded symmetric divergences like JSD are inherently exploitable through risk aversion. Unlike typical reward hacking that exploits learned errors or specification ambiguities, our experiments suggest that JSD-based rewards fail due to their mathematical structure—the reward landscape itself incentivizes conservatism regardless of target distributions.

Our findings highlight the need to analyze reward landscapes before deployment, especially for distributional objectives where metric properties can create stable but misaligned incentives. While we showed a failure case of distribution similarity-based reward estimation, future work might try to increase the robustness of the methods by leveraging multi-objective rewards (Sorensen et al., 2024) or adding constraints to the reward.

Limitations

Our mathematical analysis focuses on two-event distributions, which directly apply to our binary preference task but may not fully characterize the reward landscape for multi-class problems. Future work should examine whether similar reward hacking emerges in higher-dimensional label spaces. Our experiments were done on a limited number of models with minimal hyperparameter-tuning, as we observed the failure after early experiments on smaller models and did not extend them to larger models. Furthermore, we discuss the usage of two distinct reward functions, which both show the described behavior, but did not fine-tune the specific hyperparameters, e.g., the size of the bins or the degree of the rewards and penalties. However, to generalize the findings to other metrics and models, more extensive tests might be beneficial. Due to computational constraints, we did not do extensive RLOO hyperparameter tuning. RLOO has previously been shown to be sensitive to failure modes under certain hyperparameters. We acknowledge that this is a contributing factor to the observed behavior, but the focus of this paper is JSD’s non-uniform sensitivity and why it contributes to the failure mode.

A possible other factor which affected the resulting failure case is the RLOO leave-one-out baseline-subtraction effect and further experiments with other optimization metrics are necessary to establish how much of the effect is attributable to the metric itself compared to the algorithm.

For most of our settings, there is a clear bias towards response A. This phenomenon is well documented and can be used to make one model seem stronger than the other simply because it came first in the order of responses (Wang et al., 2024). As we did not correlate the specific models that generated the responses to model performance, this bias did not directly impact any of our results, apart from being less aligned with human views. The dataset includes model responses from various LLMs with no models being consistently encoded as either response A or B. As such we did not randomly swap the position of the model responses for Furthermore, we aim to show the mode collapse after fine-tuning, which is still evident under a biased prediction.

Acknowledgments

This research was in parts supported by ERC Consolidator Grant DIALECT 101043235. Marie-Catherine de Marneffe is a Research Associate of the Fonds de la Recherche Scientifique – FNRS. Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Équipements de Calcul Intensif en Fédération Wallonie Bruxelles (CÉCI) funded by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region.

References

- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. [Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand. Association for Computational Linguistics.
- Beiduo Chen, Xinpeng Wang, Siyao Peng, Robert Litschko, Anna Korhonen, and Barbara Plank. 2024. [“seeing the big through the small”: Can LLMs approximate human judgment distributions on NLI from a few explanations?](#) In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14396–14419, Miami, Florida, USA. Association for Computational Linguistics.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). Preprint, arXiv:2501.12948.
- Dominik Maria Endres and Johannes E Schindelin. 2003. A new metric for probability distributions. *IEEE Transactions on Information theory*, 49(7):1858–1860.
- Wouter Kool, Herke van Hoof, and Max Welling. 2019. Buy 4 reinforce samples, get a baseline for free! *ICLR 2019 Deep Reinforcement Learning meets Structured Prediction Workshop*.
- Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.
- Junyu Lu, Deyi Ji, Xuanyi Liu, Lanyun Zhu, Bo Xu, Liang Yang, and Hongfei Lin. 2026. Aligning llm uncertainty with human disagreement in subjectivity analysis. *arXiv preprint arXiv:2605.10415*.
- Nicole Meister, Carlos Guestrin, and Tatsunori Hashimoto. 2025. [Benchmarking distributional alignment of large language models](#). In *Proceedings of*

