Regularized Best-of-N Sampling with Minimum Bayes Risk Objective for Language Model Alignment

Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, Kenshi Abe
CyberAgent
Tokyo, Japan
{jinnai_yu,morimura_tetsuro,kaito_ariu,abe_kenshi}@cyberagent.co.jp

Abstract

Best-of-N (BoN) sampling with a reward model has been shown to be an effective strategy for aligning Large Language Models (LLMs) to human preferences at the time of decoding. BoN sampling is susceptible to a problem known as *reward hacking* when the accuracy of the reward model is not high enough due to the quality or the quantity of the preference dataset. Because the reward model is an imperfect proxy for the true objective, over-optimizing its value can compromise its performance on the true objective. In this research, we propose MBR-BoN, a variant of BoN that aims to mitigate reward hacking at inference time by incorporating the Minimum Bayes Risk (MBR) objective as a proximity regularization term. We show empirically and analytically that the MBR objective quantifies the proximity of the response to the reference policy, serving as a proximity regularizer. We evaluate MBR-BoN on the AlpacaFarm and Anthropic's hh-rlhf datasets and show that it outperforms both BoN sampling and MBR decoding. We also evaluate MBR-BoN to generate a pairwise preference learning dataset for Direct Preference Optimization (DPO). Empirical results show that models trained on a dataset generated with MBR-BoN outperform those with vanilla BoN. Our code is available at https://github.com/CyberAgentAILab/regularized-bon.

1 Introduction

Language model alignment is a widely used technique for optimizing the behavior of Large Language Models (LLMs) to human preferences, steering the models to generate informative, harmless, and helpful responses (Ziegler et al., 2020; Stiennon et al., 2020; Ouyang et al., 2022). **Best-of-N (BoN) sampling** is widely used to align the LLM at decoding time (Stiennon et al., 2020; Nakano et al., 2022). BoN samples N responses from the language model and selects the best response according to the proxy reward model as the output of the system.

However, BoN sampling is known to suffer from the *reward hacking* problem (Amodei et al., 2016; Ziegler et al., 2020; Stiennon et al., 2020; Skalse et al., 2022; Gao et al., 2023). The reward hacking is a phenomena where the learning agent overfits to the misspecified reward model, failing to optimise for the true intended objective (Pan et al., 2022; Lambert and Calandra, 2024). The problem occurs because of reward misspecification; the proxy reward trained from a human preference dataset of a limited quality or quantity does not perfectly reflect true human preferences. As a result, optimizing for the reward model does not always optimize for the preference of the true intended objective. For example, Dubois et al. (2023) shows that with 25% label noise, which is the amount of disagreement observed in real-world preference annotations (Stiennon et al., 2020; Ouyang et al., 2022), BoN sampling degrades performance with N greater than 16 (Figures 12 and 13 in Dubois et al. 2023). Wen et al. (2024) shows that even when the proxy reward model performs reasonably well relative to the reference model, it still exhibits overoptimization behavior. We also observe the degradation of performance with N greater than 32 when the amount of train data for the proxy reward model is limited (Appendix A).

Given that human preferences depend on the domain, language, culture, and various other factors of the users (Hu et al., 2023; Wan et al., 2023; Li et al., 2024b; Sorensen et al., 2024; Li et al., 2024a; Afzoon et al., 2024; Agrawal et al., 2024), it is desirable to develop a method that is robust to the situation where the reward model is misspecified due to limited quality and/or quality of the preference dataset. A common approach to mitigate reward hacking in preference learning is to add a proximity regularization term to the loss function to keep the trained model close to the reference

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 9321–9347

model (Stiennon et al., 2020; Ouyang et al., 2022; Rafailov et al., 2023). Previous work in BoN has shown that reducing the number of samples N mitigates the reward hacking (Nakano et al., 2022; Pan et al., 2022; Lambert and Calandra, 2024). This approach successfully increases the proximity to the reference policy (Nakano et al., 2022; Beirami et al., 2024) but at the expense of diminished improvement obtained by the method.

To this end, we propose **MBR-BoN**, a method that introduces the Minimum Bayes Risk (MBR) objective (Kumar and Byrne, 2002, 2004; Eikema and Aziz, 2020) as a proximity regularization term into the BoN to mitigate the reward hacking problem.[1] The MBR objective serves as a proximity regularizer by its nature which we show in Section 3. Instead of optimizing the raw reward score, we optimize a sum of the reward score and a regularization term. MBR-BoN can tune the regularization strength by the hyperparameter β, similar to the proximity regularization in RLHF and DPO.

We evaluate the performance of MBR-BoN on the AlpacaFarm (Dubois et al., 2023) and Anthropic's hh-rlhf datasets (Bai et al., 2022) and show that it outperforms the performance of vanilla BoN in a wide range of settings. We also use MBR-BoN to generate a pairwise preference learning dataset and show that a model trained by DPO on a dataset generated with MBR-BoN outperforms a model trained on a dataset generated with vanilla BoN.

2 Background

First, we give an overview of preference learning algorithms including RLHF and DPO. Then we introduce the decoding-time alignment algorithm, BoN sampling.

2.1 Preference Learning

Let $\mathcal{D}$ be a set of instruction, response pair, and preference over response pair: $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}$. RLHF uses the learned reward function to train the language model. Typically, the RL process is formulated as the following optimization problem:

$$\arg\max_{\pi} \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \mathop{\mathbb{E}}_{y \sim \pi(y|x)} [R(x, y)] - \beta \mathbb{D}_{\mathrm{KL}}[\pi(\cdot|x) || \pi_{\mathrm{ref}}(\cdot|x)], \quad (1)$$

[1] MBR-BoN was referred to as $\mathrm{RBoN_{WD}}$ in an earlier version of this manuscript.

where β is a hyperparameter that controls the proximity to the base reference model π_{ref}. The proximity regularization term $\mathbb{D}_{\mathrm{KL}}$ is important to prevent the model from deviating too far from the base model. Since the objective is not differentiable, reinforcement learning algorithms are used for optimization (Schulman et al., 2017; Stiennon et al., 2020; Bai et al., 2022; Ouyang et al., 2022; Zheng et al., 2023b).

DPO trains the language model to align directly with the human preference data over the responses, so it doesn't need a separate reward model (Rafailov et al., 2023). Although DPO is based on supervised learning rather than reinforcement learning, it uses essentially the same loss function under the Bradley-Terry model (Bradley and Terry, 1952). The objective function of the DPO is the following:

$$\arg\max_{\pi} \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma(\beta \log \frac{\pi(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \beta \log \frac{\pi(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)})], \quad (2)$$

where σ is the sigmoid function. Several variants of DPO also use KL-divergence as proximity regularization (Azar et al., 2023; Liu et al., 2024).

Thus, both lines of work in preference optimization have proximity regularization in common to keep the model π close to the reference model π_{ref}.

2.2 Best-of-N (BoN) Sampling

While many methods have been proposed for learning human preferences, a simple, popular, and well-performing method for preference optimization remains Best-of-N (BoN) sampling (Stiennon et al., 2020; Nakano et al., 2022). Let x be an input prompt to the language model π_{ref}. Let Y_{ref} be N responses drawn from $\pi_{\mathrm{ref}}(\cdot|x)$. BoN sampling selects the response with the highest reward score according to the proxy reward model R:

$$y_{\mathrm{BoN}}(x) = \arg\max_{y \in Y_{\mathrm{ref}}} R(x, y). \quad (3)$$

The advantages of BoN over preference learning methods are as follows. First, BoN is simple. It does not require any additional training of the language model. While learning-based alignment methods need to train the LLM, BoN can be applied on the fly. Every time human preferences are updated, learning-based methods must retrain the

LLM to adapt to them. On the other hand, BoN only requires an update of the reward model and does not require the training of the LLM, which is the most expensive process. Second, BoN is an effective strategy in its own right. Several previous works have shown that BoN sampling can outperform learning-based alignment methods (Gao et al., 2023; Eisenstein et al., 2024; Mudgal et al., 2024; Gui et al., 2024). Third, BoN is applicable to a black-box model where fine-tuning is not available. BoN does not require access to the model itself and is applicable using the output sequences from the black-box model. In summary, BoN is a practical and efficient alignment strategy that complements the shortcomings of learning-based strategies and is worthy of investigation.

2.3 Minimum Bayes Risk Decoding

MBR decoding (Kumar and Byrne, 2002, 2004; Eikema and Aziz, 2020; Bertsch et al., 2023) has recently gained attention as an effective decoding strategy in a variety of tasks including machine translation, text summarization, text simplification, and reasoning (Eikema and Aziz, 2020, 2022; Freitag et al., 2022; Suzgun et al., 2023; Bertsch et al., 2023; Heineman et al., 2024; junyou li et al., 2024; Deguchi et al., 2024a).

MBR decoding consists of the following steps. First, it samples N sequences from the model (Y_{ref}), similar to BoN sampling. Then, it computes the utility U (e.g., similarity) between each pair of sequences in Y_{ref}. Finally, it selects the sequence that maximizes the average utility between the rest of the sequences:

$$y_{\mathrm{MBR}}(x) = \underset{y \in Y_{\mathrm{ref}}}{\arg\max} \sum_{y' \in Y_{\mathrm{ref}}} \frac{1}{N} U(y, y'), \quad (4)$$

where the summation represents the Bayes risk, which we refer to as **the MBR objective** in this work. MBR decoding is based on the concept of Bayes risk minimization which originates from the decision theoretic framework (Goel and Byrne 2000; Bickel and Doksum 2015, p.27-28). Instead of selecting the output with the highest probability (maximum-a-posteriori decoding; Stahlberg and Byrne 2019; Holtzman et al. 2020), Bayes risk minimization selects the output that is robust to the inaccuracy of the probability model (Meister et al., 2022; Eikema, 2024). Bayes risk minimization is instead formalized as expected utility maximization as utility functions are more common in text generation tasks.

An alternative view of the MBR decoding is that it selects the most centered point (medoid; Kaufman and Rousseeuw 1987) in Y_{ref} where the utility function U measures the similarity between the data points (Jinnai and Ariu, 2024). In other words, the MBR objective quantifies the proximity of the data point to the rest of the samples.

3 Minimum Bayes Risk Objective is a Proximity Regularizer

Although BoN sampling is shown to be effective with a decent reward model, it is prone to the reward hacking problem under less accurate reward models (Dubois et al., 2023; Wen et al., 2024). A naive approach to prevent reward hacking is to introduce a proximity regularizer to the BoN sampling in the form of a KL-divergence term, as is common in preference learning methods (Stiennon et al., 2020; Ouyang et al., 2022; Rafailov et al., 2023). However, we observe that this strategy does not improve over BoN in most cases (Appendix D).

To this end, we propose to use the MBR objective as a proximity regularizer. First, in Section 3.1, we visually show that the MBR objective is correlated with the semantic proximity of the reference policy. Then, we show an analytical result in Section 3.2 that the MBR objective corresponds to the Wasserstein distance (Peyré and Cuturi, 2020; Villani, 2021b), indicating that the MBR objective by its nature quantifies the proximity of the text to the reference policy.

