Reward Model Underspecification in Language Model Alignment

Jacob Eisenstein[1] **Jonathan Berant**[1] **Chirag Nagpal**[2] **Alekh Agarwal**[2]

Ahmad Beirami[2] **Alex D'Amour**[1] **DJ Dvijotham**[1]

Katherine Heller[2] **Stephen Pfohl**[2] **Deepak Ramachandran**[2]

1. Google DeepMind
2. Google Research
reward-model-underspecification-distshift-2023@google.com

Abstract

Reward models play a key role in aligning language model applications towards human preferences. However, this setup can create a dynamic in which the policy model has the incentive to exploit errors in the reward model to achieve high reward. This means that the success of reward-based alignment depends on the ability of reward models to transfer to new distributions created by the aligned policy model. We show that reward models are *underspecified*, in the sense that models that perform similarly in-distribution can yield very different rewards on policy model outputs. These differences propagate to the aligned policies, which we show to be heavily influenced by the random seed used during *pretraining* of the reward model. We show that even a simple alignment strategy — best-of-n reranking — creates a semi-adversarial dynamic between the policy and reward models, promoting outputs on which the reward models are more likely to disagree. Finally, we show that a simple ensembling strategy can help to address this issue.

1 Introduction

To align machine learning systems with human preferences, it has become common practice to use *reward models*, which score potential outputs by how likely they are to be preferred by human raters [CLB$^+$17, SOW$^+$20, BJN$^+$22, RFS$^+$23]. There are several ways to use reward models to align policy models: they can act as training signals in reinforcement learning [CLB$^+$17, SOW$^+$20], they can select examples for further fine-tuning [GPS$^+$23], or they can be applied at inference time to select a high-reward output from a set of samples [e.g., GSH23]. Such architectures create a semi-adversarial dynamic between the reward model and the policy model: the policy model can try to get high reward by exploiting errors in the reward model. Furthermore, while the reward model is trained on a fixed set of human preference data, the process of alignment requires it to provide feedback on data from a shifting target distribution, increasing the likelihood of such errors. The success of reward model-based alignment therefore depends on whether it is possible to train reward models that perform well out-of-distribution.

In this paper, we explore reward model distribution shift from the perspective of underspecification [DHM$^+$22], which occurs when a machine learning pipeline yields reliable performance on held-out data from the training distribution, but variable performance out-of-distribution. For example,

Workshop on Distribution Shifts, 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

[SYW$^+$21] pretrain and fine-tune 25 BERT-base checkpoints, finding that while their in-distribution performance is consistent, their performance on out-of-distribution "challenge sets" varies significantly across pretraining random seeds, with all other aspects of the training pipeline held constant. One implication is that evaluation of individual *artifacts*, such as model checkpoints, cannot yield accurate assessment of the out-of-distribution performance of training *procedures*. For that, it is necessary to evaluate multiple runs of the training procedure. We explore the implications of these ideas and results for language model alignment.

In general, alignment can fail for at least two reasons: 1) the reward model itself does not accurately model human preferences, e.g., because it performs poorly out-of-distribution; 2) the alignment *procedure* fails to adequately incorporate the reward model, e.g., because the aligned policy overfits or underfits the reward signal. To focus on the first issue, we restrict consideration to a simple inference-time alignment procedure, best-of-n reranking: we generate n samples from the policy produced by supervised fine-tuning (SFT) and then return the one with the highest estimated reward. The procedure is described in more detail at the beginning of section 2.

Prior work on reward model robustness Prior work has explored reward over-optimization — sometimes called "reward hacking" or "Goodharting" — from several perspectives [KUM$^+$20, SHKK22, PBS22]. In reinforcement learning from human feedback (RLHF), reward overoptimization is controlled by regularizing Kullback-Leibler divergence between the aligned policy and a reference policy, typically obtained from supervised fine-tuning. But the effectiveness of this tradeoff is limited by our ability to measure reward accurately, which has been called into question by other work [BJN$^+$22, GSH23]. [BJN$^+$22] train two reward models on non-overlapping splits of the preference annotations, using one to drive alignment (via either reinforcement learning or reranking), and the other to measure the quality of the outputs. They find that RLHF increases performance according to both the driver and measurement models, but that a performance gap emerges as the policy is allowed to diverge further from the SFT distribution. However, both reward models were built on the same *pretraining* data, which, as we will show, limits their diversity (as hypothesized by [GI22]) and thus may understate the effect of reward hacking. Other work on has simulated the relationship between a "true" reward and a learned proxy, showing that it is possible to overoptimize the proxy to such an extent that the true reward starts to decrease [GSH23]. This phenomenon has been replicated in more realistic settings by examining (and creating) spurious correlations in reward model training data [PPS$^+$23].

