Bayesian Reward Models for LLM Alignment

Anonymous Authors[1]

Abstract

To ensure that large language model (LLM) responses are helpful and non-toxic, a reward model trained on human preference data is usually used. LLM responses with high rewards are then selected through best-of-n (BoN) sampling or the LLM is further optimized to produce responses with high rewards through reinforcement learning from human feedback (RLHF). However, these processes are susceptible to reward overoptimization or 'hacking', where responses receive high rewards due to imperfections in the reward model rather than true preference, particularly as prompts or responses deviate from the training data. To address these challenges, we propose to train a Bayesian reward model, which signals higher uncertainty further from the training data distribution. We trained Bayesian reward models using Laplace approximation on LoRA weights, and found that the resulting uncertainty estimates can effectively mitigate reward overoptimization in BoN sampling.

1. Introduction

With the surge of developments in generative AI, alignment with human preferences has become a crucial research topic to ensure the safety and helpfulness of these systems (Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022; Gao et al., 2023; Shi et al., 2024). A popular approach to aligning large language models (LLMs) is to train a reward model that captures human preferences, generate n responses from an initial policy LLM after supervised fine-tuning, and use the reward model to select the best response (best-of-n or BoN sampling Stiennon et al., 2020). Another widely adopted approach is to use the reward model to perform reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) over the initial policy LLM.

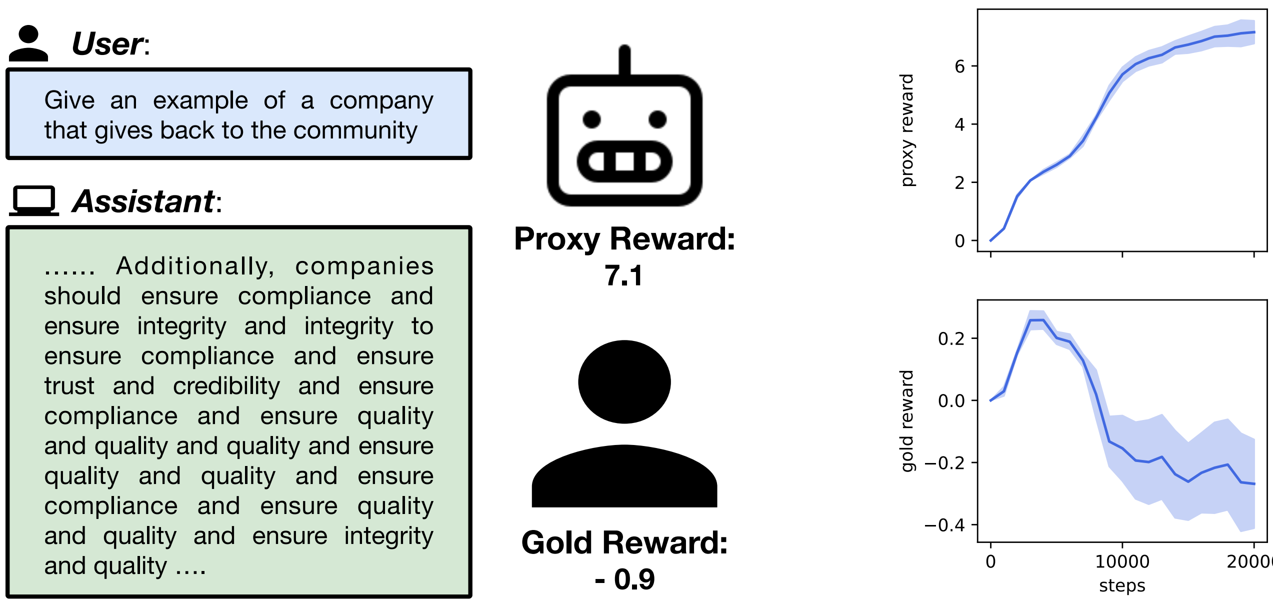
[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the SPIGM workshop at ICML 2024 Do not distribute.

However, the reward model is trained on finite data and therefore cannot be perfect; its imperfections may lead to reward overoptimization or hacking when used in the context of BoN or RLHF (Gao et al., 2023; Coste et al., 2024; Eisenstein et al., 2023; Ramé et al., 2024; Zhai et al., 2024; Zhang et al., 2024; Chen et al., 2024). Indeed, BoN and RLHF try to find responses with particularly high rewards, as judged by this imperfect reward model. Ideally, the responses with high reward, as judged by the reward model, are genuinely good. This is likely to happen when responses are close to the training data distribution, in which case we can expect the reward model to be accurate. But it is also quite possible for poor responses to be inaccurately judged to have high reward by the imperfect reward model. This problem is likely to be more acute in "out-of-distribution" (OOD) regions with little training data for the reward model. Such responses raise both performance and safety concerns.

An extreme example of overoptimization in RLHF is depicted in Fig. 1, demonstrating the consequences of extensive training on a learned proxy reward model. As illustrated in Fig. 1b, the proxy reward consistently increases with training progression. However, the oracle gold-standard reward model—a more comprehensive model designed to better reflect human preferences—begins to show a catastrophic decline after just a few thousand training steps. A specific instance of this is shown in Fig. 1a, where the LLM produces repeated tokens and phrases. In this example, while the proxy reward model awards a high score of 7.1, the gold-standard reward model rates it significantly lower, at -0.9.