3.1 Empirical Evaluation

We evaluate the effect of the MBR objective as a proximity regularizer to keep the output closer to the center of the sample distribution. In particular, we evaluate the correlation between the MBR objective and the closeness to the center of the sample distribution. We run an experiment using the first 1000 entries of the training split of the AlpacaFarm (Dubois et al., 2023) and Anthropic's hh-rlhf (Bai et al., 2022) datasets. $N = 128$ responses are sampled from `mistral-7b-sft-beta` (Mistral) for each instruction (Jiang et al., 2023a; Tunstall et al., 2024). The MBR objective (Eq 4) is calculated for each sample, and normalized to the range of [0, 1]. We use a cosine similarity of the embedding computed with `all-mpnet-base-v2` (MPNet; Reimers and Gurevych 2019; Song et al. 2020):

$$U(y, y') = \cos(\mathrm{emb}(y), \mathrm{emb}(y')), \quad (5)$$

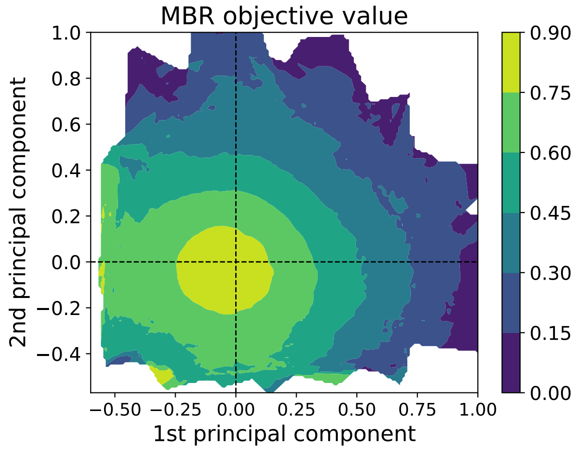


Figure 1: Mapping of the average MBR objective values to the first and the second principal components using PCA. The figure illustrates that the value of the MBR objective tends to get smaller as it moves away from the center of the distribution in the space of the principal components.

Table 1: Correlation of the distance to the center point in the component space with the MBR objective (Eq 4) on the AlpacaFarm dataset. The mean and standard deviation of the correlation are shown in the table. The result shows that the more an output y deviates from the center of the distribution, the lower the value of the MBR objective. Dim is the number of components.

Dim	PCA	ICA
2	-0.5747 $\pm$ 0.1858	-0.5621 $\pm$ 0.1830
5	-0.7494 $\pm$ 0.1329	-0.6683 $\pm$ 0.1291
10	-0.8512 $\pm$ 0.1010	-0.6809 $\pm$ 0.1222

where emb denotes the embedding function. We then compute the components of the text embedding using Principal Component Analysis (PCA; Pearson 1901) and Independent Component Analysis (ICA; Comon 1994). Since the utility matrix between samples is likely to be approximated by a low-rank matrix (Trabelsi et al., 2024), the first few components are likely to be sufficient to illustrate the proximity between samples in the utility space. We interpolate the values in component space for each instruction and then compute the average over the instructions of the dataset.

Figure 1 shows the mapping of the average MBR objective values, with the horizontal and vertical axes showing the first and second principal components of the embeddings. Table 1 shows the correlation of the distance to the center in principal component space with the MBR objective. Regardless of the dimension of the components, we observe qualitatively the same result that the correlation of the distance from the center with the MBR objective is strongly negative. On the other hand, the correlation with the log probability of the output is weak, indicating that the KL-divergence based on probability may not be a reliable measure of proximity in the embedding space (Table 6 in Appendix B). The result shows that the MBR objective value becomes smaller as it moves away from the center of the distribution. We observe the same qualitative results in Anthropic's hh-rlhf and in a machine translation dataset (WMT'21 De-En; Akhbardeh et al. 2021) which we show in Appendix B.

3.2 Analytical Evaluation

Formally, the MBR objective corresponds to selecting the output y that minimizes the Wasserstein Distance (WD; Peyré and Cuturi 2020; Villani 2021a) to the sample distribution. WD, also known as the Earth Mover's Distance (EMD; Rubner et al. 1998), measures the cost required to transform one probability distribution into another. The cost function C typically represents the "distance" or "effort" required to move a unit of probability mass from one location to another. In the context of NLP, it is also called the Word Mover's Distance to evaluate the similarity between a pair of texts (Kusner et al., 2015; Huang et al., 2016). For a pair of probability distributions P and Q over Y_{ref}, WD is defined as follows:

$$WD(P, Q) = \min_{\{\mu_{i,j}\}_{i,j} \in \mathcal{J}(P,Q)} \sum_{i=1}^{|Y_{\mathrm{ref}}|} \sum_{j=1}^{|Y_{\mathrm{ref}}|} \mu_{i,j} C(y_i, y_j), \quad (6)$$

where C is the cost function that represents the dissimilarity of the elements. $\mathcal{J}(P, Q)$ is a set of all couplings over P and Q (Villani, 2021a):

$$\mathcal{J}(P,Q) = \{\{\mu_{i,j}\}_{i,j} : \sum_{i=1}^{|Y_{\mathrm{ref}}|} \mu_{i,j} = Q(y_j), \sum_{j=1}^{|Y_{\mathrm{ref}}|} \mu_{i,j} = P(y_i), \mu_{i,j} \geq 0\}. \quad (7)$$

The objective of MBR decoding is identical to minimizing the WD to the empirical distribution of π_{ref}.

Proposition 1. *Let the cost function C for WD be $C(y, y') = -U(y, y')$ for all y and y'. MBR decoding selects the output with the smallest WD* *of the sample distribution:*

$$y_{\mathrm{MBR}}(x) = \arg\max_{y \in Y_{\mathrm{ref}}} \sum_{y' \in Y_{\mathrm{ref}}} \frac{1}{N} U(y, y') \tag{8}$$

$$= \arg\min_{y \in Y_{\mathrm{ref}}} WD(\pi_y(\cdot \mid x), \hat{\pi}_{\mathrm{ref}}(\cdot \mid x)), \tag{9}$$

where π_y *is a policy that outputs* y *with a probability of 1 and* $\hat{\pi}_{\mathrm{ref}}$ *is the empirical distribution constructed from* Y_{ref}: $\hat{\pi}_{\mathrm{ref}}(y \mid x) = \frac{1}{N} \sum_{y_i \in Y_{\mathrm{ref}}} \mathbb{I}[y_i = y]$.

Proof. The proof is in Appendix C. □

The proposition shows that the MBR objective measures the WD of the output selection strategy to the sample distribution of the reference policy. Maximizing the objective results in selecting an output that is closest to the sample distribution of the reference policy with respect to the utility function U.

Summary. Both the empirical and analytical results show that the MBR objective serves as a proximity regularizer to penalize an output that is less representative of the samples from the reference policy, as measured by the utility function.

4 MBR-Best-of-N (MBR-BoN) Sampling

We propose **MBR-Best-of-N (MBR-BoN) sampling**, a variant of BoN sampling with an MBR objective as the proximity regularizer, to mitigate the reward hacking problem of BoN sampling. MBR-BoN uses the MBR objective as the proximity regularizer:

$$y_{\text{MBR-BoN}}(x) = \arg\max_{y \in Y_{\mathrm{ref}}} R(x, y) + \beta \sum_{y' \in Y_{\mathrm{ref}}} \frac{1}{N} U(y, y'), \tag{10}$$

where β is a hyperparameter to adjust the strength of the regularization. As the MBR objective corresponds to the WD between the resulting policy and the reference policy (Section 3), it serves as a proximity regularizer to ensure that the resulting policy is close to the reference policy π_{ref}.

The hyperparameter β controls the tradeoff between the reward and proximity to the reference model. Using a small β makes the output more aligned with the proxy reward, with $\beta = 0$ recovering vanilla BoN sampling. A larger β makes the output closer to the behavior of the reference model π_{ref}, with $\beta = +\infty$ recovering MBR decoding.

Advantage of WD over KL-divergence. WD is a more suited regularizer than KL-divergence for inference-time algorithms where the number of samples is very small. While KL-divergence is useful for training-time alignment algorithms, it poses several challenges for inference-time algorithms with limited samples.

Theoretically, any high confidence lower bound on KL-divergence requires a sample size exponential in the value of KL-divergence (McAllester and Stratos, 2020). This suggests that estimating KL-divergence is unreliable in finite-sample settings. For example, for the first instance of the AlpacaEval instruction (*What are the names of some famous actors that started their careers on Broadway?*), the KL-divergence of the randomly sampled 128 responses from Mistral has a minimum of 627, a maximum of 5870, a mean of 1854, and a standard deviation of 1039.

Moreover, KL-divergence is sensitive to small differences in the sequences. Specifically, KL-divergence can be large even if the underlying sequences differ very little. For example, the two sentences: "*Yes I will do it.*" and "*Yes I'll do it.*" are considered completely different data instances when computing KL-divergence. Conversely, WD considers them to be quite similar data instances. This is because the WD uses the utility function to quantify the divergence and represents the difference between two distributions in terms of the semantic distance between the sequences. This makes the WD a more robust measure against the minor variances that naturally occur in natural language texts. See Appendix D for experimental evaluation of using KL-divergence as a regularization term.

In addition to being a good proximal regularizer, the MBR objective is a useful text generation objective in its own right. The objective is shown to be effective, outperforming MAP decoding in a variety of text generation tasks, including instruction-following task (Suzgun et al., 2023; Bertsch et al., 2023; junyou li et al., 2024).

5 Experiments

We evaluate the performance of MBR-BoN for two use cases. First, we evaluate the performance of MBR-BoN for decoding time alignment (Section 5.1). Then, we evaluate MBR-BoN as a sampling strategy to generate a preference learning dataset to be used for DPO (Section 5.2).

Table 2: Average Spearman's rank correlation coefficient of the proxy reward models to the gold reference reward model (Eurus) on AlpacaFarm.

Proxy reward	Correlation Coefficient
SHP-Large	0.32
SHP-XL	0.39
OASST	0.40

Table 3: The values of hyperparameter β used by MBR-BoN determined using the development set.

Dataset	SHP-Large	SHP-XL	OASST
AlpacaFarm	0.5	0.5	20.0
Helpfulness	0.05	0.1	20.0
Harmlessness	2.0	2.0	20.0

5.1 MBR-BoN for Decoding-Time Alignment

Setup. The evaluation is conducted using the AlpacaFarm (Dubois et al., 2023) and Anthropic's hh-rlhf datasets (Bai et al., 2022). For the AlpacaFarm dataset, we use the first 1000 entries of the train split (`alpaca_human_preference`) as the development set and the whole evaluation split (`alpaca_farm_evaluation`) (805 instructions) as a test dataset. For Anthropic's datasets, we conduct experiments on the `helpful-base` (Helpfulness) and `harmless-base` (Harmlessness) subsets separately. For each subset, we use the first 1000 entries of the train split as the development set and the first 1000 entries of the test split as a test dataset. We use `mistral-7b-sft-beta` (Mistral) and `dolly-v2-3b` (Dolly) as the language models (Jiang et al., 2023a; Tunstall et al., 2024; Conover et al., 2023).

