Variational Best-of-N Alignment

Afra Amini **Tim Vieira** **Elliott Ash** **Ryan Cotterell**
ETH Zürich
{aamini, ashe, rcotterell}@ethz.ch tim.f.vieira@gmail.com

Abstract

Best-of-N (BoN) is a popular and effective algorithm for aligning language models to human preferences. The algorithm works as follows: at inference time, N samples are drawn from the language model, and the sample with the highest reward, as judged by a reward model, is returned as the output. Despite its effectiveness, BoN is computationally expensive; it reduces sampling throughput by a factor of N. To make BoN more efficient at inference time, one strategy is to fine-tune the language model to mimic what BoN does during inference. To achieve this, we derive the distribution induced by the BoN algorithm. We then propose to fine-tune the language model to minimize backward KL divergence to the BoN distribution. Our approach is analogous to mean-field variational inference and, thus, we term it variational BoN (vBoN). To the extent this fine-tuning is successful and we end up with a good approximation, we have reduced the inference cost by a factor of N. Our experiments on controlled generation and summarization tasks show that BoN is the most effective alignment method, and our variational approximation to BoN achieves the closest performance to BoN and surpasses models fine-tuned using the standard KL-constrained RL objective. In the controlled generation task, vBoN appears more frequently on the Pareto frontier of reward and KL divergence compared to other alignment methods. In the summarization task, vBoN achieves high reward values across various sampling temperatures.

1 Introduction

Language models are pre-trained on large corpora to model a distribution over natural language text.[1] Beyond their initial pre-training, they are often additionally fine-tuned on domain-specific data through a process called **supervised fine-tuning (SFT)**. The goal of SFT is to enable the model to better perform various downstream tasks of interest. While the fine-tuned model, called the **reference model** in our paper, is indeed typically much better at performing the downstream task of interest, e.g., dialogue generation or summarization, it may still generate undesirable content, e.g., harmful or offensive text. To mitigate this issue, **aligning** the reference model to human preferences has become a fundamental step in the development of modern large language models (Touvron et al., 2023; OpenAI et al., 2023; Gemini et al., 2024).

The degree to which text is aligned with human preferences is typically operationalized using a real-valued reward function. Rather than constructing a reward function by hand, it is typically estimated from a dataset of human preferences.[2] And, after estimation, we expect the reward function to return higher values for text that is more likely to be preferred by humans, and lower values for text that is more likely to be dispreferred. Then, given an estimated reward function, an alignment algorithm further alters the reference models in a manner such that it places the highest probability on that text that is high reward under the reward model *and* high probability under the reference model.

[1] Many language models are also used to model text in non-natural languages, e.g., programming languages.

[2] In some cases, the reward model is not estimated from human preference data. It is either known, e.g., code-based execution scores, or given by a classifier, e.g., toxicity or sentiment classifiers.

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).

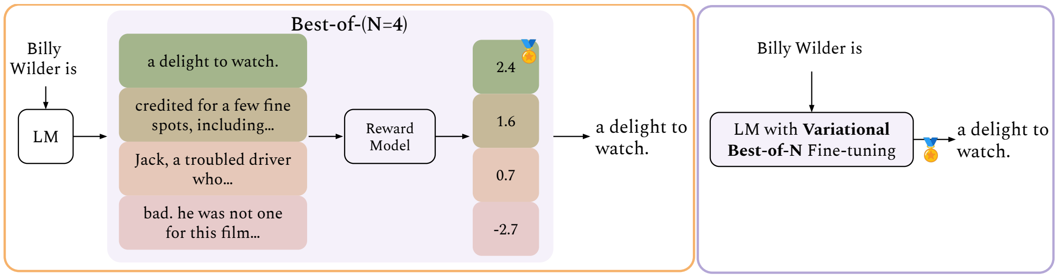

Billy Wilder is
LM
Best-of-(N=4)
a delight to watch.
credited for a few fine spots, including...
Jack, a troubled driver who...
bad. he was not one for this film...
Reward Model
2.4
1.6
0.7
-2.7
a delight to watch.
Billy Wilder is
LM with Variational Best-of-N Fine-tuning
a delight to watch.

Figure 1: Best-of-N (on the left) is an effective alignment-via-inference method: it draws N samples from the language model, ranks them according to a reward model, and outputs the best sample. Variational Best-of-N (on the right) approximates this process via fine-tuning. The goal is to ensure that sampling a single string from the fine-tuned model produces a result equivalent to applying Best-of-N. This approach allows us to achieve similar performance while increasing the throughput by a factor of N.

Alignment algorithms can be taxonomized into two groups: (i) alignment via fine-tuning, where we change the language model's parameters to achieve alignment (Christiano et al., 2017; Rafailov et al., 2023), and (ii) alignment via inference (Nakano et al., 2022; Mudgal et al., 2024). A common alignment-via-fine-tuning method is reinforcement learning from human feedback (RLHF; Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022). RLHF typically consists of further fine-tuning the language model under a **KL-constrained RL objective**, which is made up of two terms: a term that encourages the model to maximize the reward, and a term that discourages high KL divergence between the language model and the reference model. This objective is often maximized with an RL algorithm, e.g., proximal policy optimization (PPO; Schulman et al., 2017). A common alignment-via-inference method is the Best-of-N (BoN; Stiennon et al., 2020) algorithm. As such, it does *not* require any fine-tuning of the language model. The algorithm is straightforward: One draws N samples from the reference model and returns the text that achieves the highest reward among those N samples. The BoN algorithm has also been effectively applied in controlled decoding (Yang & Klein, 2021; Mudgal et al., 2024) and to generate a dataset for supervised fine-tuning (Touvron et al., 2023).