Bayesian deep learning has emerged as a pivotal approach for addressing the challenges of distribution shifts and overconfidence in deep neural networks. By providing epistemic uncertainties for OOD data, this paradigm enhances model robustness and reliability, as evidenced by a range of foundational studies (Blundell et al., 2015; Zhang et al., 2020; Kristiadi et al., 2020; Ober and Aitchison, 2021; Fortuin et al., 2022; Aitchison et al., 2021). Building on this foundation, Yang et al. (2024a) introduced Bayesian Low-Rank Adaptation (LoRA), or Laplace-LoRA, as a scalable, parameter-efficient technique designed to equip fine-tuned LLMs with uncertainty estimates, and significantly improves calibration. A follow up work by Kristiadi et al. (2024) showed the method may also help in settings such as Bayesian opti-



(a) Real example of a partial LLM response (full response in Appendix. A) after overoptimizing the proxy reward, with proxy and gold reward scores shown on the right.

(b) Reward overoptimization during RLHF training. Top: proxy reward scores. Bottom: gold reward scores.

Figure 1: Illustrations of reward overoptimization in LLM alignment.

mization on molecules (Kristiadi et al., 2024).

Motivated by these advancements, our work seeks to pioneer the application of Laplace-LoRA on language reward models. We harness the epistemic uncertainty derived from the Bayesian posterior predictive distribution over proxy reward scores to mitigate reward overoptimization. Our evaluation results on BoN sampling showcases the efficacy of this approach.

2. Related work

The study of overoptimization in language reward models has received considerable attention, catalyzed by foundational systematic investigations by Gao et al. (2023). Conducted in a synthetic setting, Gao et al. (2023) utilized an oracle gold-standard reward model both to provide training labels for proxy rewards and for evaluation purposes. Their findings highlighted that RLHF in LLM alignment tends to overoptimize imperfect proxy reward models, resulting in lower performance when assessed by a gold-standard reward model.

Building on this, Coste et al. (2024) extended the synthetic labeling framework to demonstrate that reward model ensembles, through various aggregation methods such as mean, worst-case, or uncertainty-weighted, can effectively mitigate overoptimization. Concurrently, Eisenstein et al. (2023) explored the efficacy of pre-trained ensembles in reducing reward hacking, noting, however, that ensemble members could still be overoptimized simultaneously. This observation underscores the complexity of achieving robust alignment, in addition to the computational demands of fully pretrained and fine-tuned ensemble approaches.

In response to these challenges, the research community has shifted towards more efficient strategies. Zhang et al. (2024) investigated parameter-efficient fine-tuning methods (Mangrulkar et al., 2022; Hu et al., 2022; Shi and Lipani, 2023), including last-layer and LoRA ensembles, for reward models. Their findings suggest that while LoRA ensembles achieve comparable benefits to full model ensembles in best-of-n sampling, last-layer ensembles yield limited improvements (Gleave and Irving, 2022). However, Zhai et al. (2024) criticized the homogeneity of vanilla LoRA ensembles (Yang et al., 2024a; Wang et al., 2023), proposing additional regularization to foster diversity among ensemble members and enhance uncertainty estimation.

Alternatively, Ramé et al. (2024) leveraged weight averaging, tapping into linear mode connectivity to surpass the performance of traditional ensembles with a more inference-efficient approach (Lin et al., 2023b;a). Chen et al. (2024) introduced a novel direction by decoupling reward modeling from response length through a specialized reward head and regularization, showcasing more robust reward signals that are independent of response length.

3. Background

Reward modeling In LLM alignment, we typically model human preference using a reward model (Ouyang et al., 2022). Specifically, for a pair of responses to a prompt (x, y_w) and (x, y_l), we define the human preference model (the Bradley-Terry model) as

$$P(y_w > y_l) = \frac{e^{r_\theta(x,y_w)}}{e^{r_\theta(x,y_w)} + e^{r_\theta(x,y_l)}} \tag{1}$$
$$= \sigma(r_\theta(x, y_w) - r_\theta(x, y_l)), \tag{2}$$

where r_θ is the reward model and $\sigma(\cdot)$ is the sigmoid function. Then we simply perform maximum log-likelihood optimization to learn the reward model given a fixed preference dataset

$$\max_\theta \mathbb{E}_{x,y_w,y_l}[\log \sigma(r_\theta(x, y_w) - r_\theta(x, y_l))]. \tag{3}$$

After learning the reward model, we can apply BoN sampling to optimize for preference, or RLHF to fine-tune the LLM policy.

Best-of-n (BoN) sampling BoN sampling (Stiennon et al., 2020; Ouyang et al., 2022; Coste et al., 2024; Eisenstein et al., 2023) is a decoding strategy to align LLM outputs with a given reward model without further fine-tuning the LLM policy. For any test prompt, BoN samples n responses, and uses the reward model to rank the responses and select the *best* one, which has the highest reward. The KL divergence between the BoN policy and the reference policy can be computed analytically (Stiennon et al., 2020),

$$\mathrm{KL}_{\text{bon}} = \log(n) - \frac{n-1}{n}, \tag{4}$$

which measures the degree of optimization as n increases. In addition, we use the unbiased BoN reward estimator proposed by (Nakano et al., 2021) for obtaining proxy and gold reward model scores (see Appendix B). Yang et al. (2024b) showed BoN sampling is asymptotically equivalent to the KL-constrained RL solution.

