Why Jensen-Shannon Distance Fails as a Reward Signal for Distribution Matching

Anonymous ACL submission

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 (RL) has become a key framework for aligning large language models (LLMs) with human preferences. Traditional approaches train a reward model on human-annotated preferences and optimize it with PPO (Schulman et al., 2017; Ouyang et al., 2022) or the simpler DPO, which directly estimates generation quality. Recent methods like GRPO (Shao et al., 2024) and RLOO (Ahmadian et al., 2024) extend this reward-based approach by computing numerical rewards that measure task performance.

A core challenge in this pipeline is the design of the reward function. Reward specification is highly task-dependent, and poorly defined objectives can incentivize misaligned or brittle behaviors. Previous successes mainly dealt with structured tasks like maths or code, where it is easy to find rule-based reward functions ((DeepSeek-AI, 2025; Shao et al., 2024)). Extending the method to more subjective tasks, like Natural-Language-Inference (NLI) or sentiment analysis, with a degree of human label variation (HLV) (Plank, 2022) poses a larger challenge due to the inherent uncertainty of the model prediction. Under this ambiguity, there is no longer one true label, but rather a distribution over multiple labels, each carrying valuable information, which, if ignored, can lead to marginalization of underrepresented groups. To compare a predicted distribution to a human distribution, divergence measures, like the KL-divergence (Kullback and Leibler, 1951) or its symmetrical extension, Jensen-Shannon divergence (Endres and Schindelin, 2003), can be used to measure the similarity between the two distributions. Given these measures, it might seem natural to leverage such divergence measures in a reward function to directly quantify their similarity in a scalar reward and use this to estimate the quality of a generation. In this paper, we show why the non-uniform sensitivity of the distribution-based divergence metrics makes them unsuitable for reward estimation, as they are easy to game by targeting balanced distributions. We validate these claims by providing a case study on using the Jensen-Shannon distance for binary preference distribution prediction and explain how it generalizes to other measures by analyzing the properties of the KL-divergence. Finally, we give further reasons for the failure mode and discuss its implications in distributional reward function design.

2 Related Work

RL for LLM alignment relies on reward functions that score model outputs. For objective tasks (math, code), rule-based rewards achieve strong results (Shao et al., 2024). However, subjective tasks with HLV resist simple scalar rewards, as they fail to capture legitimate disagreement among annotators. Recent work explores distributional rewards to cap-

ture subjective variation. Siththaranjan et al. (2024) frame preference learning as inherently distributional and model the reward as a distribution. Most similar to our approach Zhang et al. (2025) train a mean-variance reward model and optimize it using KL-Divergence loss on preference data. Our work differs by using distribution metrics directly as reward signals, encouraging models to match human annotation distributions and revealing a failure mode unique to distance-based distributional rewards.

3 Methodology

This section presents the formulation of our JSD-based rewards and their possible values based on JSD's properties.

3.1 JSD-based Reward Estimation

Jensen-Shannon Distance JSD is a common metric to measure the distance between two probability distributions, which is the square root of the JSDivergence (Endres and Schindelin, 2003). It is based on the KL-divergence (Kullback and Leibler, 1951) while having the advantage of being bidirectional and well defined for all probability distributions. An advantage of using the JSDistance compared to JSDivergence 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).

Reward Functions The following details the design of the reward functions, both designed around the Jensen-Shannon Distance, which allows for instance-wise comparison of the distance of the human and model distributions. For each function, the input is comprised of two probability distributions, M and H, across all possible labels for, where M is the model-predicted distribution and H is the distribution of human-annotated labels.

First, the **Direct Approach** employs the Jensen-Shannon Divergence (JSD) as a measure of similarity between distributions. In RL, rewards need to be defined such that higher values correspond to better solutions. However, JSD measures dissimilarity, meaning lower values indicate closer alignment between two distributions. This makes it unsuitable to use JSD directly as a reward signal. To address this, we define the reward as $R = 1 - JSD(H||M)$, 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 binning $JSD(H||M)$ into reward bins. By mapping JSD to specified rewards, we can provide larger reward signals for very close predictions, thereby rewarding the model for exploring reward spaces that may lead to closer alignment. For the case that no parsable probability distribution is generated, the model is punished and gets a -5 penalty to encourage valid distributions. A detailed overview of the bins can be found in Table 1.