To evaluate MBR-BoN under various conditions, we use SHP-Large, SHP-XL (Ethayarajh et al., 2022), and OASST (Köpf et al., 2023) as proxy reward models. We use Eurus as a gold reference reward model as it is one of the most accurate reward models (Lambert et al., 2024; Zhou et al., 2024) and is open-source which makes the experiments reproducible. The results using other reward models as a gold reference are reported in Appendix E and G. The average Spearman's rank correlation coefficient ρ (Spearman, 1904) to the gold reference reward (Eurus) is reported in Table 2.

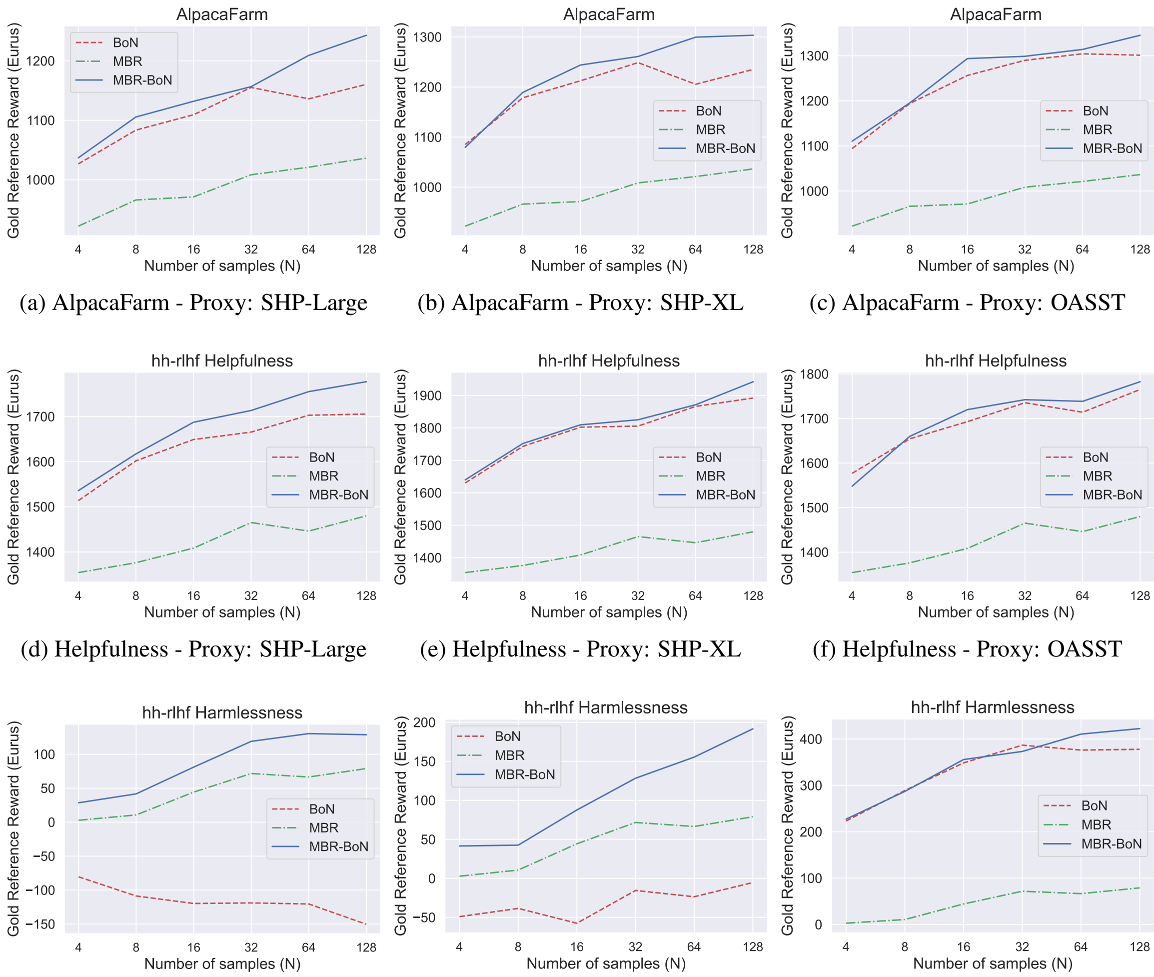
We compare the performance of BoN, MBR, and MBR-BoN. We sample up to $N = 128$ responses per instruction using nucleus sampling and select the output using the algorithms. We set the top-p to be $p = 0.9$ and the temperature to be $T = 1.0$ for the nucleus sampling (Holtzman et al., 2020). For a fair comparison, we use the same set of N responses for all algorithms. We use the Sentence BERT model (Reimers and Gurevych, 2019) based on MPNet (Song et al., 2020) to compute the sentence embedding for MBR and MBR-BoN.

MBR-BoN use the development set to select the optimal β. For each pair of a proxy reward and a gold reference reward, we run MBR-BoN with $\beta \in \{10^{-6}, 2\cdot10^{-6}, 5\cdot10^{-6}, 10^{-5}, ..., 2\cdot10^{1}\}$ and pick the best performing β for $N = 128$. We use the same β for all N in evaluation. See Appendix G and F for the ablation study on the regularization strength β.

Results. Figure 2 shows the performance of BoN, MBR, and MBR-BoN using Mistral as a language model, evaluated by Eurus score. See Appendix G for the result of Dolly. Overall, MBR-BoN outperforms BoN and MBR in most of the settings, showing that the method is effective in a wide range of tasks. Figure 3 shows the performance of MBR-BoN with $N = 128$ and with varying regularization strength β. The vertical line shows the β selected using the development set. Overall, MBR-BoN outperforms BoN in a wide range of β and is relatively robust to the choice of β.

As expected, we observe that MBR-BoN have lower scores with respect to the proxy reward than BoN (Appendix G). The regularization term effectively mitigates the reward hacking of the BoN, resulting in a higher score in the gold reference score (Eurus).

Choice of Regularization strength. Table 3 summarizes the regularization strength β picked using the development set. The optimal value of β depends on the choice of the language model, dataset, and proxy reward model, which requires the use of the development set to tune the hyperparameter β. Still, we find that the amount of development data we need for hyperparameter tuning is small. Our post-hoc analysis on the size of the developement set shows that with as little as 10 instances it already outperforms BoN and also finds β close to the optimal β (Appendix F). Also note that the computational cost of tuning the hyperparameter for MBR-BoN is marginal compared to that of RLHF or DPO as it does not involve any training of the



(a) AlpacaFarm - Proxy: SHP-Large (b) AlpacaFarm - Proxy: SHP-XL (c) AlpacaFarm - Proxy: OASST

(d) Helpfulness - Proxy: SHP-Large (e) Helpfulness - Proxy: SHP-XL (f) Helpfulness - Proxy: OASST

(g) Harmlessness - Proxy: SHP-Large (h) Harmlessness - Proxy: SHP-XL (i) Harmlessness - Proxy: OASST

Figure 2: Evaluation BoN, MBR, and MBR-BoN on the AlpacaFarm, hh-rlhf Helpfullness, and hh-rlhf Harmlessness datasets. Mistral is used as the language model.

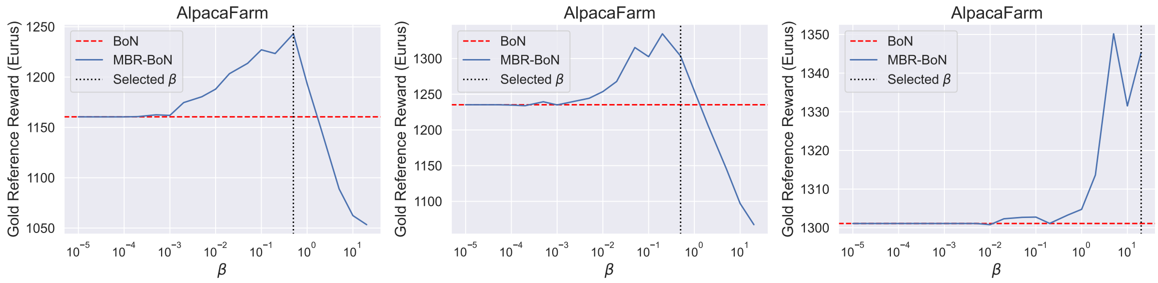
language model or reward model. Running MBR-BoN with different β only requires the computation of Eq. 10 with the different β.

5.2 MBR-BoN for Generating Preference Learning Dataset

Previous work has shown that BoN sampling is an effective strategy for generating an efficient preference dataset (Xu et al., 2023; Yuan et al., 2024b; Pace et al., 2024). They show that the efficiency of pairwise preference learning is improved by using the best and worst responses according to the reward model as the chosen and rejected responses. We evaluate the performance of DPO (Rafailov et al., 2023) using the response selected by MBR-BoN as the chosen response and the response with the lowest reward score as the rejected response.

Setup. We sample 128 responses for each instruction in the training dataset and use the response selected by MBR-BoN or BoN as the chosen response and the response with the lowest reward as the rejected response. We use all 9.69k instructions from AlpacaFarm and the first 5k instructions from each of the Helpfulness and Harmlessness subsets to train a model for the hh-rlhf datasets. We use Mistral as the language model to generate the pairwise preference dataset and train it using the generated dataset (Jiang et al., 2023a; Tunstall et al., 2024).

OASST is used as a proxy reward model and Eurus is used for evaluation (Köpf et al., 2023; Yuan et al., 2024a). We train a model with DPO using Low-Rank Adaptation (LoRA; Hu et al. 2022; Sidahmed et al. 2024). The trained models are evaluated using the evaluation split of the AlpacaFarm dataset. Other hyperparameters are described in

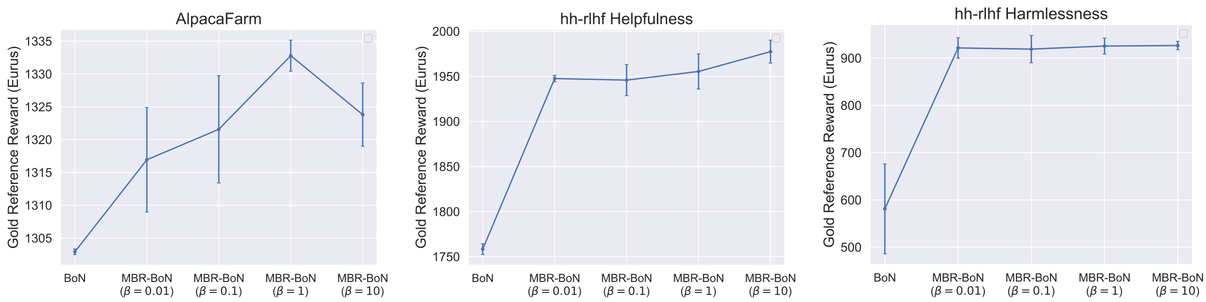


(a) Proxy: SHP-Large

(b) Proxy: SHP-XL

(c) Proxy: OASST

Figure 3: Evaluation of the MBR-BoN using Mistral on the AlpacaFarm dataset with varying regularization strength β. The number of samples is $N = 128$.