Despite its simplicity, BoN has proven incredibly practical in generating high-reward text that still has a high probability under the reference model. Theoretically, Yang et al. (2024) prove that under some simplifying assumptions, the BoN distribution is asymptotically equivalent to the optimal distribution under the KL-constrained RL objective. Empirically, it has been repeatedly shown (Gao et al., 2023; Rafailov et al., 2023; Mudgal et al., 2024) that BoN often appears on the frontier of reward and KL curves, surpassing the performance of models fine-tuned with RLHF. However, the main factor preventing BoN from replacing fine-tuning methods for alignment is its significant computational overhead during inference. Even when sampling is done in parallel, BoN decreases the text generation throughput by a factor of N. This drawback limits its practicality for generating text from large language models.

To speed up BoN, we devise a scheme to convert it into an alignment-via-fine-tuning algorithm rather than an alignment-via-inference algorithm. To this end, we first formally derive the probability distribution induced by the BoN algorithm. Then we approximate this distribution by minimizing the reverse KL divergence between the language model and the BoN distribution. This leads to an optimization objective that we refer to as the vBoN objective. By analyzing a lower bound of this objective, we find that it behaves similarly to the KL-regularization objective in the limit, i.e., $N \to 1$ or $N \to \infty$. Importantly, the vBoN objective has a unique and useful property: it is insensitive to applying any monotonically increasing function to the reward values. This distinctive feature, along with the empirical success of the BoN algorithm, suggests that the vBoN objective is a promising and interesting objective to explore. Finally, we fine-tune the language model using PPO to optimize the vBoN objective. Our scheme, depicted in Fig. 1, allows us to achieve performance close to that of the BoN algorithm while increasing the inference throughput by a factor of N.

We experiment with our method on controlled generation and summarization tasks. We compare vBoN against models fine-tuned with the KL-constrained RL objective. In the controlled generation task, our results suggest that models fine-tuned with the vBoN objective are most likely to appear on the Pareto frontier of reward vs. KL curves compared to other alignment-via-finetuning methods, suggesting a better trade-off between attaining high rewards and not diverging too far from the reference model. Moreover, in the summarization task, we observe fine-tuning with vBoN leads to higher reward values and win rates on average compared to models fine-tuned with KL-constrained RL objective.

2 Background: Reinforcement Learning from Human Feedback

Let Σ be an **alphabet**, a finite, non-empty set of symbols. A **string** is a finite sequence of symbols drawn from Σ. A **language model** is a distribution over strings $\boldsymbol{y} \in \Sigma^*$, where Σ^* is the set of all strings over the alphabet Σ. In this paper, we consider language models, e.g., those based on neural networks, that are parameterized by a real vector $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, denoted as $\pi_{\boldsymbol{\theta}}$. Furthermore, we restrict ourselves to language models that are differentiable functions of $\boldsymbol{\theta}$. In conditional generation tasks, e.g., summarization or dialogue generation, it is desirable to prompt the language model with a string $\boldsymbol{x} \in \Sigma^*$. Consequently, we consider prompted language models, i.e., those that give a conditional distribution over response strings $\boldsymbol{y}$, given a prompt string $\boldsymbol{x}$, as $\pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x})$. However, for notational convenience, we will drop the explicit conditioning on the prompt $\boldsymbol{x}$ and simply write $\pi_{\boldsymbol{\theta}}(\boldsymbol{y})$.

Algorithms for RLHF fine-tune the language model to increase the expected reward of the strings sampled from it while not diverging too far from the reference model. RLHF consists of three steps. First, the language model is fine-tuned on a task-specific dataset using the maximum-likelihood objective. Recall we term the language model after this step the reference model and show that with π_{ref}. Next, a **reward model** $r \colon \Sigma^* \to \mathbb{R}$ is trained to capture human preferences; the reward of a string is high if it is preferred by humans. Finally, the reference model is fine-tuned to maximize the KL-constrained RL objective,

$$\mathcal{J}^{\text{RL}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \pi_{\boldsymbol{\theta}}} \Big[r(\boldsymbol{y}) \Big] - \beta \, D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \, \| \, \pi_{\text{ref}}\big), \tag{1}$$

where $D_{\text{KL}}(\cdot)$ is the KL divergence between two distribution. This objective encourages the model to put more probability mass on strings that have high rewards under the reward model while penalizing it not to deviate too far from the reference model. Levine (2018) show that the optimal probability distribution that maximizes this objective is

$$\pi^\star_{\boldsymbol{\theta}}(\boldsymbol{y}) = \frac{1}{Z} \, \pi_{\text{ref}}(\boldsymbol{y}) \exp\Big(\frac{1}{\beta} r(\boldsymbol{y})\Big), \quad Z = \sum_{\boldsymbol{y} \in \Sigma^*} \pi_{\text{ref}}(\boldsymbol{y}) \exp\Big(\frac{1}{\beta} r(\boldsymbol{y})\Big). \tag{2}$$

$\pi^\star_{\boldsymbol{\theta}}$ is simply the reference model reweighted by the exponent of reward values and normalized by the partition function Z. Notably, we can not directly sample from $\pi^\star_{\boldsymbol{\theta}}$ because the partition function Z may be difficult to compute—it involves an infinite sum after all. However, a heuristic approach to sampling from $\pi^\star_{\boldsymbol{\theta}}$ would be to sample many strings from π_{ref} and only keep those that have high rewards. Indeed, this heuristic is the motivation behind the BoN algorithm.

3 Deriving the Best-of-N Objective

Best-of-N algorithm is a simple alignment-via-inference algorithm. The algorithm works as follows. Let $Y_N = \{\boldsymbol{y}^{(n)}\}_{n=1}^N$ be the multi-set containing N i.i.d samples from π_{ref}. Then, BoN algorithm returns $\boldsymbol{y}^\star$, where[3]

$$\boldsymbol{y}^\star = \mathop{\text{argmax}}_{\boldsymbol{y}^{(n)} \in Y_N} r(\boldsymbol{y}^{(n)}). \tag{3}$$

We show the probability distribution induced from BoN sampling algorithm with π_{bon}. Importantly, π_{bon} is *not* the optimal distribution under Eq. (1), the KL-constrained RL objective.[4] Nevertheless, the BoN algorithm often performs well—even compared to RLHF-based methods. This raises the question: under what optimization objective is π_{bon} the optimal distribution? To derive such an objective, we begin by computing the probability of strings under π_{bon}.