- the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: *Human Language Technologies (Volume 1: Long Papers)*, pages 24–49, Albuquerque, New Mexico. Association for Computational Linguistics.
- Barbara Plank. 2022. [The “problem” of human label variation: On ground truth in data, modeling and evaluation](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Guangming Sheng, Chi Zhang, Zilinfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. [Hybridflow: A flexible and efficient rlhf framework](#). In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys ’25, page 1279–1297, New York, NY, USA. Association for Computing Machinery.
- Anand Siththaranjan, Cassidy Laidlaw, and Dylan Hadfield-Menell. 2024. [Distributional preference learning: Understanding and accounting for hidden context in RLHF](#). In *The Twelfth International Conference on Learning Representations*.
- Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward hacking. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS ’22, Red Hook, NY, USA. Curran Associates Inc.
- Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. 2024. Position: a roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning*, ICML’24. JMLR.org.
- Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. 2007. [Measuring and testing dependence by correlation of distances](#). *The Annals of Statistics*, 35(6):2769 – 2794.
- Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. [Large language models are not fair evaluators](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.
- Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. [How far can camels go? exploring the state of instruction tuning on open resources](#). In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2025a. [HelpSteer2-Preference: Complementing Ratings with Preferences](#). *arXiv preprint*. ArXiv:2410.01257 [cs].
- Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2025b. [Helpsteer2-preference: Complementing ratings with preferences](#). In *The Thirteenth International Conference on Learning Representations*.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.
- Rushikesh Zavar, Prabhdeep Singh Sethi, and Roshan Roy. 2024. Jensen-shannon divergence in safe multi-agent rl. In *The Second Tiny Papers Track at ICLR 2024*.
- Michael JQ Zhang, Zhilin Wang, Jena D. Hwang, Yi Dong, Olivier Delalleau, Yejin Choi, Eunsol Choi, Xiang Ren, and Valentina Pyatkin. 2025. [Diverging preferences: When do annotators disagree and do models know?](#) In *Forty-second International Conference on Machine Learning*.

A Appendix

A.1 Prompt

The prompt used for fine-tuning, where the model is tasked to first output a reasoning path and the probability distribution:

Please carefully and fairly base your selection on the comments below to estimate which model-generated output, labeled 'Response A' and 'Response B', is preferable given the provided query. Base your decision on the criteria: helpfulness, correctness, coherence, complexity, and verbosity. Please first give a reasoning detailing how you come to the decision, marked with the tag 'REASONING:'. After the reasoning steps, generate a probability distribution, for how likely each response is to be chosen, marked with the tags 'PROBABILITY A:' and 'PROBABILITY B:'

Instructions: {INSTRUCTIONS}

Response A: {RESPONSE A}

Response B: {RESPONSE B}

The output should look like this:

REASONING:

PROBABILITY A:

PROBABILITY B:

Answer:

A.2 Technical details

Fine-tuning with RL algorithms requires a large amount of compute due to multiple models being necessary in the training objective. Two A100 GPUs were used to fine-tune the model for 3 epochs. Lora was used for more efficient fine-tuning with rank 8 and alpha 32. The experiments were implemented using the verl-library⁴ (Sheng et al., 2025), which supports a plethora of Reinforcement Learning algorithms for model alignment. For fine-tuning the following hyperparameters were used: learning rate was set to 1e-6, batch size of 12, rollout of n=4 and the kl-coefficient for RLOO to 0.001.

⁴<https://github.com/volcengine/verl>

A.3 Distributions

This section presents the distributions of both the test data and the predicted distributions. After fine-tuning, there is a considerable shift towards balanced distributions.

A.4 Test Data

Distribution of the human annotations in the test data.

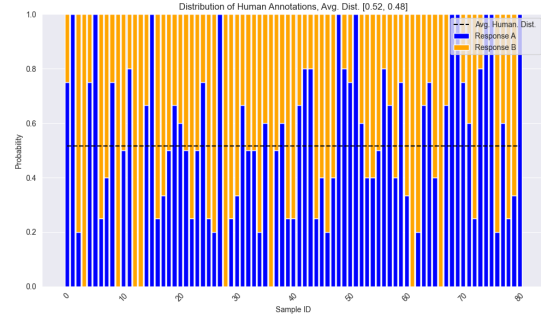


Figure 2: Distribution of the ratios of the human annotations for the valid test data. There is a wide variety of distributions in the test set.

A.5 Model Distributions

Figure 3 shows the predicted distributions on the test set of the raw and fine-tuned models. All models converge to near-uniform distributions.

A.6 Discussion of Asymmetric Metrics

In our study, we analyze the symmetric, bounded JSD and explain how it can generalize to other metrics, which show large "safe zones" in the reward landscape. Apart from such symmetric measures, other asymmetric ones warrant discussion, as their behavior might slightly differ. A common example of asymmetric divergence measures is the Kullback–Leibler divergence (Kullback and Leibler, 1951), which quantifies the information loss when one probability distribution is used to approximate another.