(a) Alpaca

(b) Helpfulness

(c) Harmlessness

Figure 4: Evaluation of the DPO using MBR-BoN to generate the preference dataset. OASST is used as the proxy reward model to generate the preference dataset, and Eurus is used as the gold reference reward. The line represents the mean of three runs, and the error bar shows the standard error of the mean.

Appendix J.

Results. Figure 4 shows the performance of models trained using MBR-BoN and BoN to generate a pairwise preference dataset. The models trained with MBR-BoN outperform a model trained with BoN. According to Figure 2, MBR-BoN generates higher quality response texts than BoN with respect to the gold reference reward. We expect the models trained by DPO on the higher quality responses to achieve higher quality generation. In addition, MBR-BoN generates on-policy responses that are representative of the reference policy (Section 3.1 and Appendix B) which is shown to be one of the important characteristics of efficient preference datasets (Chang et al., 2024; Guo et al., 2024; Xu et al., 2024b; Tajwar et al., 2024; Tang et al., 2024). Thus, we postulate that by generating high-quality and on-policy responses, models aligned with responses generated by MBR-BoN outperform that of BoN.

We additionally evaluate the performance of DPO with responses generated by random sampling (i.e., BoN with $N = 2$). According to the Eurus reward model, the scores were as follows: 1140.0 for Alpaca, 1556.9 for Helpfulness, and 433.3 for Harmlessness. Both BoN and MBR-RoN significantly outperform random sampling. This result aligns with prior work showing BoN outperforming random sampling (Xu et al., 2023; Yuan et al., 2024b; Pace et al., 2024).

The result shows the potential of MBR-BoN as a tool for generating pairwise preference datasets for preference learning. See Appendix H for the results using GPT-4o as an evaluator.

6 Related Work

Mitigating reward hacking at inference time. Using proximity regularization is not the only way to mitigate the reward hacking problem. Several studies have explored the use of multiple rewards. (Mudgal et al., 2023; Coste et al., 2024; Rame et al., 2024) propose to ensemble multiple reward functions to mitigate reward hacking. Several studies have investigated training models by the reward functions and combining by interpolating the parameters (Ramé et al., 2023; Jang et al., 2023) or ensembling the model (Mitchell et al., 2024; Shi et al., 2024). Our approach is applicable to any proxy reward model, so it can be combined with these methods.

MBR for training a model. Prior work has discovered that MBR decoding for LLM is useful for inference and for generating preference dataset in machine translation tasks (Farinhas et al., 2023; Ramos et al., 2024). Finkelstein and Freitag (2024); Guttmann et al. (2024) uses the output generated by MBR decoding for supervised fine-tuning to improve the generation quality of a machine translation model. Yang et al. (2024) trains a model by DPO to prefer outputs with higher MBR objective values than lower ones. Tomani et al. (2024) trains the machine translation model to predict the quality of the generation so that it can improve its own generation using the estimate. The novelty of our work is to introduce the MBR objective combined with BoN sampling for language model alignment, improving both the text generation and the training using the generated texts.

7 Conclusions

We propose MBR-BoN, a variant of BoN sampling with MBR objective as a proximity regularizer to mitigate the reward hacking problem. We show that the MBR objective is a proximity regularizer by its nature and show it in the experiments. We evaluate the performance of MBR-BoN using the AlpacaFarm and Anthropic's hh-rlhf datasets. The result shows that MBR-BoN outperforms BoN when the proxy reward is weakly correlated with the reference objective. As an application of the method, we also show that MBR-BoN is an effective strategy for generating a preference dataset for DPO.

We believe that MBR-BoN will be a practical choice for future decoding-time alignment methods because of its applicability and performance improvements.

8 Limitations

The drawback of the proposed method is that it requires a development set to tune the hyperparameter. Given that there is no clear strategy to pick the β parameter even for RLHF and DPO, we speculate that it would be challenging to develop a strategy to find an effective β automatically. Still, the hyperparameter tuning of MBR-BoN is much more computationally efficient than that of RLHF and DPO as it does not involve any training procedures. In fact we observe that around 10 instances are enough to find a near-optimal choice of β (Appendix F).

One of the critical limitations of MBR decoding is its generation speed. It requires computing a utility function that is quadratic to the number of samples. MBR-BoN inherits the same limitation because it is derived from MBR. Given that recent work (Cheng and Vlachos, 2023; Jinnai and Ariu, 2024; Deguchi et al., 2024b; Vamvas and Sennrich, 2024) has improved the computational complexity of MBR decoding to linear in the number of samples, we are optimistic that the overhead of MBR-BoN will be reduced in the future.

We use automated evaluation metrics to evaluate the models. Although we use one of the most accurate publicly available reward models and GPT-4o to evaluate the performance of the models (Yuan et al., 2024a; Lambert et al., 2024), it would be desirable to perform a human evaluation.

Our experiments on preference learning are limited to the evaluation of DPO. Evaluation of MBR-BoN for other preference optimization algorithms is future work (Azar et al., 2023; Liu et al., 2024; Ethayarajh et al., 2024; Xu et al., 2024a; Morimura et al., 2024; Hong et al., 2024; Meng et al., 2024; Park et al., 2024).

9 Impact Statement

We believe that this work will have a positive impact by providing a method for fine-tuning an LLM with limited annotation resources, allowing for alignment with less representative communities in language resources. LLMs would be more useful if we could prevent them from reward hacking, even when the annotation for the task is limited.

Acknowledgments

We thank the anonymous reviewers for their insightful comments and suggestions. Kaito Ariu's research is supported by JSPS KAKENHI Grant No. 23K19986.

References

Saleh Afzoon, Usman Naseem, Amin Beheshti, and Zahra Jamali. 2024. Persobench: Benchmarking personalized response generation in large language models. *arXiv preprint arXiv:2410.03198*.

Sweta Agrawal, José G. C. De Souza, Ricardo Rei, António Farinhas, Gonçalo Faria, Patrick Fernandes, Nuno M Guerreiro, and Andre Martins. 2024. Modeling user preferences with automatic metrics: Creating a high-quality preference dataset for machine translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14503–14519, Miami, Florida, USA. Association for Computational Linguistics.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*.

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

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

Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew Gormley. 2023. It's MBR all the way down: Modern generation techniques through the lens of minimum Bayes risk. In *Proceedings of the Big Picture Workshop*, pages 108–122, Singapore. Association for Computational Linguistics.

Peter J Bickel and Kjell A Doksum. 2015. *Mathematical statistics: basic ideas and selected topics, volumes I-II package*. Chapman and Hall/CRC.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

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

Jonathan D. Chang, Wenhao Zhan, Owen Oertell, Kianté Brantley, Dipendra Misra, Jason D. Lee, and Wen Sun. 2024. Dataset reset policy optimization for RLHF. *arXiv preprint arXiv:2404.08495*.

Julius Cheng and Andreas Vlachos. 2023. Faster minimum Bayes risk decoding with confidence-based pruning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12473–12480, Singapore. Association for Computational Linguistics.

Pierre Comon. 1994. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2024. Reward model ensembles help mitigate overoptimization. In *The Twelfth International Conference on Learning Representations*.

Hiroyuki Deguchi, Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2024a. mbrs: A library for minimum Bayes risk decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 351–362, Miami, Florida, USA. Association for Computational Linguistics.

Hiroyuki Deguchi, Yusuke Sakai, Hidetaka Kamigaito, Taro Watanabe, Hideki Tanaka, and Masao Utiyama. 2024b. Centroid-based efficient minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11009–11018, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 30039–30069. Curran Associates, Inc.

Bryan Eikema. 2024. The effect of generalisation on the inadequacy of the mode. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 87–92, St Julians, Malta. Association for Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2022. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10978–10993, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alexander Nicholas D'Amour, Krishnamurthy Dj Dvijotham, Adam Fisch, Katherine A Heller, Stephen Robert Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. 2024. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. In *First Conference on Language Modeling*.

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

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

António Farinhas, José de Souza, and Andre Martins. 2023. An empirical study of translation hypothesis ensembling with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11956–11970, Singapore. Association for Computational Linguistics.

Mara Finkelstein and Markus Freitag. 2024. MBR and QE finetuning: Training-time distillation of the best and most expensive decoding methods. In *The Twelfth International Conference on Learning Representations*.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR.

Vaibhava Goel and William J Byrne. 2000. Minimum bayes-risk automatic speech recognition. *Computer Speech & Language*, 14(2):115–135.

Lin Gui, Cristina Garbacea, and Victor Veitch. 2024. BoNBon alignment for large language models and the sweetness of best-of-n sampling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, Johan Ferret, and Mathieu Blondel. 2024. Direct language model alignment from online AI feedback. *arXiv preprint arXiv:2402.04792*.

Kamil Guttmann, Mikołaj Pokrywka, Adrian Charkiewicz, and Artur Nowakowski. 2024. Chasing COMET: Leveraging minimum Bayes risk decoding for self-improving machine translation. In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 80–99, Sheffield, UK. European Association for Machine Translation (EAMT).

David Heineman, Yao Dou, and Wei Xu. 2024. Improving minimum Bayes risk decoding with multi-prompt. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22525–22545, Miami, Florida, USA. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, Florida, USA. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yebowen Hu, Kaiqiang Song, Sangwoo Cho, Xiaoyang Wang, Hassan Foroosh, and Fei Liu. 2023. DecipherPref: Analyzing influential factors in human preference judgments via GPT-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8344–8357, Singapore. Association for Computational Linguistics.

Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha, and Kilian Q Weinberger. 2016. Supervised word mover's distance. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *arXiv*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023b. Llm-blender: Ensembling large language models with pairwise comparison and generative fusion. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL 2023)*.

Yuu Jinnai and Kaito Ariu. 2024. Hyperparameter-free approach for faster minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 8547–8566, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

junyou li, Qin Zhang, Yangbin Yu, QIANG FU, and Deheng Ye. 2024. More agents is all you need. *Transactions on Machine Learning Research*.

Leonard Kaufman and Peter J. Rousseeuw. 1987. Clustering by means of medoids. In *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland*, volume 31.

Shankar Kumar and William Byrne. 2002. Minimum Bayes-risk word alignments of bilingual texts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 140–147. Association for Computational Linguistics.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations – democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Nathan Lambert and Roberto Calandra. 2024. The alignment ceiling: Objective mismatch in reinforcement learning from human feedback. *arXiv preprint arXiv:2311.00168*.

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

Bryan Li, Samar Haider, and Chris Callison-Burch. 2024a. This land is Your, My land: Evaluating geopolitical bias in language models through territorial disputes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3855–3871, Mexico City, Mexico. Association for Computational Linguistics.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. 2024b. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*.

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, Peter J. Liu, and Xuanhui Wang. 2024. LiPO: Listwise preference optimization through learning-to-rank. *arXiv preprint arXiv:2402.01878*.

David McAllester and Karl Stratos. 2020. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884. PMLR.

Clara Meister, Gian Wiher, Tiago Pimentel, and Ryan Cotterell. 2022. On the probability–quality paradox in language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 36–45, Dublin, Ireland. Association for Computational Linguistics.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple preference optimization with a reference-free reward. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D Manning. 2024. An emulator for fine-tuning large language models using small language models. In *The Twelfth International Conference on Learning Representations*.