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 $JSD(H||M)$.

3.2 Reward Landscape

Before conducting experiments, we analyze the theoretical reward landscape to identify potential failure modes. 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, we hypothesize that RL optimization may converge to near-uniform predictions that minimize expected penalty rather than accurately matching diverse human distributions. Our experiments test whether this predicted failure mode occurs in practice.

4 Case study - Preference disagreement

To show the failure of the method, we present a case study on the preference distribution prediction task, where, given some instructions, two model outputs are labeled by multiple annotators and compared with a model prediction. The methodology used in this study is based on Chen et al. (2024)'s setup for employing LLM-as-a-judge for distribution prediction for NLI.

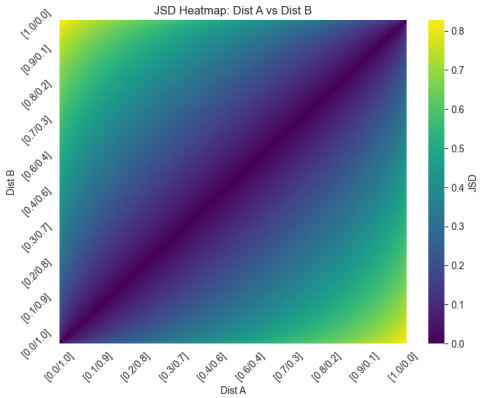

Figure 1: Values JSD can take with two-element distributions using base e. A large part of the surface is covered by values below 0.4.

4.1 Experimental Setup

Data The dataset used is HelpSteer2 (Wang et al., 2025), 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 diverse 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 E 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 first model, Qwen2.5-3B-Instruct,[1] subsequently referred to as **`Qwen`**, is part of the Qwen2.5 family of models, and the second model used in the experiments is Llama-3.2-3B-Instruct[2] or **`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).

[1] https://huggingface.co/Qwen/Qwen2.5-3B-Instruct

[2] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

Implementation To extract the model's probability distribution, we employ a verbalized probability estimation approach that expresses the distribution in the generated output using numerical values. The final distribution was extracted from the generation using regular expressions. The detailed prompt can be found in Appendix A. To calculate the JSD, we used the jensenshannon method of the scipy.entropy[3] library (Virtanen et al., 2020) with the base e, bounding the possible values in $[0, \sqrt{\ln 2}] \approx [0, 0.833]$.

4.2 Evaluation

To evaluate the method, Distance-correlation (Székely et al., 2007) and Average JSD are used to get an overview of the models' performance across the test-set on a global (Dist.-Corr.) and instance-wise (Average JSD) level.

Metrics. **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.

Expected result Both metrics are compared before and after fine-tuning: while JSD is expected to decrease as it is minimized during training, Dist.-Corr. may worsen, reflecting the tendency of JSD-based rewards to produce conservative distributions, independent of the target distribution.

4.3 Results

The following results on the case study validate our predicted failure mode, with all tuned models predicting distributions close to [0.5/0.5] and the evaluation metrics showing both improvements and deterioration after tuning.

Distributions In the baseline, both models show a pronounced position bias toward Response A - approximately [0.6/0.4] for `Qwen` and [0.7/0.3] for `Llama`. 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. The `Qwen` baseline model also shows a

[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.entropy.html

high degree of variance with many highly divergent samples. As expected, the models largely predict balanced distribution after fine-tuning with predictions approaching [0.5/0.5]. Remaining skewness appears in Qwen-JSD, Llama-JSD, and Llama-bins ([0.6/0.4]) and in Qwen-bins ([0.4/0.6]), which can be attributed to the inherent position bias of both models. See Figure C.2 in the Appendix for the detailed distributions

Performance Across both models and functions, fine-tuning was able to diminish Avg. JSD, showing the success of the algorithm. Given the even distributions, the predicted distributions were expected to be close to independent of the human distributions. There is, however, still a slight correlation after fine-tuning with Dist.-Corr. values ranging from 0.095 to 0.278, probably due to balanced instances appearing in the test set. Surprisingly, Dist.-Corr. improved over the baseline for Llama, which might be due to the very low Dist.-Corr. the baseline model showed to begin with, but decreased for Qwen.