[3] We assume that the argmax is unique, or ties are broken in a well-defined manner.

[4] Under some simplifying assumptions, however, Yang et al. (2024) show that π_{bon} is asymptotically equal to $\pi^\star_{\boldsymbol{\theta}}$.

Proposition 1. *Suppose* $r\colon \Sigma^* \to \mathbb{R}$ *is a one-to-one mapping. Then, the probability that a string* $\boldsymbol{y} \sim \pi_{bon}$ *is given by*

$$\pi_{bon}(\boldsymbol{y}) = \sum_{i=1}^{N} \binom{N}{i} \mathrm{F}\big(r(\boldsymbol{y})\big)^{N-i} \pi_{\text{ref}}(\boldsymbol{y})^{i}, \tag{4}$$

where we define $\mathrm{F}(R)$ *as*

$$\mathrm{F}(R) \overset{\text{def}}{=} \mathbb{P}\left(r(\boldsymbol{y}) < R\right). \tag{5}$$

Proof. See App. A. ■

We now describe how to fine-tune the language model to approximate π_{bon} by minimizing the reverse KL divergence between $\pi_{\boldsymbol{\theta}}$ and π_{bon}. Concretely, we maximize the following objective:

$$\begin{aligned}
\mathcal{J}^{\text{vBoN}}(\boldsymbol{\theta}) = -D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \mid\mid \pi_{\text{bon}}\big) &= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}} \Big[\log \pi_{\text{bon}}(\boldsymbol{y}) - \log \pi_{\boldsymbol{\theta}}(\boldsymbol{y})\Big] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}} \Big[\log \pi_{\text{bon}}(\boldsymbol{y})\Big] + \mathrm{H}\big(\pi_{\boldsymbol{\theta}}\big) \\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}} \Big[\log \sum_{i=1}^{N} \binom{N}{i} \mathrm{F}\big(r(\boldsymbol{y})\big)^{N-i} \pi_{\text{ref}}(\boldsymbol{y})^{i}\Big] + \mathrm{H}\big(\pi_{\boldsymbol{\theta}}\big).
\end{aligned} \tag{6}$$

This is an entropy-regularized objective, where we use the probability of the string under the BoN distribution as the reward and discourage the model from having low entropy.

Monotonically invariant. An important property of the variational BoN objective is that it is invariant to applying any monotonically increasing function to rewards. This is because the vBoN objective relies on reward values solely through F, which, as defined in Eq. (5), remains unchanged under any monotonically increasing transformation of reward values. This implies that the vBoN objective is insensitive to the outliers and the scale of rewards. This property is especially important as RL algorithms are notoriously sensitive to the scale of reward values (Henderson et al., 2018; Schaul et al., 2021).

Approximating $\log \mathrm{F}(\cdot)$**.** Maximizing Eq. (6) requires us to compute $\log \mathrm{F}(\cdot)$ for any $r(\boldsymbol{y})$. This, however, is computationally expensive, as we have to sum over the probabilities of all strings that have rewards less than $r(\boldsymbol{y})$. Fortunately, we can instead maximize a lower bound of Eq. (6) using a Monte Carlo estimator of $\mathrm{F}(\cdot)$. Concretely, we can write $\mathrm{F}(\cdot)$ as an expectation,

$$\mathrm{F}\big(R\big) = \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\text{ref}}} \big[\mathbb{1}\{r(\boldsymbol{y}) < R\}\big]. \tag{7}$$

We approximate $\mathrm{F}\big(R\big)$ using M i.i.d. samples from π_{ref}, termed $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(M)} \overset{\text{i.i.d.}}{\sim} \pi_{\text{ref}}$, and $\widehat{\mathrm{F}}\big(R\big) \overset{\text{def}}{=} \frac{1}{M}\sum_{m=1}^{M} \mathbb{1}\{r(\boldsymbol{y}^{(m)}) < R\}$. We then take log of this Monte Carlo estimator as a biased, but consistent estimator of $\log \mathrm{F}(\cdot)$ in Eq. (6).[5] In §5.1 we empirically assess the number of samples we need so that $\log \widehat{\mathrm{F}}$ converges to $\log \mathrm{F}$.

[5] Using Jensen's inequality, we show biasedness. Concretely, note the following lower bound

$$\log \mathrm{F}\big(R\big) = \log \mathop{\mathbb{E}}_{\boldsymbol{y}^{(1)},\ldots,\boldsymbol{y}^{(M)}} \left[\frac{1}{M}\sum_{m=1}^{M} \mathbb{1}\{r(\boldsymbol{y}^{(m)}) < R\}\right] \tag{8a}$$

$$\geq \mathop{\mathbb{E}}_{\boldsymbol{y}^{(1)},\ldots,\boldsymbol{y}^{(M)}} \left[\log\left(\frac{1}{M}\sum_{m=1}^{M} \mathbb{1}\{r(\boldsymbol{y}^{(m)}) < R\}\right)\right], \tag{8b}$$

where Jensen's inequality is applicable because log is concave. Consistency can be shown with an application of the delta method (§5.5.4; Casella & Berger, 2001).

4 Comparing BoN and RL Objectives

To explore the connection between the vBoN objective and the KL-regularized RL objective, we derive two lower bounds for $\mathcal{J}^{\text{vBoN}}$. Through these lower bounds, we can get more insights on how the reward function is used in the variational BoN objective, and why this objective discourages high KL divergence from the reference model.

To derive the first lower bound, we substitute the BoN distribution in Eq. (4) into the vBoN objective in Eq. (6). We then use Jensen's inequality to bound this objective, as explained in the following theorem.