Tetsuro Morimura, Mitsuki Sakamoto, Yuu Jinnai, Kenshi Abe, and Kaito Ariu. 2024. Filtered direct preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22729–22770, Miami, Florida, USA. Association for Computational Linguistics.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Jilin Chen, Alex Beutel, and Ahmad Beirami. 2023. Controlled decoding from language models. In *Socially Responsible Language Modelling Research*.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. 2024. Controlled decoding from language models. In *Forty-first International Conference on Machine Learning*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret

Zoph. 2024. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Alizée Pace, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2024. West-of-n: Synthetic preference generation for improved reward modeling. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*.

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

Karl Pearson. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Gabriel Peyré and Marco Cuturi. 2020. Computational optimal transport. *arXiv preprint arXiv:1803.00567*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Alexandre Ramé, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2023. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *Advances in neural information processing systems*.

Alexandre Rame, Nino Vieillard, Leonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. 2024. WARM: On the benefits of weight averaged reward models. In *Forty-first International Conference on Machine Learning*.

Miguel Ramos, Patrick Fernandes, António Farinhas, and Andre Martins. 2024. Aligning neural machine translation models: Human feedback in training and inference. In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 258–274, Sheffield, UK. European Association for Machine Translation (EAMT).

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 1998. A metric for distributions with applications to image databases. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 59–66. IEEE.

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

Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hannaneh Hajishirzi, Noah A. Smith, and Simon Shaolei Du. 2024. Decoding-time language model alignment with multiple objectives. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*.

Hakim Sidahmed, Samrat Phatale, Alex Hutcheson, Zhuonan Lin, Zhang Chen, Zac Yu, Jarvis Jin, Roman Komarytsia, Christiane Ahlheim, Yonghao Zhu, Simral Chaudhary, Bowen Li, Saravanan Ganesh, Bill Byrne, Jessica Hoffmann, Hassan Mansoor, Wei Li, Abhinav Rastogi, and Lucas Dixon. 2024. Perl: Parameter efficient reinforcement learning from human feedback. *arXiv preprint arXiv:2403.10704*.

Joar Max Viktor Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Taylor Sorensen, Liwei Jiang, Jena D. Hwang, Sydney Levine, Valentina Pyatkin, Peter West, Nouha Dziri, Ximing Lu, Kavel Rao, Chandra Bhagavatula, Maarten Sap, John Tasioulas, and Yejin Choi. 2024. Value kaleidoscope: Engaging AI with pluralistic human values, rights, and duties. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(18):19937–19947.

Charles Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1).

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2023. Follow the wisdom of the crowd: Effective text generation via minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4265–4293, Toronto, Canada. Association for Computational Linguistics.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. 2024. Preference fine-tuning of LLMs should leverage suboptimal, on-policy data. In *Forty-first International Conference on Machine Learning*.

Yunhao Tang, Daniel Zhaohan Guo, Zeyu Zheng, Daniele Calandriello, Yuan Cao, Eugene Tarassov, Rémi Munos, Bernardo Ávila Pires, Michal Valko, Yong Cheng, and Will Dabney. 2024. Understanding the performance gap between online and offline alignment algorithms. *arXiv preprint arXiv:2405.08448*.

Christian Tomani, David Vilar, Markus Freitag, Colin Cherry, Subhajit Naskar, Mara Finkelstein, Xavier Garcia, and Daniel Cremers. 2024. Quality-aware translation models: Efficient generation and quality estimation in a single model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15660–15679, Bangkok, Thailand. Association for Computational Linguistics.

Firas Trabelsi, David Vilar, Mara Finkelstein, and Markus Freitag. 2024. Efficient minimum bayes risk decoding using low-rank matrix completion algorithms. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook AI's WMT21 news translation task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 205–215, Online. Association for Computational Linguistics.

Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. 2024. Zephyr: Direct distillation of LM alignment. In *First Conference on Language Modeling*.

Jannis Vamvas and Rico Sennrich. 2024. Linear-time minimum Bayes risk decoding with reference aggregation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 790–801, Bangkok, Thailand. Association for Computational Linguistics.

Cédric Villani. 2021a. *Topics in optimal transportation*, volume 58. American Mathematical Soc.

Rossana Villani. 2021b. The changing profile of the translator profession at the European central bank. In *Proceedings of the Translation and Interpreting Technology Online Conference*, pages 7–14, Held Online. INCOMA Ltd.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Ruyuan Wan, Jaehyung Kim, and Dongyeop Kang. 2023. Everyone's voice matters: Quantifying annotation disagreement using demographic information. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12):14523–14530.

Xueru Wen, Jie Lou, Yaojie Lu, Hongyu Lin, Xing Yu, Xinyu Lu, Ben He, Xianpei Han, Debing Zhang, and Le Sun. 2024. Rethinking reward model evaluation: Are we barking up the wrong tree? *arXiv preprint arXiv:2410.05584*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024a. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *Forty-first International Conference on Machine Learning*.

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. 2023. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024b. Is DPO superior to PPO for LLM alignment? a comprehensive study. In *Forty-first International Conference on Machine Learning*.

Guangyu Yang, Jinghong Chen, Weizhe Lin, and Bill Byrne. 2024. Direct preference optimization for neural machine translation with minimum Bayes risk decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 391–398, Mexico City, Mexico. Association for Computational Linguistics.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. 2024a. Advancing LLM reasoning generalists with preference trees. In *AI for Math Workshop @ ICML 2024*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. 2024b. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023a. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. 2023b. Secrets of RLHF in large language models part i: PPO. *arXiv preprint arXiv:2307.04964*.

Enyu Zhou, Guodong Zheng, Binghai Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao Xiong, Jessica Fan, Yurong Mou, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. RMB: Comprehensively benchmarking reward models in LLM alignment. *arXiv preprint arXiv:2410.09893*.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Overoptimization of BoN Sampling

Figure 5 shows the performance of BoN sampling using proxy reward models evaluated by a gold reference reward model. The proxy reward models are based on the Pythia-1B model (Biderman et al., 2023) and trained using the first 1000, 2000, and 4000 entries of the training set of AlpacaFarm. The gold reference reward model is based on the Pythia-2.8B model and trained using the entire training set (9600 entries). Spearman's rank correlation coefficients (Spearman, 1904) of the proxy reward models with the gold reference reward models are present in Table 4. The hyperparameters used in the reward model training are described in Table 10.

The performance of BoN sampling improves with larger samples up to some point and it then decreases with more samples from that point.

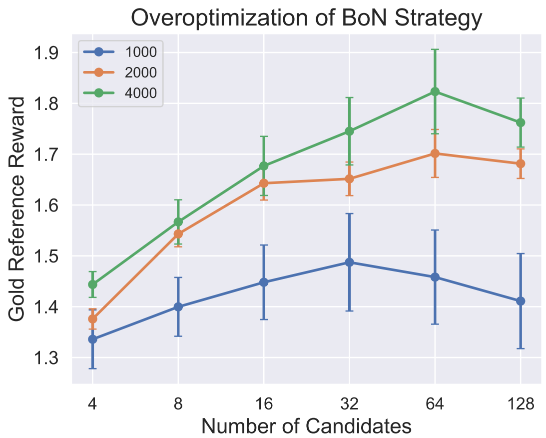


Figure 5: Performance of BoN sampling using proxy reward models. The lines show the mean and the bars show the standard deviation of three runs.

Table 4: Spearman's rank correlation coefficients of the proxy reward models with the gold reference reward model (Pythia 2.8B). The proxy reward models are trained with 1000, 2000, and 4000 instances of the training split.

#Training	ρ
1000	0.189 ± 0.264
2000	0.327 ± 0.215
4000	0.358 ± 0.224

B Evaluation of MBR Objective as a Proximity Regularizer

Table 5 shows the correlation of the distance to the center of the distribution of the sentence embeddings (i.e., L_1-norm of the component vector) with the value of the MBR objective in the hh-rlhf datasets. See Section 3 for the experimental setups. The distance from the center of the distribution has a strong negative correlation with the MBR objective. On the other hand, the correlation with the log probability of the output is weak which shows that the log probability and the KL-divergence using that is not a reliable source to quantify the proximity of the output with respect to the embedding space (Table 6).

Figure 6 shows the average normalized MBR objective values mapped to the first and second principal components. The result shows that the outputs that lie in the center of the distribution tend to have higher MBR scores, which indicates that the MBR score serves as a regularizer to keep the output faithful to the reference policy.

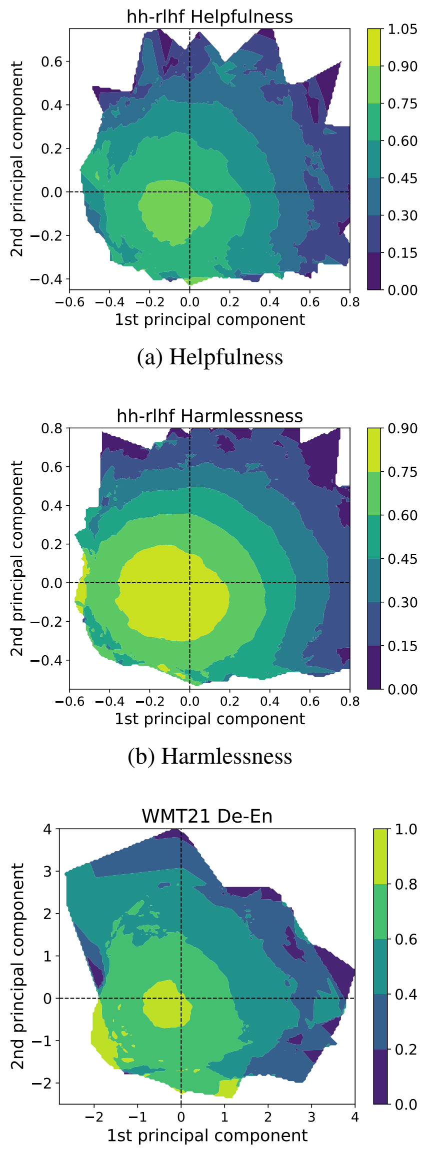
As an ablation study, we evaluate the correlation for a machine translation task using a machine translation model and a utility function for machine translation. We use WMT'21 De-En (Akhbardeh et al., 2021) as a dataset and `wmt21-dense-24-wide-x-en` (Tran et al., 2021) as the translation model. Both the embedding function and the utility function use `wmt20-comet-da` (Rei et al., 2020b). Note that `wmt20-comet-da` is not designed to be a symmetric function with respect to y and y' as the model measures the utility over y and y' and also the translation quality directly using the source text x. Figure 6c shows the mapping of the values of the MBR objective on WMT'21 De-En. Overall, we observe qualitatively the same result as in the AlpacaFarm and hh-rlhf datasets. The result shows that the MBR objective serving as a regularizer is observed in a machine translation task in addition to the instruction-following tasks.

Table 5: Correlation of the distance from the center of the distribution in the component space with the MBR objective.