Running an analysis of its reward landscape (Figure 4) under the same two-event distribution setting as in our study reveals a very large "safe zone" with small values ranging from balanced distributions to more divergent ones. However, similar to JSD, there are high peaks in the corners where divergent predictions run the risk of achieving low rewards. This shows that the same issue persists in the asymmetrical metric, with the model preferring balanced distributions to avoid risking low rewards, albeit

less extreme than in JSD due to the large "safe zone". This suggests that our claim also generalizes to asymmetric distribution measures.

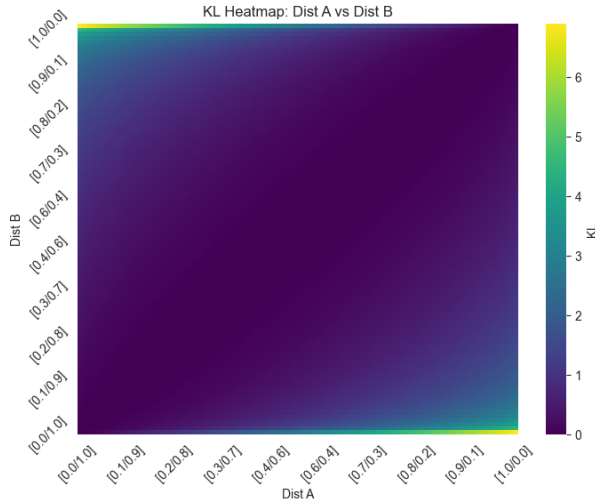


Figure 4: Values KL-divergence can take with two-element distributions. Similarly to the JSD reward landscape, the surface is covered by low values.

A.7 Logit-based probability expression

A common alternative to verbalized probabilities is the logit-based expression method, where a probability distribution is inferred by extracting specific logits from the model’s distribution over the vocabulary and normalizing it. Using the direct approach, we fine-tuned Qwen3.1-1.7B on the complete helpsteer dataset. Prior to fine-tuning, the model substantially overestimated one of the labels (avg. entropy of predictions = 0.04), and this behavior remained essentially unchanged following fine-tuning. Moreover, neither of the evaluation metrics exhibited any improvement after fine-tuning, suggesting that the adopted approach did not effectively alter the model’s behavior. A plausible explanation is that, because the initial model policy was strongly biased toward unbalanced label distributions, the model primarily received rewards for predictions corresponding to these unbalanced instances. Consequently, it failed to adequately explore more balanced distributions and instead continued to prioritise the unbalanced cases. For this experiment, the learning rate and the KL-coefficient in RLOO were set to $1e-5$ and 0.0005, respectively.

Version	Dist.-Corr.↑	Avg. JSD↓	Avg. Entropy
Base	0.04	0.43	0.04
Direct	0.04	0.44	0.08

Table 3: Results of the fine-tuning experiments. The low mean entropy shows that the predicted outcomes were highly unbalanced.

A.8 Sample

Table 4 shows a sample from the test set. The prompt is composed of instructions on how to solve the task, two model outputs from distinct models, and a template of how to format the output. The lower part details the models’ predictions after fine-tuning for both approaches and shows the JSD.