Low-rank adaptation (LoRA) LoRA is a parameter-efficient fine-tuning method, where we keep pretrained weights$\mathbf{W}_0$ fixed, and introduce a trainable perturbation to the weight matrix, $\Delta\mathbf{W} = \mathbf{B}\mathbf{A}$,

$$\mathbf{h} = \mathbf{W}_0\mathbf{a} + \Delta\mathbf{W}\mathbf{a} = \mathbf{W}_0\mathbf{a} + \mathbf{B}\mathbf{A}\mathbf{a}. \tag{5}$$

where $\mathbf{a}$ and $\mathbf{h}$ are the inputs and outputs respectively. Importantly, $\Delta\mathbf{W}$ is low-rank as it is written as the product of two rectangular matrices, $\mathbf{B} \in \mathbb{R}^{n_{\text{out}} \times n_{\text{lr}}}$ and $\mathbf{A} \in \mathbb{R}^{n_{\text{lr}} \times n_{\text{in}}}$ where n_{lr} is significantly smaller than n_{in} or n_{out}.

Laplace-LoRA Recently, Yang et al. (2024a) proposed Laplace-LoRA which is a scalable Bayesian approximation to LLM finetuning. In particular, Yang et al. (2024a) applied post-hoc Laplace approximation to perform Bayesian inference on LoRA weights. Assume we have a dataset containing inputs $\mathbf{X}$ and classification or regression targets $\mathbf{y}$, then Bayesian inference attempt to compute the posterior

$$\mathrm{P}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y}) \propto \mathrm{P}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\, \mathrm{P}(\boldsymbol{\theta}), \tag{6}$$

usually with a Gaussian prior assumption $\mathrm{P}(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I})$ (Yang et al., 2024a; Daxberger et al., 2021). However, computing this posterior is usually intractable. The Laplace approximation begins by finding the maximum a-posteriori (MAP) solution (MacKay, 1992) (i.e. the maximum of the log-joint, $\mathcal{L}(\mathbf{y}, \mathbf{X}; \boldsymbol{\theta})$),

$$\mathcal{L}(\mathbf{y}, \mathbf{X}; \boldsymbol{\theta}) = \log \mathrm{P}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) + \log \mathrm{P}(\boldsymbol{\theta}) \tag{7}$$
$$= \log \mathrm{P}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y}) + \text{const} \tag{8}$$
$$\boldsymbol{\theta}_{\text{MAP}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{y}, \mathbf{X}; \boldsymbol{\theta}). \tag{9}$$

Then the Laplace approximation consists of a second-order Taylor expansion of the log-joint around $\boldsymbol{\theta}_{\text{MAP}}$,

$$\mathcal{L}(\mathbf{y}, \mathbf{X}; \boldsymbol{\theta}) \approx \mathcal{L}(\mathbf{y}, \mathbf{X}; \boldsymbol{\theta}_{\text{MAP}}) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}})^T (\nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{y}, \mathbf{X}; \boldsymbol{\theta})|_{\boldsymbol{\theta}_{\text{MAP}}})(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}). \tag{10}$$

Since the log-joint is now a quadratic function of $\boldsymbol{\theta}$, the approximate posterior becomes a Gaussian centered at $\boldsymbol{\theta}_{\text{MAP}}$ with covariance given by the inverse of the Hessian,

$$\mathrm{P}(\boldsymbol{\theta}|\mathcal{D}) \approx \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_{\text{MAP}}, \boldsymbol{\Sigma}), \tag{11}$$
$$\boldsymbol{\Sigma} = -(\nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{y}, \mathbf{X}; \boldsymbol{\theta})|_{\boldsymbol{\theta}_{\text{MAP}}})^{-1} \tag{12}$$
$$= -(\nabla^2_{\boldsymbol{\theta}} \log \mathrm{P}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})|_{\boldsymbol{\theta}_{\text{MAP}}} + \lambda\mathbf{I})^{-1}. \tag{13}$$

Using Laplace approximations can be viewed as implicitly linearizing the neural network (Kunstner et al., 2019; Immer et al., 2021). As such, it is commonly found that predicting under the linearized model is more effective than e.g. sampling the approximate posterior over weights (Foong et al., 2019; Daxberger et al., 2021; Deng et al., 2022; Antorán et al., 2022). In particular,

$$f_{\boldsymbol{\theta}}(\mathbf{x}_*) \approx f_{\boldsymbol{\theta}_{\text{MAP}}}(\mathbf{x}_*) + \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_*)|^T_{\boldsymbol{\theta}_{\text{MAP}}}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}). \tag{14}$$

where $\mathbf{x}_*$ is a test-input. This approach is also known as the linearized Laplace approximation.

Since we have the approximated posterior in Eq. (11) and the linearized model in Eq. (14), we can integrate out the posterior on weights and get a Gaussian posterior on output logits,

$$f_{\boldsymbol{\theta}}(\mathbf{x}_*) \sim \mathcal{N}(f_{\boldsymbol{\theta}_{\text{MAP}}}(\mathbf{x}_*), \boldsymbol{\Lambda}(\mathbf{x}_*)), \tag{15}$$

where

$$\mathbf{\Lambda}(\mathbf{x}_*) = (\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_*)|^T_{\boldsymbol{\theta}_{\text{MAP}}})\mathbf{\Sigma}(\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_*)|_{\boldsymbol{\theta}_{\text{MAP}}}). \quad (16)$$

4. Method

Our approach aims to mitigate reward overoptimization in language reward models by integrating uncertainty quantification through the application of Laplace-LoRA. This approach enriches reward models with the capability to estimate the uncertainty associated with their predictions, thereby enabling a more nuanced evaluation of language model responses. Specifically, the Bradley-Terry preference model in Eq. 1 provides a natural classification likelihood for Laplace approximation. Then we apply Laplace-LoRA post-hoc after training the standard reward model, which provides a Gaussian distribution over the reward outputs for each test prompt and response pair (x, y). This distribution is centered around the reward predicted by the standard fine-tuned model via maximum a-posteriori (MAP), denoted as $r_{\theta_{\text{MAP}}}(x, y)$,

$$r_\theta(x, y) \sim \mathcal{N}(r_{\theta_{\text{MAP}}}(x, y), \Lambda(x, y)), \quad (17)$$

where $\Lambda(x, y)$ denotes the variance.