Model	Version	Dist.-Corr.↑	Avg. JSD↓
Qwen	Base	0.243	0.276
	Direct	0.208	0.206
	Binned	0.095	0.210
Llama	Base	0.187	0.218
	Direct	0.201	0.205
	Binned	0.278	0.210

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.

5 Discussion

As predicted by the reward landscape analysis, models converged to near-uniform predictions. These results validate the claim that JSD is unsuitable for reward modeling. Both the Direct and Binned approach led to the same observation, leading us to deduce that it is the metric itself that causes the failure mode. The following discusses further reasons for the failure and its implications.

5.1 Reasons for failure

As previously discussed, the main issue of the method is the non-uniform sensitivity of the metric, with target predictions close to the center having a smaller margin of error 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. RLOO's leave-one-out baseline subtraction exacerbates this problem. 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.

5.2 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 D 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.

This exemplifies reward hacking, where models exploit 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, 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 applies 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 case study was done on a limited number of small models with minimal hyperparameter-tuning, as the focus of the paper was not to solidify the method, but to validate the hypothesis of the failure mode. However, to generalize the findings to other metrics and models, more extensive tests might be beneficial.

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. Furthermore, we aim to show the mode collapse after fine-tuning, which is still evident under a biased prediction.

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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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.

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

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, Zilingfeng 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.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, and 16 others. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

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. 2025. 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.

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 Prompt

The prompt used for fine-tuning was the following and contains an additional reasoning prompt for an Explain than Predict approach :

> 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:

B 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-library[4] (Sheng et al., 2025), which supports a plethora of Reinforcement Learning algorithms for model alignment.

C 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.

[4] https://github.com/volcengine/verl

C.1 Test Data

Distribution of the human annotations in the test data.

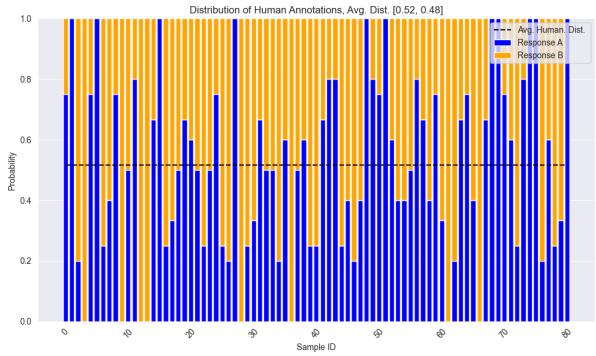

Figure 2: Distribution of the ratios of the human annotations for the valid test data. There is a wide variety of distributions in the test set.

C.2 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.

D Discussion of non-symmetrical 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.

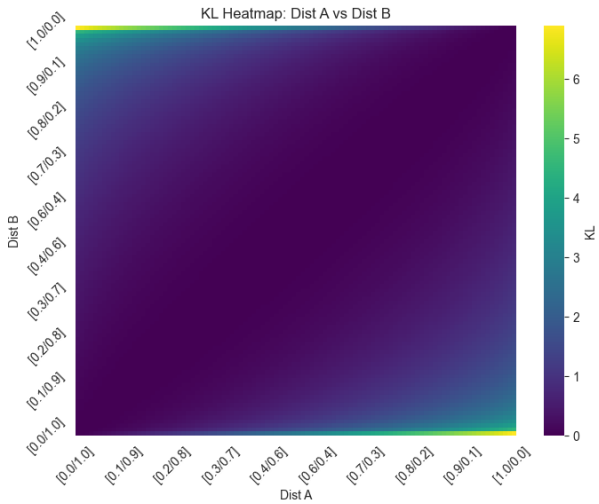

Figure 4: Distribution of the ratios of the human annotations for the valid test data. There is a wide variety of distributions in the test set.