Dim	PCA	ICA
hh-rlhf Helpfulness		
2	-0.5702 $\pm$ 0.2013	-0.5696 $\pm$ 0.1906
5	-0.7478 $\pm$ 0.1339	-0.6931 $\pm$ 0.1299
10	-0.8407 $\pm$ 0.1136	-0.6792 $\pm$ 0.1375
hh-rlhf Harmlessness		
2	-0.6050 $\pm$ 0.1770	-0.5917 $\pm$ 0.1727
5	-0.7536 $\pm$ 0.1305	-0.6920 $\pm$ 0.1298
10	-0.8550 $\pm$ 0.1066	-0.6909 $\pm$ 0.1311
WMT'21 De-En		
2	-0.3917 $\pm$ 0.2108	-0.3820 $\pm$ 0.2055
5	-0.5676 $\pm$ 0.1540	-0.5287 $\pm$ 0.1458
10	-0.6705 $\pm$ 0.1306	-0.5612 $\pm$ 0.1360

Table 6: Correlation of the distance from the center of the distribution in the component space with the log probability on AplacaFarm.

Dim	PCA	ICA
2	0.0826 $\pm$ 0.2340	0.0806 $\pm$ 0.2373
5	0.0954 $\pm$ 0.2315	0.0905 $\pm$ 0.2075
10	0.0784 $\pm$ 0.2357	0.0425 $\pm$ 0.2061



(a) Helpfulness

(b) Harmlessness

(c) WMT'21 De-En

Figure 6: Visualization of the mean values of the MBR objective in the space of the first and second principal components.

C Derivation of Proposition 1

We show the derivation of Proposition. 1. From the definition of Wasserstein distance with $p = 1$ (Peyré and Cuturi, 2020; Villani, 2021a), we get the following:

$$WD(\pi_y, \hat{\pi}_{\text{ref}}(\cdot|x)) = \min_{\{\mu_{i,j}\}_{i,j} \in \mathcal{J}} \sum_{i=1}^{|Y_{\text{ref}}|} \sum_{j=1}^{|Y_{\text{ref}}|} \mu_{i,j} C(y_i, y_j), \quad (11)$$

where $\mathcal{J}$ is a set of all couplings $\{\mu_{i,j}\}_{i,j}$ (Villani, 2021a):

$$\mathcal{J} = \Big\{\{\mu_{i,j}\}_{i,j} : \sum_{i=1}^{|Y_{\text{ref}}|} \mu_{i,j} = \hat{\pi}_{\text{ref}}(y_j|x), \sum_{j=1}^{|Y_{\text{ref}}|} \mu_{i,j} = \pi_y(y_i), \mu_{i,j} \geq 0\Big\}. \quad (12)$$

Because $\pi_y(y_i) = 0$ for all $y_i \neq y$ and $\mu_{i,j} \geq 0$, we get $\mu_{i,j} = 0$ for all $y_i \neq y$. Thus,

$$(11) = \min_{\mathcal{J}} \sum_{j=1}^{|Y_{\text{ref}}|} \mu_{y,j} C(y, y_j) \quad (13)$$

Using $\mu_{i,j} = 0$ for all $i \neq y$ and $\sum_{i=1}^{|Y_{\text{ref}}|} \mu_{i,j} = \hat{\pi}_{\text{ref}}(y_j|x)$, we get $\mu_{y,j} = \hat{\pi}_{\text{ref}}(y_j|x)$. Thus,

$$(13) = \min_{\mathcal{J}} \sum_{j=1}^{|Y_{\text{ref}}|} \hat{\pi}_{\text{ref}}(y_j|x) C(y, y_j) = \sum_{j=1}^{|Y_{\text{ref}}|} \hat{\pi}_{\text{ref}}(y_j|x) C(y, y_j). \quad (14)$$

Because $\hat{\pi}_{\text{ref}}(y_j|x)$ is an empirical distribution from the set of samples Y_{ref}, $\hat{\pi}_{\text{ref}}(y_j \mid x) = \frac{1}{N} \sum_{y_i \in Y_{\text{ref}}} \mathbb{I}[y_j = y_i]$. Thus,

$$(14) = \sum_{y' \in Y_{\text{ref}}} \frac{1}{N} C(y, y') \quad (15)$$

$$= - \sum_{y' \in Y_{\text{ref}}} \frac{1}{N} U(y, y'). \quad (16)$$

Thus, we get Proposition 1.

D Evaluation of KL-Regularized BoN

A naive implementation of proximity regularization for BoN sampling is to introduce KL-regularization. BoN with KL-regularization (RBoN_{KL}) can be derived from Eq. (1) as follows:

$$y_{\text{RBoN}_{\text{KL}}}(x) = \arg\max_{y \in Y_{\text{ref}}} R(x, y) - \beta \mathbb{D}_{\text{KL}}[\pi_y || \pi_{\text{ref}}(\cdot|x)], \quad (17)$$

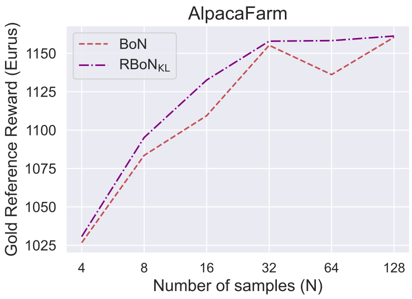
where π_y represents a policy of choosing y with a probability of 1, which is the policy it will end up with if it chooses y as the output. Thus, $\mathbb{D}_{\text{KL}}[\pi_y || \pi_{\text{ref}}(\cdot|x)]$ represents the KL divergence between the resulting policy and the reference policy. Intuitively, RBoN_{KL} optimizes the same objective as Eq. (1) but with modifications to make it available at decoding time. Eq. (17) is derived from Eq. (1) by computing the optimal response for a given x instead of computing the optimal policy.

The tradeoff between the reward and the proximity to the reference model is controlled by the hyperparameter β. With a small β, the output is more aligned with the proxy reward model. With $\beta = 0$, the vanilla BoN is restored. With larger β, the output is closer to the behavior of the reference model π_{ref}, where $\beta = +\infty$ selects the response with the highest model probability, recovering the maximum a posteriori (MAP) decoding (Stahlberg and Byrne, 2019; Eikema and Aziz, 2020; Holtzman et al., 2020).

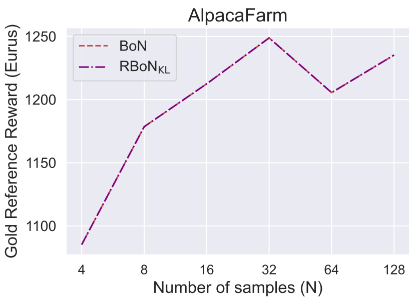
Figure 7 shows the performance of RBoN_{KL}. Overall, its improvement over BoN is marginal.

E Evaluation using Reward Model Trained on AlpacaFarm

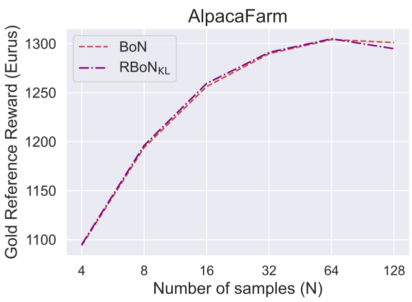
Figure 8 shows the performance of MBR-BoN compared to BoN using a reward model trained on the AlpacaFarm training set. The reward model is the gold reference reward model based on Pythia-2.8B used in Appendix A. The improvement of MBR-BoN over BoN is large when the number of samples are large and also when the proxy reward model is less aligned with the gold reference reward model. On the other hand, when the proxy reward model is trained using noiseless (the same preference annotations as the gold reference model) and large enough dataset ($|\mathcal{D}| = 4000$), the performance of MBR-BoN is on par with BoN.



(a) Proxy: SHP-Large



(b) Proxy: SHP-XL



(c) Proxy: OASST

Figure 7: Evaluation of the RBoN$_{\mathrm{KL}}$ using Mistral on the AlpacaFarm dataset.

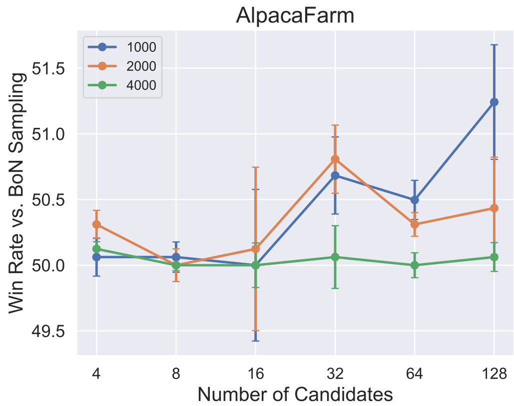


Figure 8: The average win rate of the MBR-BoN against BoN using a reward model trained on the training set of AlpacaFarm.

F Analysis on the Size of the Development Set for Tuning Beta

We run a posthoc analysis to evaluate the effect of the size of the development set to tune the hyperparameter β for MBR-BoN. Figure 9 shows the performance of MBR-BoN with varying sizes of development set to compute the β, from 10 to 1000. We observe that the score is relatively consistent and MBR-BoN outperforms BoN even with 10 examples for fine-tuning β.

G Effect of the Regularization Strength

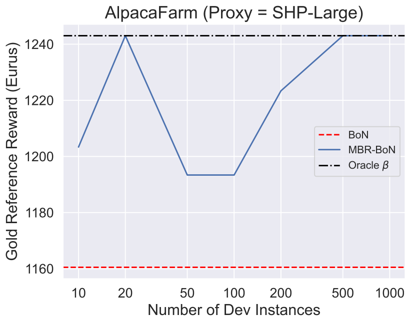
Correlation Coefficient. To understand the effect of the regularization strength on the performance of RBoN under different pairs of proxy and gold reward models, we evaluate RBoN using SHP-Large, SHP-XL, OASST, and PairRM (Jiang et al., 2023b) as gold reward models. Figure 10 reports the average Spearman's rank correlation coefficient ρ of a pair of reward models (Spearman, 1904). Note that SHP-Large and SHP-XL reward models are highly correlated as they are trained on the same training procedure.

Tradeoff between Proxy Reward and Proximity Scores. Figure 11 shows the tradeoff of the proxy reward score and the MBR objective score with different values of β on MBR-BoN. The result shows that the hyperparameter β effectively controls the weights over the proxy reward model and proximity to the reference policy.

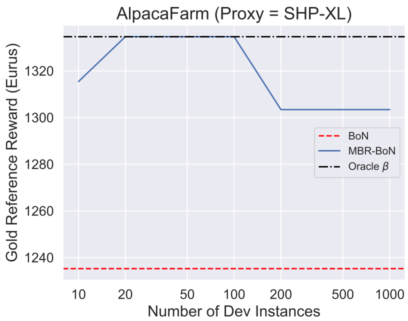
Evaluation of MBR-BoN using Various Reference Reward Models. We perform the generation (BoN and MBR-BoN) using one of the reward models as the proxy reward model and evaluate the selected responses using the remaining reward models as the gold reference rewards. We do not use PairRM as a proxy reward model because it is a pairwise reward model that estimates the preference for a pair of responses rather than computing an absolute preference for a response. The use of a pairwise reward model as a proxy reward model for RBoN is future work.