In offline reinforcement learning, uncertainty on out-of-distribution data points has been quantified by ensembles [AMKS21]. Concurrent work, which appeared after the submission of this paper, also examines the impact of ensembling on best-of-n reranking and reinforcement learning [CAKK23]. Using synthetic data in the style of [GSH23], they find that ensembles improve generalization in both reranking and RLHF. None of this prior work considers how *pretraining* determines the OOD behavior of the learned reward model, which may exert a significant and arbitrary impact on the aligned policy.

Contributions

- We show how even best-of-n alignment can induce distribution shifts that make reward models less reliable.
- We connect this lack of reliability to underspecification, demonstrating that the behavior of reward models (and the corresponding policies) is strongly affected by the random seed used during pretraining.
- We propose a simple mitigation for reward model underspecification: pretrain ensemble reward models, which aggregate over multiple pretrains.

2 Underspecification in Reward Models

We explore underspecification in the context of best-of-n reranking, a simple but effective approach which has been shown to improve rewards while limiting divergence from the fine-tuned (SFT) policy model [GSH23, RSM$^+$23]. For prompt x, let $y^{(s)}$ indicate sample $s \in \{1 \dots n\}$ from the policy. For reward model m, we select the reward-maximizing sample, $\hat{y}_m = \arg\max_{s \in 1 \dots n} r_m(y^{(s)}; x)$.

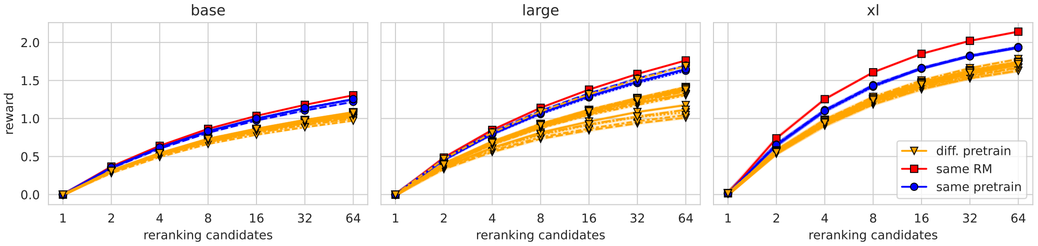

Figure 1: Reward of the best-of-n reranked output, as judged by: the ranker reward model itself (squares); reward models fine-tuned from the same pretrain as the ranker (circles); reward models fine-tuned from different pretrains from the ranker (triangles). The reward models that do not share a pretrain with the ranker regard its outputs as significantly worse, particularly for larger scale reward models. Confidence intervals reflect finite-sample variance.

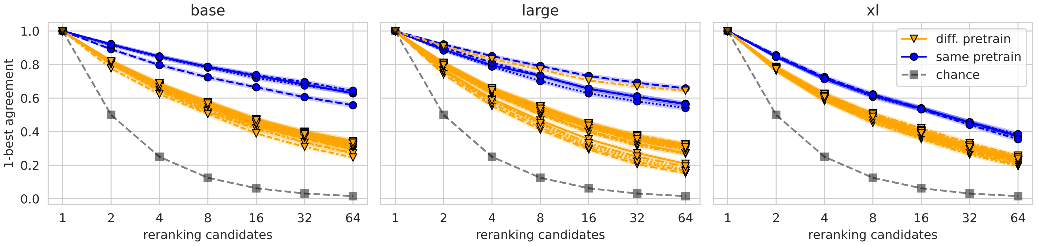

Figure 2: Agreement of the top-ranked summary between reward models that do (circles) and do not (triangles) share pretrainings. Underspecification of reward models directly affects the behavior of the aligned policy. Chance agreement is $1/n$. Confidence intervals reflect finite-sample variance.

The Kullback–Leibler divergence of this policy from the SFT policy has upper bound $\log n - \frac{n-1}{n}$, and it generally outperforms more elaborate alignment techniques like RLHF in the low-KL regime [GSH23], albeit with the cost of generating multiple samples at inference time. In the experiments that follow, we consider $n \in \{2^1, 2^2, \ldots 2^6\}$.