Theorem 2. *Let* $\alpha = \frac{(N+2)(N-1)}{2}$, $\beta = \frac{N(N+1)}{2}$, *and* $\gamma = \frac{N(N-1)}{2}$. *Then, we have* $\mathcal{J}^{\text{vBoN}}(\boldsymbol{\theta}) \geq L_1(\boldsymbol{\theta})$, *where we further define*

$$L_1(\boldsymbol{\theta}) \overset{\text{def}}{=} \gamma \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \pi_{\boldsymbol{\theta}}} \Big[\log \mathrm{F}\big(r(\boldsymbol{y})\big) \Big] - \alpha \mathrm{H}\big(\pi_{\boldsymbol{\theta}}\big) - \beta D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \mid\mid \pi_{\text{ref}}\big). \tag{9}$$

Proof. See App. C. ■

We can already see the connection between $L_1(\boldsymbol{\theta})$ and the KL-regularized RL objective, Eq. (1). They both encourage maximizing a function of reward values. In the BoN objective this function is $F(\cdot)$, while in the KL-regularized RL objective, it is an identity function, as we directly maximize the expected rewards. Furthermore, both objectives include a negative KL-divergence term between the language model and the reference model. $L_1(\boldsymbol{\theta})$ further encourages the model to have low entropy. In fact, we can show that Thm. 2 also holds for $\alpha = 0, \beta = 1, \gamma = N - 1$. This further simplifies L_1 to arrive at a tighter lower bound, which we call L_2, that is even more similar to the KL-regularized RL objective.

Theorem 3. *We have* $\mathcal{J}^{\text{vBoN}}(\boldsymbol{\theta}) \geq L_2(\boldsymbol{\theta}) \geq L_1(\boldsymbol{\theta})$, *where*

$$L_2(\boldsymbol{\theta}) \overset{\text{def}}{=} (N-1) \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \pi_{\boldsymbol{\theta}}} \Big[\log \mathrm{F}\big(r(\boldsymbol{y})\big) \Big] - D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \,||\, \pi_{\text{ref}}\big). \tag{10}$$

Proof. See App. D. ■

Empirically, we observe that models that are fine-tuned to maximize $L_2(\boldsymbol{\theta})$ perform competitively to the ones that are fine-tuned to maximize the vBoN objective; see App. F for experimental results. Interestingly, if we compare Eq. (10) to the KL-constrained RL objective, Eq. (1), we see they are very similar. Both objectives consist of two terms: one for maximizing a function of reward values and another to prevent the model from diverging too much from the reference model.

Comparing Eq. (10) and Eq. (1) further suggests that N (in the vBoN objective) acts as a regularization parameter. As $N \to 1$, the optimal distribution gets closer to π_{ref}, which has the same effect as $\beta \to \infty$ in Eq. (1). Furthermore, as $N \to \infty$, the optimal distribution only generates the string with the maximum rewards, which is equivalent to $\beta \to 0$ in Eq. (1). Importantly, in both limits, the optimal distribution under the KL-regularized RL objective and the vBoN objective are equivalent.

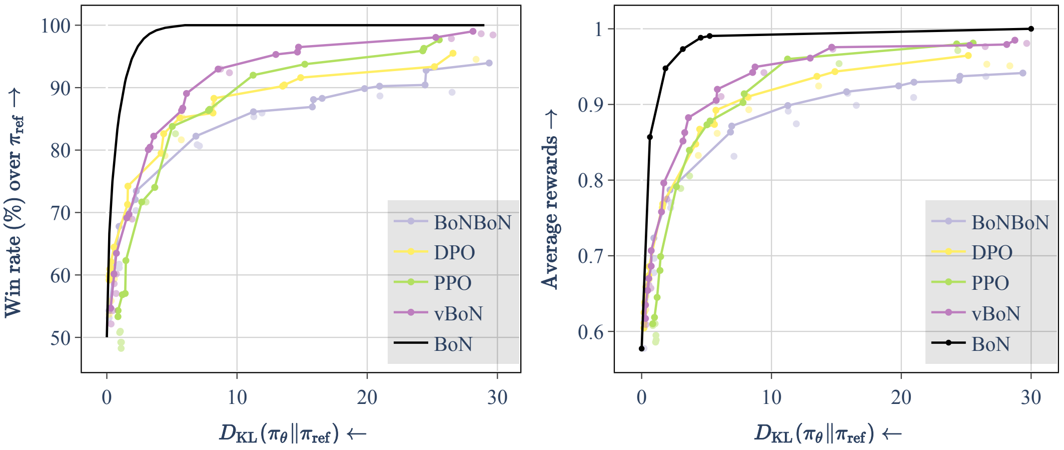
5 Sentiment Control

We now employ the variational BoN objective, Eq. (6), to fine-tune language models. We perform an open-ended text generation task where the goal is to generate movie reviews with positive sentiment.

The reference model, π_{ref}, is GPT-IMDB[6], a GPT-2 (Radford et al., 2019) model fine-tuned on IMDB corpus (Maas et al., 2011). We use a binary sentiment classifier,[7] denoted as p, with two classes $\{\text{POS}, \text{NEG}\}$ as the reward model, and define $r(\boldsymbol{y}) \overset{\text{def}}{=} p(\text{POS} \mid \boldsymbol{y})$. Following Rafailov et al. (2023), we sample 5000 movie reviews from the training set of IMDB dataset and for each sample, we randomly choose a prefix length between $2 - 8$ and take that prefix as the prompt. We further generate 512 prompts in the same way from the test set of IMDB that we use to evaluate our models.

[6] Specifically, we use https://huggingface.co/lvwerra/gpt2-imdb.

[7] Specifically, we use https://huggingface.co/lvwerra/distilbert-imdb.



(a) 4% of points on Pareto front belong to BoNBoN, 4% to PPO, 42% to DPO, and 50% to vBoN.

(b) 7% of points on Pareto from belong to BoNBoN, 10% DPO, 33% PPO, and 50% vBoN.

Figure 2: Steering generated movie reviews towards positive sentiment. Points that are not on the Pareto front of each method have lower opacity. BoN is the most effective approach in achieving high win rates and high rewards while not diverging too far from the reference model. Our variational approximation to BoN gets closest to the performance of BoN compared to other fine-tuning methods, as reflected in the percentage of times it appears on the Pareto front.