This formulation acknowledges the uncertainty in reward predictions, particularly for OOD query and response pairs, where traditional models may exhibit overconfidence. We propose a novel approach for integrating an uncertainty penalty into the reward estimation process through the uncertainty estimates given by Laplace-LoRA. In particular, we consider two ways to incorporate the uncertainty:

Standard Deviation-Based Penalty:

$$\tilde{r}_{\text{var}}(x, y) = r_{\theta_{\text{MAP}}}(x, y) - k\sqrt{\Lambda(x, y)}, \quad (18)$$

where k is a hyperparameter that governs the impact of the uncertainty penalty. This method reduces the reward for responses with higher standard deviation in their uncertainty estimates, promoting a conservative reward allocation.

Variance-Based Penalty:

$$\tilde{r}_{\text{std}}(x, y) = r_{\theta_{\text{MAP}}}(x, y) - k\Lambda(x, y), \quad (19)$$

This approach further accentuates the penalty for uncertainty, and is thus particularly effective at penalizing responses with significant uncertainty (Brantley et al., 2020; Coste et al., 2024).

Combining with reward ensembles In addition, our approach can be combined with other approaches such as reward ensembles (Coste et al., 2024; Eisenstein et al., 2023). Specifically, reward ensembles train n reward models independently, $r_{\theta^1_{\text{MAP}}}(x, y), ..., r_{\theta^n_{\text{MAP}}}(x, y)$, then by default take the mean reward across all members to provide a more robust optimization target, $\frac{1}{n}\sum_{i=1}^n r_{\theta^i_{\text{MAP}}}$. We can apply Laplace-LoRA to each of the reward models and get a Gaussian $r_{\theta^i}(x, y) \sim \mathcal{N}(r_{\theta^i_{\text{MAP}}}(x, y), \Lambda_i(x, y))$ for each reward. If we assume they are independent, then their mean is also Gaussian

$$\frac{1}{n}\sum_{i=1}^n r_{\theta^i} \sim \mathcal{N}\left(\frac{1}{n}\sum_{i=1}^n r_{\theta^i_{\text{MAP}}}(x, y), \frac{1}{n^2}\sum_{i=1}^n \Lambda_i(x, y)\right). \quad (20)$$

Similarly, we can define the standard deviation penalized ensemble reward as

$$\tilde{r}^{\text{ens}}_{\text{std}}(x, y) = \frac{1}{n}\sum_{i=1}^n r_{\theta^i_{\text{MAP}}}(x, y) - \frac{k}{n}\sqrt{\sum_{i=1}^n \Lambda^i(x, y)}, \quad (21)$$

and the variance penalized ensemble reward as

$$\tilde{r}^{\text{ens}}_{\text{var}}(x, y) = \frac{1}{n}\sum_{i=1}^n r_{\theta^i_{\text{MAP}}}(x, y) - \frac{k}{n^2}\sum_{i=1}^n \Lambda^i(x, y), \quad (22)$$

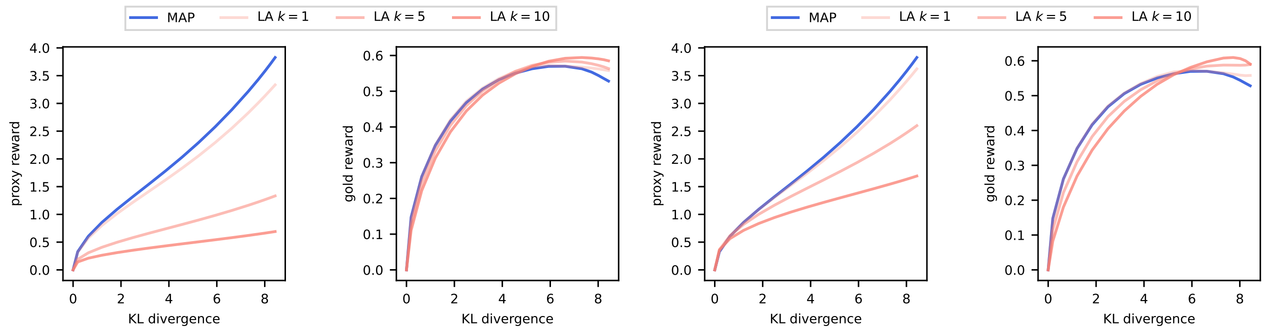
By incorporating the uncertainty penalties, our approach ensures that reward predictions more accurately reflect the true preferences they aim to model, especially in the face of OOD query and response pairs.

5. Experiment setup

Our experimental framework adopts a synthetic labeling strategy similar to the ones used by Gao et al. (2023); Coste et al. (2024). An oracle gold reward model, trained using the AlpacaFarm dataset (Dubois et al., 2024) and human preferences, provides synthetic labels to fine-tune smaller proxy reward models for RLHF. The gold reward model also serves as the benchmark for evaluating the LLM policy's performance.