To demonstrate underspecification, we compute three relationships: (1) the reward under reward models that do *not* drive the reranking, i.e. $r_{m'}(\hat{y}_{m \neq m'})$; (2) the agreement across reward models, i.e. $\delta(\hat{y}_m, \hat{y}_{m' \neq m})$; (3) the relationship between the variance of rewards and the number of reranking candidates, i.e. $\mathbb{V}[r_m(\hat{y}_{m'})]$. In each case, a single reward model (per scale) is used to rerank candidate outputs from the policy, and then the other models are used to score the top-ranked outputs.

2.1 Experimental setup

We explore these questions in the context of the TLDR summarization task and preference annotations [VPSS17, SOW+20].[1] We build on a fine-tuned T5-large policy model (770M parameters), which was trained on reference summaries, with maximum input and output lengths of 1024 and 128 tokens respectively, a dropout rate of 0.1, a constant learning rate of 10^{-3}, and a batch size of 128. To examine the effect of pretraining on the reward models, we pretrain five T5 models at each of the base (220M), large (770M), and XL (3B) scales, using a denoising objective on the C4 corpus [RSR+20]. The pretrained checkpoints differ only in their random seed, which controls the parameter initialization and the pretraining data. We then finetune the TLDR reward model by optimizing a Bradley-Terry objective over pairs of preferred/dispreferred summaries [SOW+20], minimizing the log sigmoid of the difference in rewards between the dispreferred and preferred outputs. The Bradley-Terry model is underdetermined with respect to constant shifts in the rewards, which is problematic for ensembles based on order statistics such as min and median, as well as for

[1] We obtained similar results in pilot studies on the XSUM dataset [NCL18] with an NLI-based reward model [NWD+19, RFS+23], but we omit them here due to space limitations.

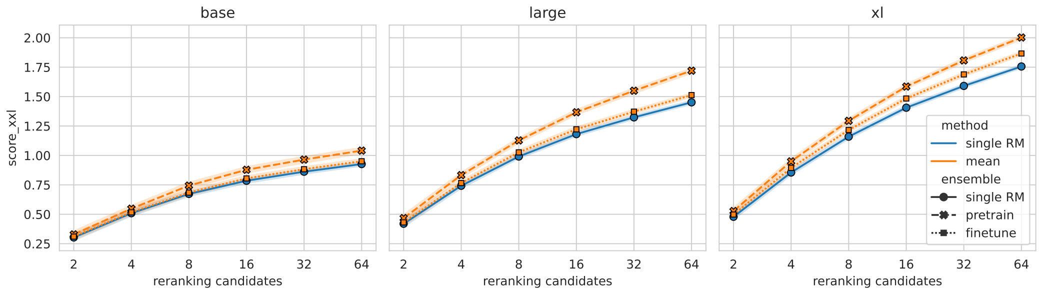

Figure 4: Pretrain ensemble reward models (PERMs) significantly improve the quality of summarization outputs, as measured by a more capable XXL-scale evaluation model.

approaches based on the variance of rewards [e.g., CAKK23]. For this reason, we add an additional regularization-like term $\alpha||r_+ + r_-||_2^2$, which encourages the rewards to be centered around zero.

To examine the effect of random seed at finetuning time, we finetune one pretrained checkpoint five times, with five distinct random seeds. Because our focus is on reward model underspecification, we do not explore the effect of pretraining random seed on the policy models.

2.2 Evidence of Underspecification

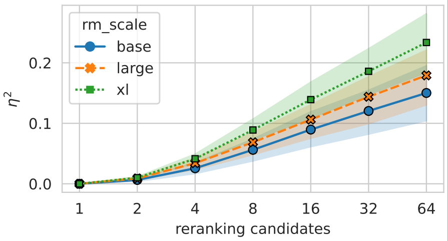

Figure 3: Fraction of variance explained by variation across reward models.

On held-out in-distribution data, the pairwise accuracy of different reward model checkpoints is similar: 65.4–66.7% (base), 68.8–70.0% (large), and 71.2–72.9% (XL). However, when we move out-of-distribution we observe significant disagreements between the reward models on the quality of candidate summaries, as shown in Figure 1. The red line (square markers) shows the expected reward of the top-ranked output according to the ranker itself. Below the XL scale, other reward models from the same pretrain tend to score these outputs highly, although a gap emerges at the XL scale. But in all cases, these outputs are scored significantly less favorably by reward models which do not share a pretrain with the ranker. We emphasize that the pretrains differ only by random seed.