We fine-tune the reference model with PPO using the vBoN objective Eq. (6). Then, we compare the performance of the fine-tuned model (**vBo**N) to the exact BoN (**Bo**N), i.e., applying BoN at inference time.

We implement and compare the following existing methods for language model alignment:

- **BoN-SFT:** Perhaps the most straightforward way to approximate BoN distribution is to fine-tune the model to maximize the likelihood of the samples taken with BoN algorithm. Unfortunately, we find that SFT is incapable of achieving a good trade-off between achieving high rewards and low KL divergence, see App. G for the experimental results.
- **PPO:** We use PPO to optimize the KL-constrained objective in Eq. (1). We use the default hyperparameters in `trlx` library (Havrilla et al., 2023) for fine-tuning with PPO.

  DPO. Direct preference optimization (DPO; Rafailov et al., 2023) is a popular alternative to RLHF that does not require training a reward model. Following DPO's experimental setup, we generate 6 reviews per prompt and use the resulting 12 pairwise comparisons per prompt to construct DPO's contrastive loss.[8]
- **BoNBoN:** Concurrent work (Gui et al., 2024) explores another approach to approximate BoN distribution. Assuming that the reference model distribution π_{ref} is continuous, Gui et al. (Theorem 3; 2024) prove that the expected difference between the relative likelihood, i.e., $\frac{\pi_{\text{bon}}(\cdot)}{\pi_{\text{ref}}(\cdot)}$, of the Best-of-$N$ response and the Worst-of-N response is $\frac{1}{2\beta} = \frac{1}{(N-1)\sum_{k=1}^{N-1} 1/k}$. They use this property to construct a loss function similar to that of IPO (Azar et al., 2023). Furthermore, they add another term to the loss function, which simply maximizes the likelihood of the Best-of-N response. The final loss function is a convex combination of the IPO-like loss and the negative log-likelihood loss, regulated by a hyperparameter α.[9]

We fine-tune models by varying the degree of regularization. For BoN approaches, that is achieved by varying N, and for DPO and PPO, we vary β.[10] Conveniently, N in vBoN is a hyperparameter,

[8] One could argue that DPO has a slight advantage over other methods in this setup since it has seen 6 unique generations per prompt during training, while the others only have seen one (or 2 with BoNBoN). Nevertheless, we observe that vBoN is more effective than DPO.

[9] Following the authors' recommendation, we set α so that both terms contribute equally to the final loss.

[10] See App. E for more details regarding the regularization hyperparameters.

meaning that we do *not* need to generate more samples from π_{ref} when we increase N. However, with BoN and BoNBoN methods, we need to increase the number of samples from the reference model as we increase N.

We generate movie reviews based on prompts from our test set using fine-tuned models and then measure three metrics: (i) KL divergence between the fine-tuned model and the reference model; (ii) win rate, defined as the percentage of times the fine-tuned model's generations receive higher rewards compared to the reference model's generations; and (iii) average rewards obtained by the fine-tuned model's sampled strings.

For the BoN method, we report the empirical upper bound of $\log N - \frac{N-1}{N}$ for KL divergence (Beirami et al., 2024; Mroueh, 2024) in our plots. Furthermore, the win rate of BoN over the reference model can be computed analytically and is equal to $\frac{N}{N+1}$.

We visualize the win rate vs. KL curves in Fig. 6a, and Fig. 6b the average rewards of generations under $\pi_{\boldsymbol{\theta}}$ vs. the KL divergence. As expected, BoN is the most effective approach; however, this comes at an extra inference cost that grows with N. We observe that among the fine-tuning methods, our variational approximation to BoN gets closest to the performance of BoN, as it appears more often on the Pareto front of the two curves compared to other methods. Notably, we observe that DPO performs better than PPO in terms of win rates but worse in terms of average rewards; this could be attributed to the contrastive nature of DPO's loss function.

5.1 Error in Estimating $\log \mathrm{F}(\cdot)$

We empirically quantify the error when estimating $\log \mathrm{F}(\cdot)$ with a finite number of i.i.d samples from π_{ref}. To get a better intuition on the error of our estimators, in Fig. 3, we visualize the estimators for 3 different prompts: one adversarial prompt (left plot), where the prompt itself has a negative sentiment, one neutral prompt (middle plot), and one prompt with a positive sentiment (right plot). We vary the number of Monte Carlo samples from 10 to 600. We observe that for all the 3 prompts, the estimated CDF hardly changes after 200 samples. When using the adversarial prompt, the reward distribution is negatively peaked, and the estimated CDF does not change after taking only 100 samples.

We then quantify the change in the estimator by performing a two-sample Kolmogorov–Smirnov test (Hodges, 1958). This test measures the closeness of two empirical cumulative distribution functions. Concretely, the test statistic is

$$\sup_{\boldsymbol{y} \in \Sigma^*} \left| \widehat{\mathrm{F}}_{M_1}\big(r(\boldsymbol{y})\big) - \widehat{\mathrm{F}}_{M_2}\big(r(\boldsymbol{y})\big) \right|, \tag{11}$$

where $\widehat{\mathrm{F}}_{M_1}, \widehat{\mathrm{F}}_{M_2}$ are estimated CDFs from M_1 and M_2 samples respectively. The statistics show the magnitude of the difference between the two empirical distributions of samples. The null hypothesis is that the two distributions are identical.

In Tab. 1, for each sample size M, we compare the estimated CDF with M samples to the estimated CDF with 600 samples. If the two distributions are identical according to the test, we can reliably use the M sample to estimate the CDF. We report the number of prompts (out of 5000 prompts) for which we reject the null hypothesis, meaning that the distributions are not identical. Furthermore, for those prompts, we report the average test statistics and p-values. In general, for very few prompts, the null hypothesis is rejected. Moreover, with 250 samples, the estimated CDFs are identical to the estimated CDF with 600 samples for all prompts.