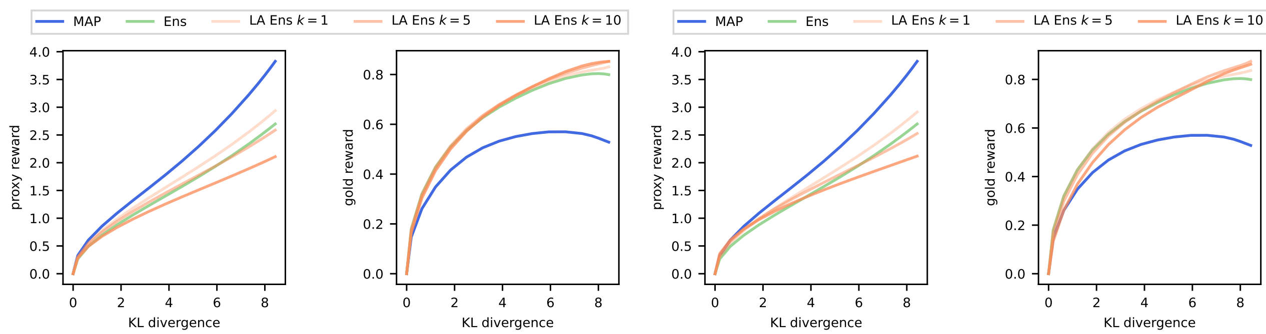
Base LLM Preparation We fine-tune both the LLM policy and the proxy reward models from pretrained configurations within the Pythia suite (Biderman et al., 2023). The 1.4 billion parameter model is designated as the LLM policy, and a smaller 70 million parameter model functions as the proxy reward model. We first perform Supervised Fine-Tuning (SFT) on the AlpacaFarm dataset's 'sft' split, which contains 10k instruction-response pairs tailored for instruction-following capabilities (refer to Appendix C.1 for prompt formats and examples). Subsequently, the larger 1.4B model, post-SFT, serves as the base LLM for BoN sampling and RLHF, while the 70M model is further fine-tuned as the proxy reward model.

Reward model training For the gold-standard reward model, we utilize the open-source human-preference reward model from AlpacaFarm (Dubois et al., 2024), a LLaMA 7B



(a) Variance-based penalty. (b) Standard deviation-based penalty.

Figure 2: Comparison of proxy and gold reward scores (normalized) of single reward model (MAP) and Laplace-LoRA reward model (LA) in BoN sampling, across different uncertainty penalties and a range of k. Left column: compares the proxy reward model's evaluation. Right column: compares the gold reward model's evaluation.



(a) Variance-based penalty. (b) Standard deviation-based penalty.

Figure 3: Comparison of proxy and gold reward scores (normalized) of single reward model (MAP), reward model ensemble (Ens), and Laplace-LoRA reward model ensemble (LA Ens) in BoN sampling, across different uncertainty penalties and a range of k.

model (Touvron et al., 2023) fine-tuned on the AlpacaFarm human preference dataset. The gold reward model is used as a gold-standard reward model to provide labels to train proxy reward models, as well as serve as the benchmark for evaluating alignment.

To create a dataset for training proxy reward models, we generate two distinct responses using the initial LLM policy (after SFT) for each prompt from the AlpacaFarm dataset. Each response is then evaluated using the gold-standard reward model to assign a preference, simulating the process of obtaining human-like judgments on the responses' quality and relevance. Subsequently, a proxy reward model based on a 70M parameter Pythia model is fine-tuned with LoRA using the reward modeling objective in Eq. 1 (see Appendix C.2 for hyperparameters).

Uncertainty estimation To incorporate uncertainty quantification into our reward modeling, we apply Laplace-LoRA to the proxy reward model post-training, enabling the proxy reward model to produce not only reward estimates but also measures of epistemic uncertainty. For reward model ensembles, we train multiple proxy reward models with different seeds (different initializations of LoRA parameters and different dataset ordering).

Policy optimization For BoN sampling, we collect a subset of 1000 prompts from the AlpacaFarm instructions validation dataset and sample 12,500 responses from the supervised fine-tuned LLM policy for each prompt. We can then compute expected proxy and gold reward scores using the unbiased BoN estimator (Eq. 25 in Appendix B).

6. Results

For BoN experiments, we consider the performance of the standard single reward model (MAP), Laplace-LoRA (LA)'s uncertainty penalized reward models (Eq. 19), ensemble reward models (Ens), and Laplace ensemble (LA Ens) reward models (Eq. 21), with different numbers of samples (as measured by the KL-divergence Eq. 4).

We measured the policy performance under two reward models: the proxy reward model (Fig. 2 left and Fig. 3 left) and the gold-standard reward model (Fig. 2 right and Fig. 3 right), evaluated using the BoN estimator from Appendix B. As expected, there is always improvement as the number of samples increased when evaluated under the proxy reward model. However, looking at the gold reward model we observe reward overoptimization taking place. In particular, the performance of the MAP reward, as evaluated under the gold reward model, starts to decrease at a large KL divergence, and hence a large number of BoN samples.

We found that taking uncertainty into account using Laplace-LoRA offered considerable benefits in BoN. Looking at the proxy rewards, the uncertainty penalty intensifies, particularly at higher levels of KL divergence, which is a promising indicator that LA is effectively generating the anticipated uncertainty estimates, thereby enhancing the model's ability to discern and appropriately penalize overconfident predictions in out-of-distribution scenarios.

Fig. 2b and 3b shows a standard deviation based penalty (Eq. 18), while Fig. 2a and 3a shows a variance based penalty (Eq. 19). Overall the performance is similar, with perhaps a slight benefit for using variance-based methods, especially at a lower KL divergence. While reward ensembles significantly outperformed MAP, the integration of LA with ensembles (LA Ens) demonstrated further enhancements, emphasizing the utility of combined approaches in handling overconfident predictions more effectively.