These differences in estimated rewards induce different policies from the best-of-n alignment: as shown in Figure 2, different reward models tend to produce different 1-best outputs. Again these differences are strongly associated with the pretraining seed: for example, two reward models from different pretrains will choose a different best-of-16 output more than half the time.

Finally, we investigate whether the effect of underspecification is increased by stronger forms of alignment, which drive the policy further from the initial policy as represented by the SFT model. We compute a matrix of instance-reward model scores $X_{i,m} = r_m(y_i; x_i)$, and apply ANOVA to compute the fraction of the variance in scores that is explained by per-model offsets. As shown in Figure 3, this fraction increases with the number of reranking candidates, which indicates stronger alignment and greater divergence from the initial policy.

3 Pretrain Ensemble Reward Models

If reward model behavior is strongly associated with the pretraining seed, then a simple mitigation for underspecification is to build an ensemble of reward models, each from a different pretrain: $\overline{r}(y; x) = \text{agg}(\{r_m(y; x)\}_{m \in \mathcal{M}}$, with agg indicating an aggregation function and $\mathcal{M}$ a set of reward models [Die00, LPB17, RSR$^+$20, ZZE$^+$21]. As before, we build on five different pretrains, constructing five corresponding reward models. We consider simple aggregation functions: MEAN, MEDIAN, MIN, and MEAN MINUS STDEV. Before aggregation, each reward model is standardized to zero mean reward per prompt. To evaluate the quality of the best-of-n outputs under these rewards, we train an evaluation model by finetuning a T5-XXL checkpoint. The evaluator is thus more capable than any of the ranker models (it achieves a 79.5% pairwise accuracy on heldout in-distribution data) and does not share a pretraining checkpoint with any member of the ensemble.

According to the evaluator, the ensembled reward models yield significantly higher rewards than individual reward models (Figure 4). Win rates over the SFT policy were also significantly higher: for example, at $n = 64$ and XL-scale, the win rate was 90.0% for the pretrain-ensemble reward model, versus 87.7% for a fine-tune ensemble and 85.7% for an individual reward model. Differences between the aggregation functions were smaller, with the MEAN aggregation function performing best at the XL scale and the more conservative MIN and MEAN MINUS STDEV aggregators performing best at the base scale (see Table 1 in the supplement). In pilot studies we compared pretrain ensembles to aggregations of MC Dropout samples from the reward model, and found that pretrain ensembles showed a consistent advantage even when aggregating a much larger number of MC Dropout samples [similar to LPB17]. Future work will compare these possibilities more rigorously.

4 Conclusion

Online alignment methods like RLHF and reranking rely on the robustness of reward models to distributional shift. However, we find that the OOD performance of reward models are underspecified, with a strong dependence on the random seed used during pretraining. Pretrain ensembles can help to address this issue, yielding significantly better performance over individual reward models. However, producing and maintaining multiple pretrains is expensive, suggesting that more efficient ensembling approaches should be sought in future work [e.g., WIG+22]. Another topic for future work is to move beyond reranking. Of particular interest are alignment techniques that enable exploration, such as RLHF. These methods offer the opportunity to seek greater rewards by moving further from the SFT policy, but at the cost of imposing even greater distribution shift on the reward model. Such techniques thus stand to benefit from robustness-promoting architectures and learning objectives.

Acknowledgments Thanks to Sharat Chikkerur, Mohammad Havaei, and the anonymous reviewers for feedback on this paper. The research also benefited from feedback from David Bruns-Smith, Ming-Wei Chang, Michael Collins, Patrick Fernandez, Adam Fisch, Mandar Joshi, Rishabh Joshi, Balaji Lakshminarayanan, Kenton Lee, Pete Shaw, Kristina Toutanova, Victor Veitch, and Zihao Wang. Finally, we thank the people who built the infrastructure used in our experiments, including the T5X team and Léonard Hussenot, Johan Ferret, Robert Dadashi, Geoffrey Cideron, Alexis Jacq, Sabela Ramos, Piotr Stanczyk, Sertan Girgin, Danila Sinopalnikov, Amélie Héliou, Bobak Shahriari, Bilal Piot, Matt Hoffmann, Nikola Momchev, and Olivier Bachem.