Table 1: Measuring the estimation error with increasing the sample size. After 250 samples, the estimated CDF is unchanged for all the prompts.

M	Rejection rate	Test statistics	p-value
5	6.14%	0.63	0.02
20	4.02%	0.33	0.03
100	1.14%	0.17	0.02
200	0.06%	0.12	0.02
250	0	-	-

6 Summarization

We further employ variational BoN in a summarization task, where the goal is to generate summaries that align with human preferences. The reference model, π_{ref}, is a `pythia-2.8B` model fine-tuned on

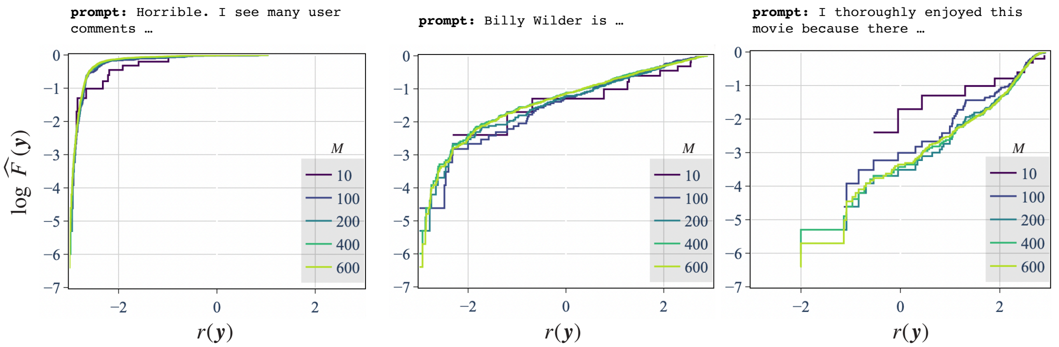


Figure 3: Estimates of $\log \mathrm{F}(\cdot)$ with increasing the number of Monte Carlo samples. We test an adversarial prompt (left plot), a neutral prompt (middle plot), and a prompt with a positive sentiment (right plot). Overall, we hardly see any difference between the estimates after taking 200 samples. For the adversarial prompt, the distribution of rewards is peaked, and we do not see any changes in our estimator after taking only 100 samples.

human-written summaries of Reddit posts Stiennon et al. (2020).[11] We use SFT to refer to this model in the plots. We use two separate reward models: a `pythia-2.8B`[12] reward model for fine-tuning and a larger `pythia-6.9B`[13] model for evaluation.

Dataset. To evaluate the generalization ability of the aligned models on out-of-distribution data, we fine-tune the models using only posts from the `relationship` and `relationship_advice` subreddits of the `Reddit TL;DR` (Stiennon et al., 2020) dataset. We then assess the models' performance on the two types of data by dividing the the test set into two equally-sized groups: in-distribution Reddit posts from the `relationship` and `relationship_advice` subreddits, and out-of-distribution posts from the rest of the subreddits. We visualize the performance of methods on in-distribution data with a solid trace and on out-of-distribution data with a dashed trace.

Experimental setup. We fine-tune the model with both the KL-constrained RL objective and vBoN objective for 10000 episodes. Similar to the previous experiment, we use 200 samples to estimate $\log \mathrm{F}(\cdot)$ values. To create a smooth and continuous reward function, we further fit an exponential curve[14] to the estimates. We set $N = 100$ for BoN and vBoN methods and the equivalent value of $\beta = 0.05$ for the KL-constrained RL objective. We closely follow Huang et al. (2024) for setting the hyperparameters of the PPO algorithm; please refer to App. E for more experimental details. After fine-tuning, we sample from the aligned models with different sampling temperatures $t \in [0.25, 0.5, 0.75, 1.]$, each with 3 different random seeds.

Win rates. In Fig. 4a we visualize the average and standard deviation of win rates compared against the samples from the SFT model. Our results show that all alignment methods outperform SFT, achieving win rates greater than 0.5 across all temperature settings. Notably, BoN achieves the highest win rates, which is consistent with findings from previous studies (Rafailov et al., 2023). We do not observe any significant differences between BoN performance on in-distribution (solid trace) and out-of-distribution data,[15] which is expected as BoN is an alignment-via-inference method. Similarly, we do not observe significant differences between in- and out-of-distribution performance of PPO and vBoN, indicating that both methods can generalize effectively in this experimental setup.

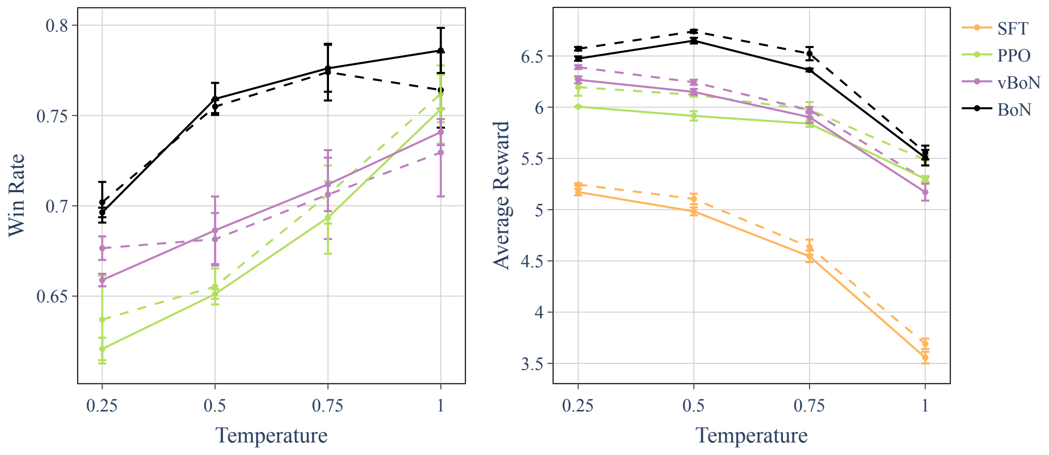
[11] Specifically, we use `https://huggingface.co/cleanrl/EleutherAI_pythia-2.8b-deduped__sft__tldr`.