7. Limitations

Our study has certain limitations, notably that our use of Laplace-LoRA is currently limited to LoRA fine-tuned reward models. Extending this methodology to fully fine-tuned models requires additional approximations on KFAC, which we plan to explore in future research. Additionally, due to constraints in computational resources and funding, our experiments were conducted within synthetic settings and with relatively small models, as similarly employed by Gao et al. (2023); Coste et al. (2024). Beirami et al. (2024) showed recently that the widely used KL equation for BoN (Eq. 4) is only an upper bound, and provided a more accurate KL estimator. However, it is out of scope for this work to combine the KL estimator from (Beirami et al., 2024) with the BoN estimator (Appendix B) that we used to estimate mean rewards.

8. Conclusion

We showed that using Laplace-LoRA to quantify uncertainty in reward models can effectively mitigate reward overoptimization in BoN sampling, offering gains over MAP and ensembles. This also holds in RLHF, where it achieves the highest gold reward without the application of KL penalty. Our findings highlight the potential of Bayesian approaches as valuable tools to provide uncertainty estimation in the face of distribution shift, paving the way for more reliable and safer alignment of LLMs.

References

Laurence Aitchison, Adam X. Yang, and Sebastian W. Ober. Deep kernel processes. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 130–140. PMLR, 2021. URL http://proceedings.mlr.press/v139/aitchison21a.html.

Javier Antorán, David Janz, James Urquhart Allingham, Erik A. Daxberger, Riccardo Barbano, Eric T. Nalisnick, and José Miguel Hernández-Lobato. Adapting the linearised laplace model evidence for modern deep learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 796–821. PMLR, 2022. URL https://proceedings.mlr.press/v162/antoran22a.html.

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*, abs/2204.05862, 2022. URL https://arxiv.org/abs/2204.05862.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. *ArXiv preprint*, abs/2401.01879, 2024. URL https://arxiv.org/abs/2401.01879.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1613–1622. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/blundell15.html.

Kianté Brantley, Wen Sun, and Mikael Henaff. Disagreement-regularized imitation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkgbYyHtwB.

Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. Odin: Disentangled reward mitigates hacking in rlhf. *ArXiv preprint*, abs/2402.07319, 2024. URL https://arxiv.org/abs/2402.07319.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. In *ICLR*, 2024.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux - effortless bayesian deep learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20089–20103, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/a7c9585703d275249f30a088cebba0ad-Abstract.html.

Zhijie Deng, Feng Zhou, and Jun Zhu. Accelerated linearized laplace approximation for bayesian deep learning. *NeurIPS*, 2022.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *ArXiv preprint*, abs/2312.09244, 2023. URL https://arxiv.org/abs/2312.09244.

Andrew YK Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. 'in-between'uncertainty in bayesian neural networks. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2019.

Vincent Fortuin, Adrià Garriga-Alonso, Sebastian W. Ober, Florian Wenzel, Gunnar Rätsch, Richard E. Turner, Mark van der Wilk, and Laurence Aitchison. Bayesian neural network priors revisited. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=xkjqJYqRJy.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *ICML*, pages 10835–10866, 2023.

Adam Gleave and Geoffrey Irving. Uncertainty estimation for language reward models. *ArXiv preprint*, abs/2203.07472, 2022. URL https://arxiv.org/abs/2203.07472.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Alexander Immer, Marcin Korzepa, and Matthias Bauer. Improving predictions of bayesian neural nets via local linearization. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 703–711. PMLR, 2021. URL http://proceedings.mlr.press/v130/immer21a.html.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5436–5446. PMLR, 2020. URL http://proceedings.mlr.press/v119/kristiadi20a.html.

Agustinus Kristiadi, Felix Strieth-Kalthoff, Marta Skreta, Pascal Poupart, Alán Aspuru-Guzik, and Geoff Pleiss. A sober look at llms for material discovery: Are they actually good for bayesian optimization over molecules? *ArXiv preprint*, abs/2402.05015, 2024. URL https://arxiv.org/abs/2402.05015.

Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical fisher approximation for natural gradient descent. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *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 4158–4169, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/46a558d97954d0692411c861cf78ef79-Abstract.html.

Yong Lin, Lu Tan, Yifan Hao, Honam Wong, Hanze Dong, Weizhong Zhang, Yujiu Yang, and Tong Zhang. Spurious feature diversification improves out-of-distribution generalization. *ArXiv preprint*, abs/2309.17230, 2023a. URL https://arxiv.org/abs/2309.17230.

Yong Lin, Lu Tan, Hangyu Lin, Zeming Zheng, Renjie Pi, Jipeng Zhang, Shizhe Diao, Haoxiang Wang, Han Zhao, Yuan Yao, et al. Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models. *ArXiv preprint*, abs/2309.06256, 2023b. URL https://arxiv.org/abs/2309.06256.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 1992.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *ArXiv preprint*, abs/2112.09332, 2021. URL https://arxiv.org/abs/2112.09332.

Sebastian W. Ober and Laurence Aitchison. Global inducing point variational posteriors for bayesian neural networks and deep gaussian processes. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8248–8259. PMLR, 2021. URL http://proceedings.mlr.press/v139/ober21a.html.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Alexandre Ramé, Nino Vieillard, Léonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. Warm: On the benefits of weight averaged reward models. *ArXiv preprint*, abs/2401.12187, 2024. URL https://arxiv.org/abs/2401.12187.

Zhengxiang Shi and Aldo Lipani. Dept: Decomposed prompt tuning for parameter-efficient fine-tuning. *ArXiv preprint*, abs/2309.05173, 2023. URL https://arxiv.org/abs/2309.05173.