References

[AMKS21] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

[BJN+22] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[CAKK23] Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*, 2023.

[CLB+17] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[DHM+22] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *The Journal of Machine Learning Research*, 23(1):10237–10297, 2022.

[Die00] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[GI22] Adam Gleave and Geoffrey Irving. Uncertainty estimation for language reward models. *arXiv preprint arXiv: 2203.07472*, 2022.

[GPS$^+$23] Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

[GSH23] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

[KUM$^+$20] Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane Legg. Specification gaming: the flip side of ai ingenuity. *DeepMind Blog*, 3, 2020.

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[NCL18] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[NWD$^+$19] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

[PBS22] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. *arXiv preprint arXiv:2201.03544*, 2022.

[PPS$^+$23] Richard Yuanzhe Pang, Vishakh Padmakumar, Thibault Sellam, Ankur Parikh, and He He. Reward gaming in conditional text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4746–4763, Toronto, Canada, July 2023. Association for Computational Linguistics.

[RFS$^+$23] Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Leonard Hussenot, Orgad Keller, Nikola Momchev, Sabela Ramos Garea, Piotr Stanczyk, Nino Vieillard, Olivier Bachem, Gal Elidan, Avinatan Hassidim, Olivier Pietquin, and Idan Szpektor. Factually consistent summarization via reinforcement learning with textual entailment feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6252–6272, Toronto, Canada, July 2023. Association for Computational Linguistics.

[RSM$^+$23] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

[RSR$^+$20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[SHKK22] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

[SOW^{+}20] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[SYW^{+}21] Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, et al. The multiberts: Bert reproductions for robustness analysis. *arXiv preprint arXiv:2106.16163*, 2021.

[VPSS17] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, 2017.

[WIG^{+}22] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022.

[ZZE^{+}21] Sheheryar Zaidi, Arber Zela, Thomas Elsken, Chris C Holmes, Frank Hutter, and Yee Teh. Neural ensemble search for uncertainty estimation and dataset shift. *Advances in Neural Information Processing Systems*, 34:7898–7911, 2021.

Supplementary Material

4.1 Autoeval results

Numerical results for the ensembles are shown in Table 1.

scale	ensemble	method	reward	win rate
base	finetune	mean	0.951 ± 0.010	0.698 ± 0.005
		mean minus stdev	0.995 ± 0.010	0.704 ± 0.005
		median	0.938 ± 0.010	0.694 ± 0.005
		min	1.004 ± 0.010	0.706 ± 0.005
	pretrain	mean	1.041 ± 0.013	0.715 ± 0.005
		mean minus stdev	1.085 ± 0.012	0.724 ± 0.005
		median	1.018 ± 0.012	0.711 ± 0.005
		min	1.099 ± 0.012	0.726 ± 0.005
	single RM	single RM	0.927 ± 0.009	0.691 ± 0.004
large	finetune	mean	1.513 ± 0.009	0.810 ± 0.004
		mean minus stdev	1.516 ± 0.009	0.812 ± 0.004
		median	1.489 ± 0.009	0.806 ± 0.004
		min	1.523 ± 0.009	0.814 ± 0.004
	pretrain	mean	1.720 ± 0.011	0.847 ± 0.004
		mean minus stdev	1.687 ± 0.011	0.843 ± 0.004
		median	1.688 ± 0.010	0.842 ± 0.004
		min	1.630 ± 0.010	0.836 ± 0.004
	single RM	single RM	1.451 ± 0.008	0.797 ± 0.004
xl	finetune	mean	1.866 ± 0.009	0.877 ± 0.003
		mean minus stdev	1.807 ± 0.009	0.867 ± 0.003
		median	1.815 ± 0.009	0.868 ± 0.003
		min	1.817 ± 0.009	0.868 ± 0.003
	pretrain	mean	2.002 ± 0.010	0.900 ± 0.003
		mean minus stdev	1.947 ± 0.009	0.890 ± 0.003
		median	1.944 ± 0.009	0.890 ± 0.003
		min	1.921 ± 0.009	0.887 ± 0.003
	single RM	single RM	1.756 ± 0.008	0.857 ± 0.003

Table 1: Auto-evaluator rewards and win rate versus the SFT model on TLDR, with standard errors.