Figure 12 shows the performance of BoN and MBR-BoN with varying β with $N = 128$ using Mistral on the AlpacaFarm dataset. MBR-BoN outperforms BoN in all settings except when the proxy reward model is highly correlated with the gold reward model (e.g., SHP-Large and SHP-XL).

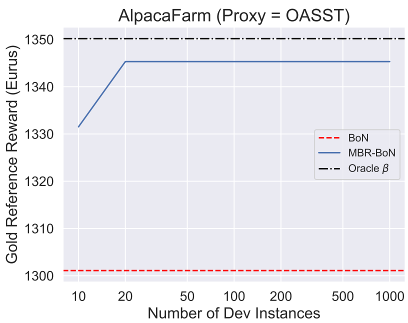
The experiment shows that the optimal β depends on various factors, but the strength of the correlation between the proxy reward model and



(a) SHP-Large



(b) SHP-XL



(c) OASST

Figure 9: Evaluation of MBR-BoN with varying sizes of development set to tune the optimal β.

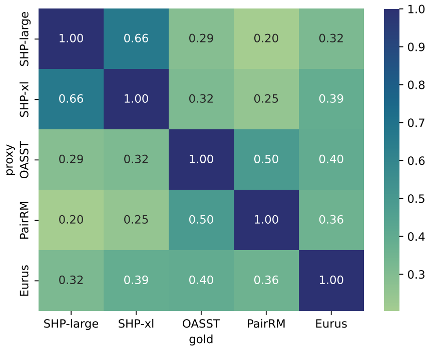
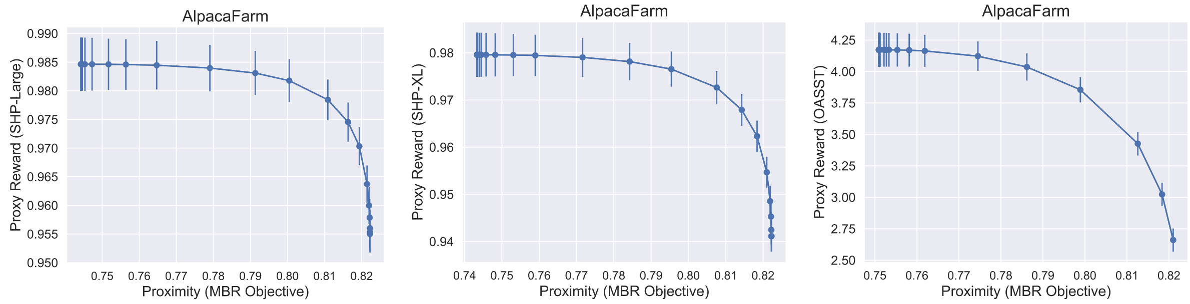


Figure 10: Average Spearman's rank correlation coefficient of the reward models in the evaluation split of the AlpacaFarm dataset for the responses generated by Mistral. 128 responses are used to compute Spearman's rank correlation for each instruction, averaged over the 805 instructions.

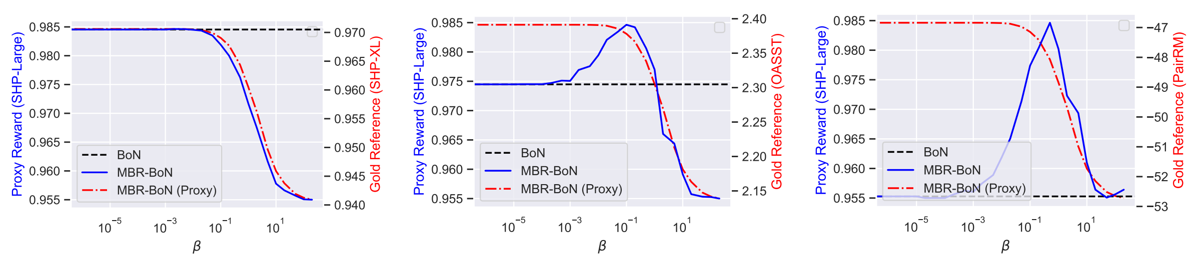
the gold reference reward seems to be the key factor. For example, SHP-Large is strongly correlated with SHP-XL ($\rho = 0.66$), so the optimal β is close to 0. In this case, MBR-BoN has little to no advantage over BoN. On the other hand, SHP-Large is only weakly correlated with OASST and PairRM ($\rho = 0.29, 0.20$), where the optimal β for SHP-Large $\rightarrow$ OASST and PairRM is large ($\beta = 0.1 - 1.0$).

Figures 13 and 14 show the performance of BoN ($\beta = 0$), MBR decoding ($\beta = +\infty$), and MBR-BoN with different number of samples N using Mistral and Dolly on AlpacaFarm. We observe qualitatively similar results with smaller N to the result of $N = 128$ in Figure 2.

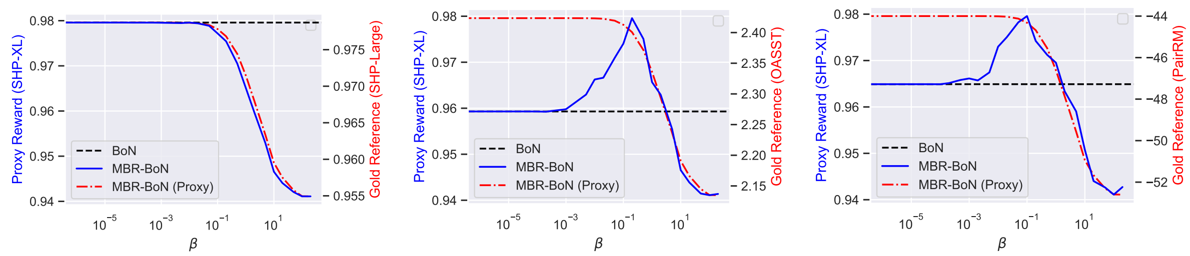


(a) SHP-Large (b) SHP-XL (c) OASST

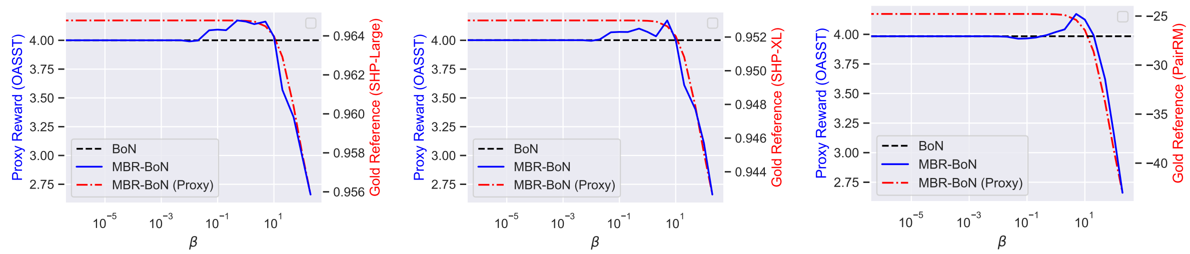
Figure 11: The tradeoff of the proxy reward score and proximity (MBR objective) with MBR-BoN using different β strengths on AlpacaFarm. The responses are generated by Mistral. The number of samples N is 128. The line shows the mean and the error bar shows the standard error of the estimation of the mean value.



(a) SHP-Large → SHP-XL (b) SHP-Large → OASST (c) SHP-Large → PairRM

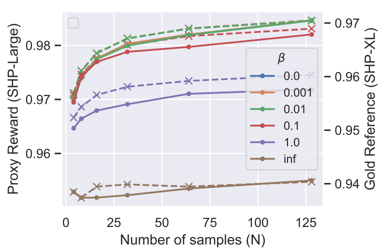


(d) SHP-XL → SHP-Large (e) SHP-XL → OASST (f) SHP-XL → PairRM

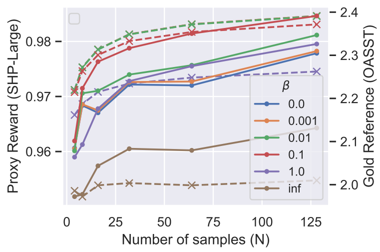


(g) OASST → SHP-Large (h) OASST → SHP-XL (i) OASST → PairRM

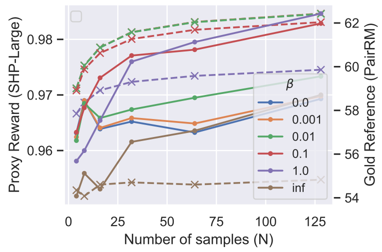
Figure 12: The gold reward score and the proxy reward score of the MBR-BoN with different regularization strengths and reward models. The captions of the subfigures show the proxy and the gold reward model (Proxy → Gold). The performance of BoN is shown in the horizontal lines. The responses are generated by Mistral. The number of samples N is 128.



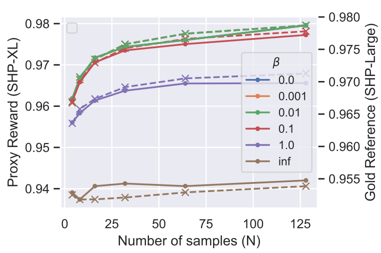
(a) SHP-Large → SHP-XL



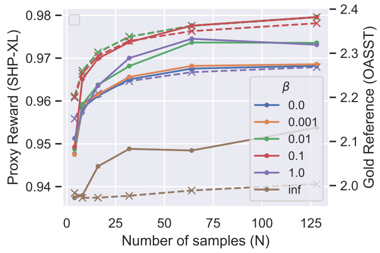
(b) SHP-Large → OASST



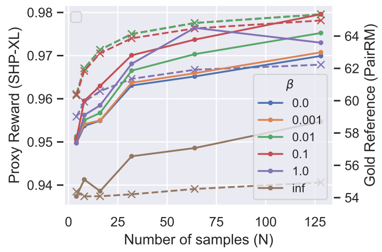
(c) SHP-Large → PairRM



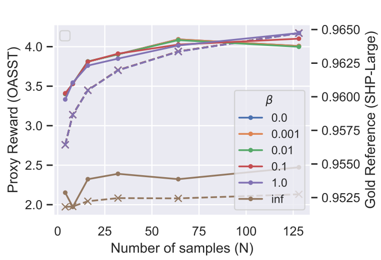
(d) SHP-XL → SHP-Large



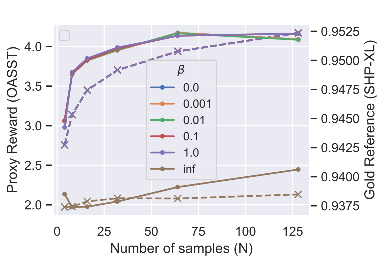
(e) SHP-XL → OASST



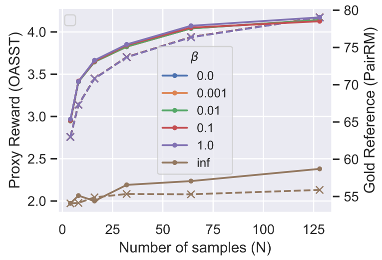
(f) SHP-XL → PairRM



(g) OASST → SHP-Large

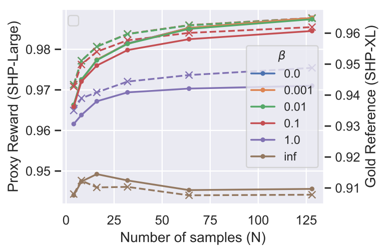


(h) OASST → SHP-XL

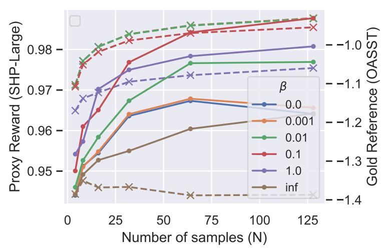


(i) OASST → PairRM

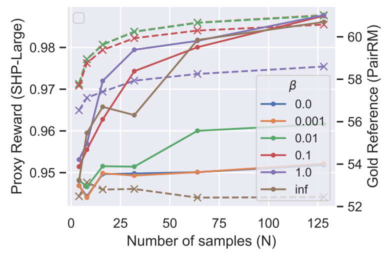
Figure 13: Evaluation of MBR-BoN using Mistral on AlpacaFarm. The gold reward score and the proxy reward score of the MBR-BoN with different regularization strengths and reward models. The captions of the subfigures show the proxy and the gold reward model (Proxy → Gold). The reward scores of the reference reward (right axis) are shown in solid lines whereas the reward scores of the proxy reward (left axis) are shown in dashed lines. $\beta = \text{inf}$ corresponds to the MBR decoding.