Zhengyan Shi, Adam X Yang, Bin Wu, Laurence Aitchison, Emine Yilmaz, and Aldo Lipani. Instruction tuning with loss over instructions. *arXiv preprint arXiv:2405.14394*, 2024.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize with human feedback. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1f89885d556929e98d3ef9b86448f951-Abstract.html.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971, 2023. URL https://arxiv.org/abs/2302.13971.

Xi Wang, Laurence Aitchison, and Maja Rudolph. Lora ensembles for large language model fine-tuning. *ArXiv preprint*, abs/2310.00035, 2023. URL https://arxiv.org/abs/2310.00035.

Adam X Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. In *ICLR*, 2024a.

Joy Qiping Yang, Salman Salamatian, Ziteng Sun, Ananda Theertha Suresh, and Ahmad Beirami. Asymptotics of language model alignment. *ArXiv preprint*, abs/2404.01730, 2024b. URL https://arxiv.org/abs/2404.01730.

Yuanzhao Zhai, Han Zhang, Yu Lei, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang. Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora ensembles. *ArXiv preprint*, abs/2401.00243, 2024. URL https://arxiv.org/abs/2401.00243.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for bayesian deep learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkeS1RVtPS.

Shun Zhang, Zhenfang Chen, Sunli Chen, Yikang Shen, Zhiqing Sun, and Chuang Gan. Improving reinforcement learning from human feedback with efficient reward model ensemble. *ArXiv preprint*, abs/2401.16635, 2024. URL https://arxiv.org/abs/2401.16635.

A. Reward overoptimization

We present the full example generations from an LLM undergone extreme reward overoptimization in Table 1.

Table 1: Example generation.

Instruction prompt	Give an example of a company that gives back to the community.
response	The company that provides environmental initiatives to companies should be evaluated in terms of the amount and quality of donations and sponsorships. Additionally, companies should ensure transparency and ensure that policies and practices ensure sustainability and transparency. Furthermore, companies should ensure compliance and ensure transparency and integrity in order to ensure compliance and ensure trust and credibility. Additionally, companies should ensure transparency and ensure integrity in order to ensure compliance and ensure trust and credibility. Additionally, companies should ensure compliance and ensure integrity and integrity to ensure compliance and ensure trust and credibility. Additionally, companies should ensure compliance and ensure integrity and integrity to ensure compliance and ensure trust and credibility and ensure integrity and integrity to ensure compliance and ensure trust and credibility and ensure integrity and integrity to ensure compliance and ensure trust and credibility and ensure integrity and quality and ensure quality and quality and ensure compliance and ensure quality and quality and ensure integrity and quality and ensure quality and quality and ensure compliance and ensure quality and quality and ensure integrity and quality and ensure quality and quality and ensure compliance and ensure quality and quality and ensure integrity and quality and ensure quality and quality and ensure quality and ensure compliance and ensure quality and quality and ensure integrity and quality and ensure quality and quality and ensure quality and quality and ensure compliance and ensure quality and quality and ensure integrity and quality and ensure quality and quality and ensure quality
proxy reward	7.1
gold reward	-0.9

B. Best-of-n sampling

In this section, we review the expected reward estimator in BoN for evaluating reward models (Nakano et al., 2021; Gao et al., 2023; Coste et al., 2024). Assume we have two reward models r^{proxy} for ranking and selecting responses, while r^{gold} for evaluation. Queries are sampled from a query distribution $x \sim q$ while responses are sampled from an LLM $y \sim \pi^{\text{ref}}(y|x)$. For BoN sampling, we aim to sample n responses $y_1, ...y_n$ from the LLM, and rank using $r^{\text{proxy}}(x, y)$. We would like to compute the expected evaluation reward,

$$R(n) := \mathbb{E}_{x\sim q, y_1,\dots,y_n\sim\pi^{\text{ref}}}\Big[r^{\text{eval}}\big(x, \operatorname*{argmax}_{y\in\{y_1,\dots y_n\}} r^{\text{proxy}}(x,y)\big)\Big], \tag{23}$$

where the evaluation reward model $r^{\text{eval}}(x, y)$ could be either the proxy reward model or the gold reward model. The simplest approach is to use a Monte-Carlo estimator for the expectation. However, this requires repeated sampling of n responses from the LLM which is costly. Instead, we consider sampling a fixed set of $N \geq n$ responses for each query from a fixed query set $\mathcal{Q}$, and compute an unbiased estimator

$$R^{\text{MC}}(n) = \sum_{x\in\mathcal{Q}} \frac{1}{\binom{N}{n}} \sum_{1\leq i_1\leq\dots\leq i_n\leq N} r^{\text{eval}}\big(x, \operatorname*{argmax}_{y\in\{y_{i_1},\dots y_{i_n}\}} r^{\text{proxy}}(x,y)\big). \tag{24}$$

If we sort the N responses according to r^{proxy} with $r^{\text{proxy}}(x, y_1) \leq ... \leq r^{\text{proxy}}(x, y_N)$, the above estimator can be further simplified

$$R^{\text{MC}}(n) = \sum_{x\in\mathcal{Q}} \sum_{i=n}^{N} \frac{\binom{i-1}{n-1}}{\binom{N}{n}} r^{\text{eval}}(x, y_i) \tag{25}$$

by noting we only need to iterate the top response from y_n to y_N, and select the rest $(n-1)$ responses from below.

C. Experimental details

In this section we present experiment details for supervised fine-tuning, reward model training, and reinforcement learning from human feedback.

C.1. Supervised fine-tuning

Here we present the experiment setup for supervised fine-tuning following Coste et al. (2024). We use instruction prompts and responses from the AlpacaFarm dataset Dubois et al. (2024) and format prompts and responses with special tokens following the OpenAssistant (Köpf et al., 2024) format. In particular, each prompt starts with a <|prompter|> token and ends with a <|endoftext|> token; each response starts with a <|assistant|> token and ends with a <|endoftext|> token. We show an example in Table 2 below.