Distribution of Predicted Probabilities

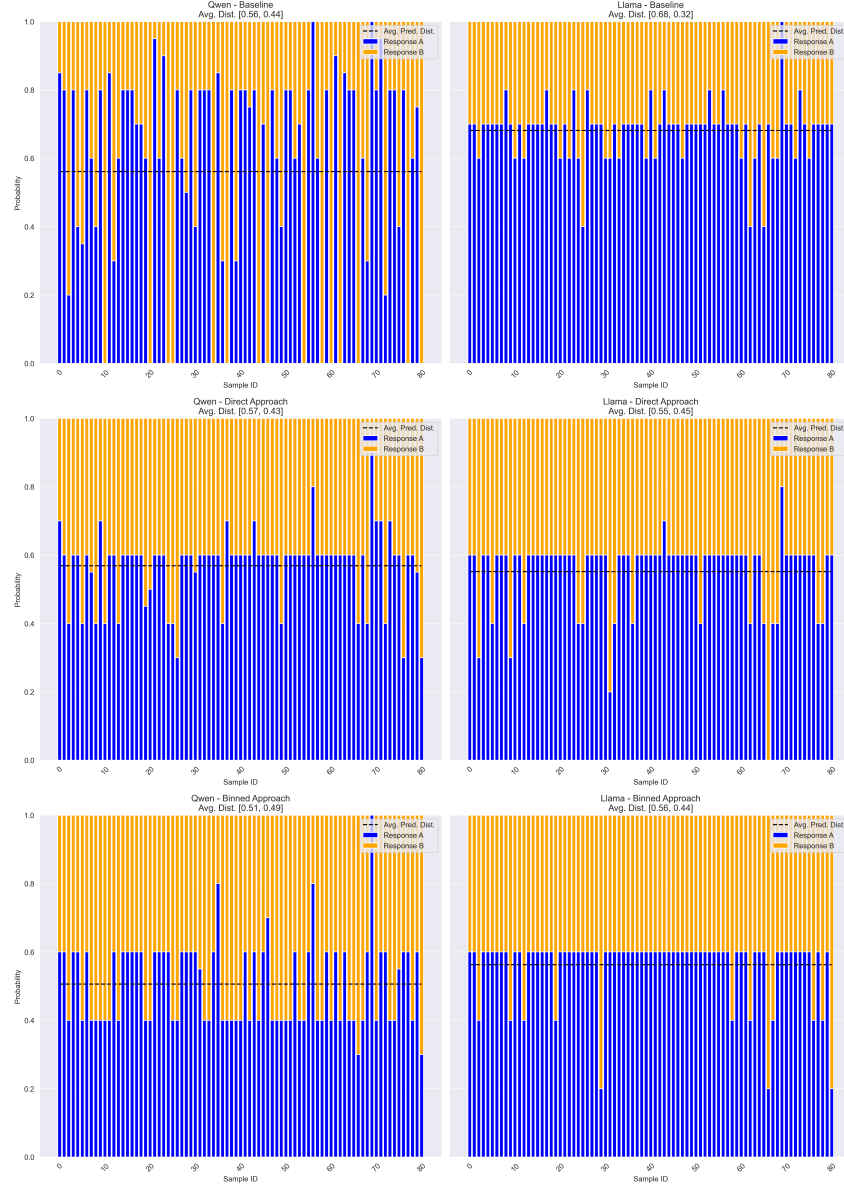


Figure 3: Overview of predicted Distributions. Each bar is made up of **Response A** at the bottom and **Response B** at the top. The average prediction is provided per model and highlighted in the plots as a dotted line. The Llama models and Qwen-direct tended to predict a [0.6/0.4] distribution, and Qwen-bins predicted [0.4/0.6].

Prompt			
Please carefully and fairly base your selection on the comments below to estimate which model generated output, labeled 'Response A' and 'Response B', is preferable given the provided query. Base your decision on the criteria: helpfulness, correctness, coherence, complexity, verbosity. Please first give a reasoning detailing how you come to the decision, marked with the tag REASONING:. After the reasoning steps, generate a probability distribution, for how likely each response is to be chosen, marked with the tags 'PROBABILITY A:' and 'PROBABILITY B:'			
Instructions: Make it to an academic writing setting.			
The observations to be made of the experiment was through data analysis of the data. that was that as depth or height increased so to as the pressure value. In the experiment involving the center of gravity apparatus, as the water, weight, and depth are proportionally direct to each other.			
Response A: Sure, here's a revised version of your sentence that is more appropriate for academic writing:			
As the depth or height of the experiment increased, so did the pressure value. In the experiment involving the center of gravity apparatus, the water, weight, and depth were proportionally direct to each other.			
Response B: In the experiment involving the center of gravity apparatus, the data analysis showed a direct proportionality between water weight, depth, and pressure value. As the depth or height increased, so did the pressure value. This is in accordance with the theory that water weight and depth are directly proportional to each other. This experiment provides evidence to support the idea that as the depth of a liquid increases, so does its pressure value. This is due to the fact that the weight of the liquid is distributed over a larger area, resulting in a greater pressure. This experiment also provides evidence to support the idea that the center of gravity of an object can be determined by measuring the pressure values at different points on the object.			
The output should look like this:			
REASONING:			
PROBABILITY A:			
PROBABILITY B:			
Answer:			
Method	Predicted Dist.	Human Dist.	JSD↓
Direct	[0.3, 0.7]	[0.2, 0.8]	0.0819
Binned	[0.4, 0.6]	[0.2, 0.8]	0.1554

Table 4: Random sample taken from the test-set of a formatted prompt containing the task and both model outputs. The predicted distribution and model distributions are presented along with the achieved JSD.