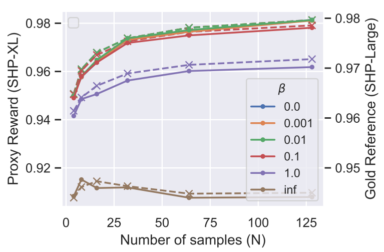
(a) SHP-Large → SHP-XL



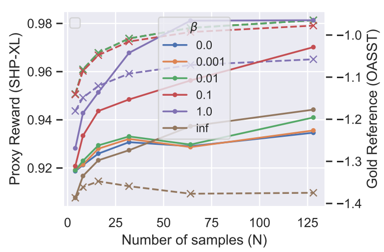
(b) SHP-Large → OASST



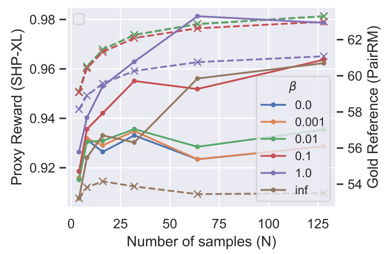
(c) SHP-Large → PairRM



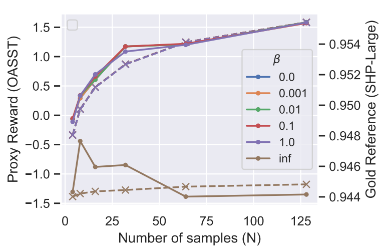
(d) SHP-XL → SHP-Large



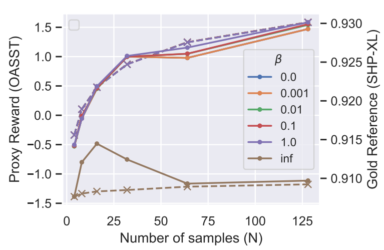
(e) SHP-XL → OASST



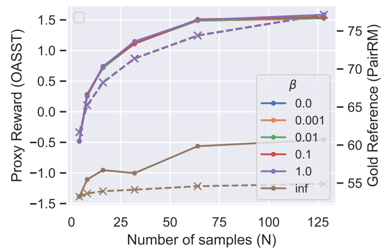
(f) SHP-XL → PairRM



(g) OASST → SHP-Large



(h) OASST → SHP-XL



(i) OASST → PairRM

Figure 14: Evaluation of MBR-BoN using Dolly on AlpacaFarm. The gold reward score and the proxy reward score of the MBR-BoN with different regularization strengths and reward models. The captions of the subfigures show the proxy and the gold reward model (Proxy → Gold). The reward scores of the reference reward (right axis) are shown in solid lines whereas the reward scores of the proxy reward (left axis) are shown in dashed lines. $\beta =$inf corresponds to the MBR decoding.

H GPT-4o Evaluation of the DPO

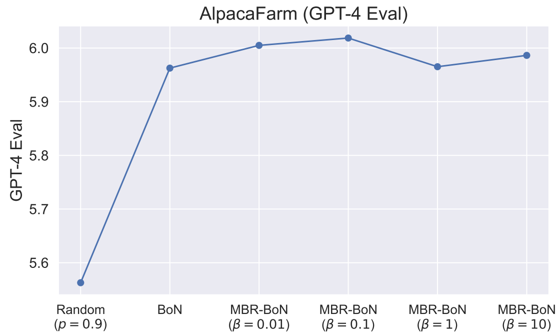
Figure 15 shows the average score of the models trained by DPO in Section 5.2 using GPT-4o as a judge (Zheng et al., 2023a; OpenAI et al., 2024). We evaluate using the first 300 entries of the test split of the datasets. We use the following prompt to ask GPT-4o to evaluate the quality of the output.

> Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".
>
> [Question]
> {question}
> [The Start of Assistant's Answer]
> {answer}
> [The End of Assistant's Answer]

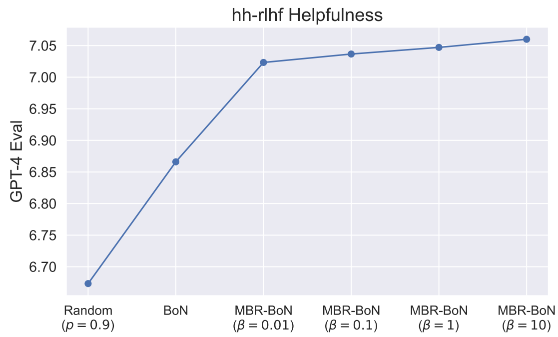
The model name is gpt-4o and the model version is 2024-05-13. We set the model temperature, frequency penalty, and presence penalty to 0. Overall, we observe the same qualitative result that models trained using the proposed method outperform the model using the BoN sampling. For the generations of the fine-tuned models we evaluate, the average agreement of GPT-4o evaluation with the Eurus reward model is 0.708 for AlpacaFarm and 0.750 for hh-rlhf datasets.

I Walltime

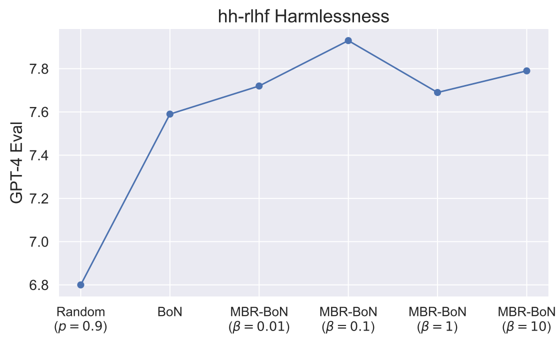
We report the wall clock time of BoN and MBR-BoN in Table 7. The batch size for generating samples is set to 4. The code base is based on Huggingface's Transformers library (Wolf et al., 2020) and is not based on a library optimized for inference speed (e.g., vLLM; Kwon et al., 2023). We use OASST reward model with the batch size set to 8. We set the batch size for the computation of the similarity between sequences for the MBR values to 64. In our code base, we store the generated samples, computed reward values, and



(a) AlpacaFarm



(b) Helpfulness



(c) Harmlessness

Figure 15: GPT-4o Evaluation of the fine-tuned models trained using MBR-BoN.

Table 7: Summary of wall clock time of BoN and MBR-BoN with $N = 128$ for AlpacaFarm dataset. All experiements are run on an NVIDIA T4 GPU.

	Run time (seconds)	
	BoN	MBR-BoN
Generate samples	134	134
Compute the reward values	0.1	0.1
Compute the MBR values	-	2

Table 8: Generation hyperparameters used in Section 5.1 and 5.2

Parameter	Value
Max instruction length	256
Max new tokens	256
Temperature	1.0
Top-p	0.9

the MBR values to a cloud storage. The reported wall clock time may also include the time for the logging procedures. The wall clock time depends on various factors including the code base and the hardware. All the experiments are conducted using an NVIDIA T4 GPU.

J Hyperparameters

Table 8 describes the hyperparameters used to generate responses from the π_{ref}. The parameters are used for both Sections 5.1 and 5.2. Table 9 summarizes the hyperparameters used for DPO in Section 5.2.

K Reproducibility Statement

All datasets and models used in the experiments are publicly available except for GPT-4o (Table 11). The code is implemented using Huggingface's Transformers library (Wolf et al., 2020) and TRL library (von Werra et al., 2020). The PCA and ICA are implemented using scikit-learn (Pedregosa et al., 2011). Our code is available at https://github.com/CyberAgentAILab/regularized-bon with an MIT license.

Table 9: DPO hyperparameters used in Section 5.2.

Parameter	Value
Epochs	3
Learning rate	1e-5
Optimizer	AdamW
Batch size	4
Regularization factor (β)	0.1
LoRA r	128
LoRA α	32

Table 10: Hyperparameters for training reward models used in Appendix A. The values follow the defaults of the TRL library.

Parameter	Value
Epochs	3
Learning rate	5e-05
Optimizer	AdamW
Batch size	8

Table 11: List of datasets and models used in the experiments.

Name	Reference
AlpacaFarm	(Dubois et al., 2023) https://huggingface.co/datasets/tatsu-lab/alpaca_farm
Anthropic's hh-rlhf	(Bai et al., 2022) https://huggingface.co/datasets/Anthropic/hh-rlhf
WMT'21 De-En	(Akhbardeh et al., 2021) https://github.com/wmt-conference/wmt21-news-systems
mistral-7b-sft-beta (Mistral)	(Jiang et al., 2023a; Tunstall et al., 2024) https://huggingface.co/HuggingFaceH4/mistral-7b-sft-beta
dolly-v2-3b (Dolly)	(Conover et al., 2023) https://huggingface.co/databricks/dolly-v2-3b
Pythia-1B	(Biderman et al., 2023) https://huggingface.co/EleutherAI/pythia-1b
Pythia-2.8B	(Biderman et al., 2023) https://huggingface.co/EleutherAI/pythia-2.8b
wmt21-dense-24-wide	(Tran et al., 2021) https://huggingface.co/facebook/wmt21-dense-24-wide-x-en
SHP-Large	(Ethayarajh et al., 2022) https://huggingface.co/stanfordnlp/SteamSHP-flan-t5-large
SHP-XL	(Ethayarajh et al., 2022) https://huggingface.co/stanfordnlp/SteamSHP-flan-t5-xl
OASST	(Köpf et al., 2023) https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2
PairRM	(Jiang et al., 2023b) https://huggingface.co/llm-blender/PairRM
Eurus	(Yuan et al., 2024a) https://huggingface.co/openbmb/Eurus-RM-7b
MPNet	(Song et al., 2020) https://huggingface.co/sentence-transformers/all-mpnet-base-v2
wmt20-comet-da	(Rei et al., 2020a) https://huggingface.co/Unbabel/wmt20-comet-da