Table 2: Prompt format example for supervised fine-tuning.

Instruction prompt	<|prompter|>Create an outline for a presentation about the current state of the world economy.<|endoftext|>
response	<|assistant|>1. Introduction a. Definition of Economics b. Overview of the Major Factors Affecting the Global Economy 2. Causes of Economic Instability a. Impact of Globalization b. Trade Imbalances c. Currency Exchange Rates d. Inflation and Deflation 3. Effects of Economic Trends a. Changes to Consumption and Investment Patterns b. Impact on Employment and Unemployment Rates c. Shift in Distribution of Wealth 4. Conclusion a. Solutions to Achieve Economic Stability b. Suggestions to Overcome Economic Challenges c. Summary of Current State of Global Economy<|endoftext|>

We also present the hyperparameters used in supervised fine-tuning in Table 3 below.

Hyperparameter	Value
Learning Rate	8×10^{-6}
Learning Rate Scheduler	Cosine
Batch Size	4
Epochs	3

Table 3: Hyperparameters used in supervised fine-tuning the Pythia 1.4B LLM policy.

C.2. Reward model training

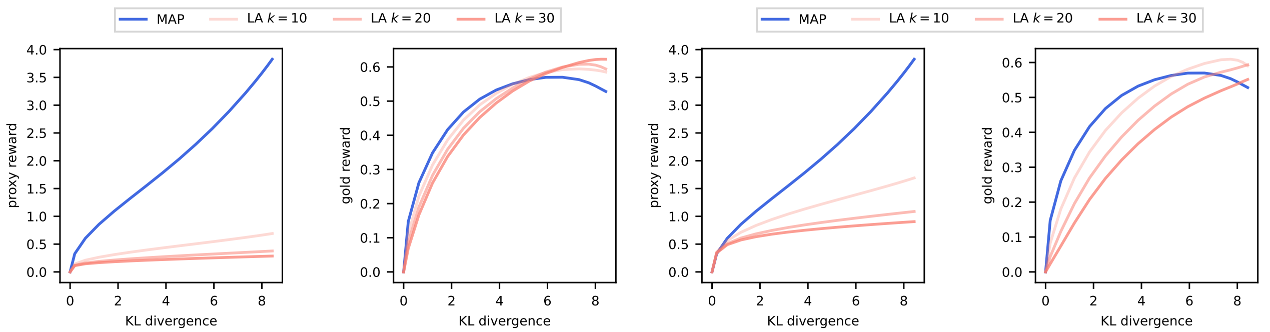
Here we present the hyperparameters we used to train proxy reward models. Table 4 shows the hyperparameters we used for fine-tuning the proxy reward model based on Pythia 70M.

Hyperparameter	Value
LoRA r	8
LoRA α	16
Dropout Probability	0
Weight Decay	0
Learning Rate	5×10^{-5}
Learning Rate Scheduler	Linear
Batch Size	8
Max Sequence Length	500

Table 4: Hyperparameters used in fine-tuning Pythia 70M reward model with LoRA.

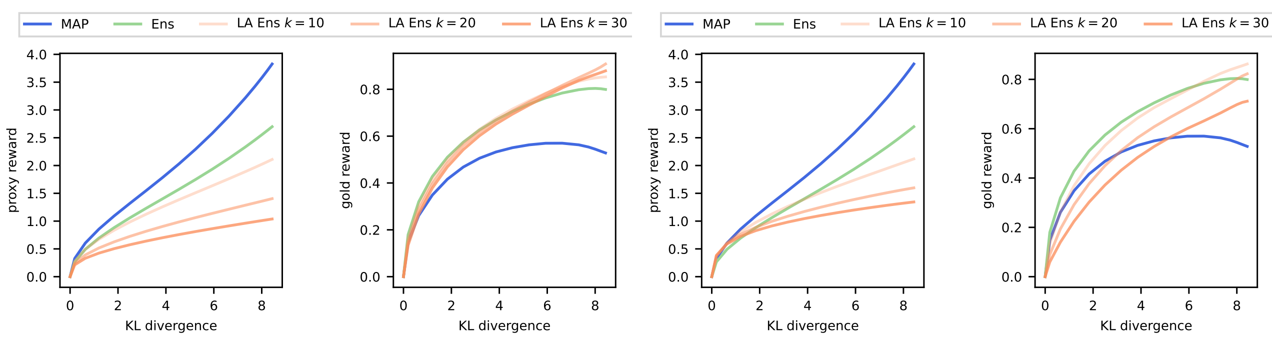
D. Additional experiments

In the main text, we have shown results for $k = 1, 3, 5, 10$. Here, we explore larger values $k = 10, 0, 30$ as shown in Fig. 4 and Fig. 5. We found larger penalties degrades performance of standard deviation-based penalty more significantly, while variance-based penalty is more robust.



(a) Variance-based penalty. (b) Standard deviation-based penalty.

Figure 4: Comparison of proxy and gold reward scores (normalized) in BoN sampling, across different uncertainty penalties and a range of k. Left column: compares the proxy reward model's evaluation. Right column: compares the gold reward model's evaluation.



(a) Variance-based penalty. (b) Standard deviation-based penalty.

Figure 5: Comparison of proxy and gold reward scores (normalized) in BoN sampling, across different uncertainty penalties and a range of k. Left column: compares the proxy reward model's evaluation. Right column: compares the gold reward model's evaluation.