E Sample

Table 3 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.

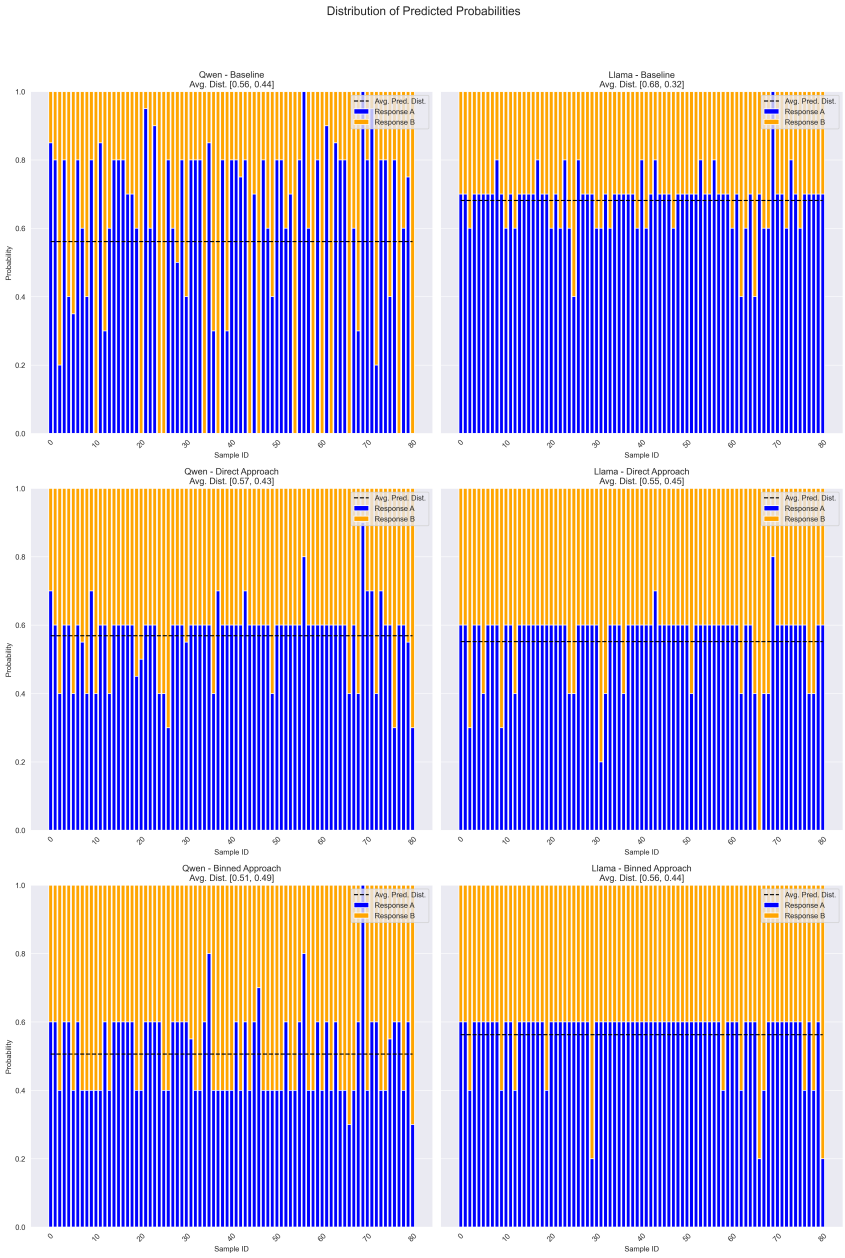

Figure 3: Predicted Distributions for the RL-version, where Response A is at the bottom and Response B is at the top. The average prediction is provided per model and highlighted in the plots as a dotted line. The `Llama` models and `Qwen`-JSD tended to predict a [0.6/0.4] distribution, and `Qwen`-bins overwhelmingly 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 3: 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.