[12] Specifically, we use `https://huggingface.co/cleanrl/EleutherAI_pythia-2.8b-deduped__reward__tldr`.

[13] Specifically, we use `https://huggingface.co/cleanrl/EleutherAI_pythia-6.9b-deduped__reward__tldr`.

[14] We fit an exponential function of the form $f(x) = -a\exp(-bx)$ to the data using non-linear least squares method.

[15] The difference between the two data distributions becomes more apparent at temperature 1, potentially due to increased sample diversity in this setting.



(a) Comparing the win rates of alignment methods against samples from the π_{ref}. The win rates are calculated based on the rewards obtained from the evaluator model. vBoN achieves closer results to BoN compared to PPO.

(b) Comparing the average rewards of alignment methods obtained from the evaluator reward model. BoN outperforms other alignment methods, and vBoN achieves closer results to BoN compared to PPO.

Figure 4: Performance of different alignment methods on the summarization task. Solid traces show the performance on in-distribution Reddit posts, while dashed lines demonstrate the out-of-distribution performance. Overall, BoN is the most effective approach in achieving high win rates and average rewards across all sampling temperatures. Our variational approximation to BoN (vBoN) gets closest to the performance of BoN, while being significantly cheaper at inference time.

Importantly, while PPO and vBoN perform comparably at higher temperatures, vBoN significantly outperforms PPO at lower temperatures (0.25 and 0.5).

Average rewards. In Fig. 4b, we measure the average rewards across different temperatures. As the temperature increases, the average reward decreases consistently across all methods. This trend is also evident in the qualitative analysis in App. H, where we show sampled summaries at different temperatures. Generally, the average reward results align with the win-rate trends, and we observe that vBoN achieves significantly higher rewards compared to PPO at lower temperatures. In Tab. 2 we show an example of summaries generated from the fine-tuned models with their associated reward values.

7 Related Work

Best-of-N. BoN is a straightforward alignment-via-inference algorithm to optimize the output of the language model using a trained reward model (Charniak & Johnson, 2005; Stiennon et al., 2020). Despite its simplicity, BoN performs comparably or even better than other alignment methods, such as RLHF and direct preference optimization (Nakano et al., 2022; Gao et al., 2023; Rafailov et al., 2023). However, as noted by Stiennon et al. (2020), BoN is an inefficient algorithm due to the reduced throughput at inference time.

Applications. BoN has been applied successfully at various stages of the development of language models. Touvron et al. (2023); Dong et al. (2023) employ iterative supervised fine-tuning on the outputs of the BoN algorithm to clone its behavior in the model. Pace et al. (2024) leverage BoN to enhance reward modeling by training the reward model on both the best and worst responses. Additionally, Brown et al. (2024); Snell et al. (2024) explore the scaling laws for alignment-via-inference methods and demonstrate how to utilize the limited inference budget to achieve the alignment.

Best-of-N as an alignment-via-fine-tuning method. Two concurrent efforts to ours have also attempted to convert BoN to an alignment-via-fine-tuning method. First, Gui et al. (2024) approxi-

Table 2: An example of summaries sampled at temperature 0.5 and their corresponding reward obtained from the evaluator reward model.

Content	Reward
SUBREDDIT: r/relationship_advice TITLE: Stuck in a rut and in need of advice/inspiration! POST: My boyfriend and I have been together for 3 years, and living together for 2. I'm quite the homebody, and when we first met, he was very outgoing and loved partying and socialising (although he was a student at the time). We're both working now, and most nights we find ourselves doing the same things: watching series (luckily we enjoy the same shows), playing Minecraft or playing various board games. We're tired after work, and can't bring ourselves to leave the house. The weekend is much the same – lots of sleep, or sitting around staring at one screen or another. We do party occasionally (we'll head to a pub once every few months) and there are a few mutual friends we enjoy spending time with, but I worry that we've become stuck in our boring ways. I really enjoy our lifestyle, and would be quite happy to never leave the house again, but I'm starting to feel guilty for turning him into a 50 year-old when he's only 24. Any ideas for shaking things up a little? Bear in mind that we live in a small town in South Africa, and neither of us has a car.	-
SFT: I'm stuck in a rut, and need to shake things up to see if it'll work out. Any advice?	3.08
PPO: In need of inspiration to break out of rut and live life fully! Any ideas welcome!	4.59
vBoN: Been happily living together for 2yr+, feeling bored after work regularly, looking for ideas to spice things up!	6.79
BoN: My boyfriend and I have been together for 3 years, and are both working full time. We spend most of our time in the house, and have become boring. What can we do to shake things up?	9.18

mate the BoN by maximizing the likelihood of the Best-of-N response and adjusting the relative likelihood of the Best-of-N and the Worst-of-N response. Second, Sessa et al. (2024) similar to ours uses reinforcement learning to minimize the distance between the language model and the BoN policy. Different from ours, and to reduce the fine-tuning time, the authors use a crude estimation of $\log F$ and approximate the distance to Best-of-N by iteratively distilling the Best-of-2 model as a moving anchor.

8 Conclusion

Motivated by the effectiveness of the BoN algorithm, we formally derive a variational approximation to the distribution induced by BoN algorithm via fine-tuning language models. Our analysis highlights the similarities and distinctions between the variational BoN objective and the KL-constrained RL objectives. Our empirical findings reveal that models fine-tuned using the variational approximation to BoN not only attain high reward values but also maintain proximity to the reference models. Crucially, inference on the fine-tuned models with the vBoN objective remains as cost-effective as inference on the original reference model.

Acknowledgements

We thank Ahmad Beirami for the fruitful discussion in the early stages of this project. Afra Amini is supported by the ETH AI Center doctoral fellowship.

References

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *Computing Research Repository*, arXiv:2310.12036, 2023. URL https://arxiv.org/abs/2310.12036.

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

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *Computing Research Repository*, arXiv:2407.21787, 2024. URL https://arxiv.org/abs/2407.21787.

George Casella and Roger L. Berger. *Statistical Inference*. Chapman and Hall/CRC, Pacific Grove, CA, 2nd edition, 2001. ISBN 9781032593036. URL https://www.routledge.com/Statistical-Inference/Casella-Berger/p/book/9781032593036.

Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In Kevin Knight, Hwee Tou Ng, and Kemal Oflazer (eds.), *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 173–180, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219862. URL https://aclanthology.org/P05-1022.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=m7p5O7zblY.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/gao23h.html.

Gemini, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam

Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze

Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kepa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei ”Louis” Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam

Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang,

Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models. Technical report, Google, 2024. URL https://arxiv.org/pdf/2312.11805.

Lin Gui, Cristina Gârbacea, and Victor Veitch. BoNBoN alignment for large language models and the sweetness of best-of-n sampling. *Computing Research Repository*, arXiv:2406.00832, 2024. URL https://arxiv.org/pdf/2406.00832.

Alexander Havrilla, Maksym Zhuravinskyi, Duy Phung, Aman Tiwari, Jonathan Tow, Stella Biderman, Quentin Anthony, and Louis Castricato. trlX: A framework for large scale reinforcement learning from human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8578–8595, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.530. URL https://aclanthology.org/2023.emnlp-main.530.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8. URL https://dl.acm.org/doi/pdf/10.5555/3504035.3504427.

Joseph L. Hodges. The significance probability of the smirnov two-sample test. *Arkiv för Matematik*, 3:469–486, 1958. URL https://api.semanticscholar.org/CorpusID:121451525.

Shengyi Huang, Michael Noukhovitch, Arian Hosseini, Kashif Rasul, Weixun Wang, and Lewis Tunstall. The N+ implementation details of RLHF with PPO: A case study on TL;DR summarization. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kHO2ZTa8e3.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *Computing Research Repository*, arXiv:1805.00909, 2018. URL https://arxiv.org/pdf/1805.00909.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015.

Youssef Mroueh. Information theoretic guarantees for policy alignment in large language models. *Computing Research Repository*, arXiv:2406.05883, 2024. URL https://arxiv.org/abs/2406.05883.

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. Controlled decoding from language models. In *Proceedings of The 41st International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2024. URL https://arxiv.org/pdf/2310.17022.

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. WebGPT: Browser-assisted question-answering with human feedback. *Computing Research Repository*, arXiv:2112.09332, 2022. URL https://arxiv.org/pdf/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, Mo 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, 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 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 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. GPT-4 technical report. Technical report, OpenAI, 2023. URL https://cdn.openai.com/papers/gpt-4.pdf.

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. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Alizée Pace, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. West-of-n: Synthetic preference generation for improved reward modeling. *Computing Research Repository*, arXiv:2401.12086, 2024. URL https://arxiv.org/abs/2401.12086.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. URL https://arxiv.org/pdf/2305.18290.pdf.

Tom Schaul, Georg Ostrovski, Iurii Kemaev, and Diana Borsa. Return-based scaling: Yet another normalisation trick for deep rl. *Computing Research Repository*, arXiv:2105.05347, 2021. URL https://arxiv.org/abs/2105.05347.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *Computing Research Repository*, arXiv:1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shariari, Sarah Perrin, Abe Friesen, Geoffrey Cideron, Sertan Girgin, Piotr Stanczyk, Andrea Michi, Danila Sinopalnikov, Sabela Ramos, Amélie Héliou, Aliaksei Severyn, Matt Hoffman, Nikola Momchev, and Olivier Bachem. Bond: Aligning llms with best-of-n distillation. *Computing Research Repository*, arXiv:2401.12086, 2024. URL https://arxiv.org/abs/2401.12086.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Computing Research Repository*, arXiv:2408.03314, 2024. URL https://arxiv.org/abs/2408.03314.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. Technical report, Meta, 2023. URL https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models/.

Joy Qiping Yang, Salman Salamatian, Ziteng Sun, Ananda Theertha Suresh, and Ahmad Beirami. Asymptotics of language model alignment. *Computing Research Repository*, arXiv:2404.01730, 2024. URL https://arxiv.org/pdf/2404.01730.

Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535, Online, June 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.naacl-main.276.

A Proof of Prop. 1

Proposition 1. *Suppose $r \colon \Sigma^* \to \mathbb{R}$ is a one-to-one mapping. Then, the probability that a string $\boldsymbol{y} \sim \pi_{bon}$ is given by*

$$\pi_{bon}(\boldsymbol{y}) = \sum_{i=1}^{N} \binom{N}{i} \mathrm{F}\big(r(\boldsymbol{y})\big)^{N-i} \pi_{\mathrm{ref}}(\boldsymbol{y})^{i}, \tag{4}$$

where we define $\mathrm{F}(R)$ *as*

$$\mathrm{F}(R) \overset{\text{def}}{=} \mathbb{P}\left(r(\boldsymbol{y}) < R\right). \tag{5}$$

Proof. The proof follows Casella & Berger (2001, Theorem 5.4.3). To compute $\pi_{\mathrm{bon}}(\boldsymbol{y})$, we first define two events: (i) the event that all N samples have rewards less than or equal to $r(\boldsymbol{y})$, and (ii) the event that all N samples have rewards less than $r(\boldsymbol{y})$. The probability of those events is as follows:[16]

$$p_1(\boldsymbol{y}) \overset{\text{def}}{=} \mathbb{P}(\text{all } N \text{ samples have rewards} \leq r(\boldsymbol{y})) = \Big(\mathrm{F}\big(r(\boldsymbol{y})\big) + \pi_{\mathrm{ref}}(\boldsymbol{y})\Big)^{N} \tag{12a}$$

$$p_2(\boldsymbol{y}) \overset{\text{def}}{=} \mathbb{P}(\text{all } N \text{ samples have rewards} < r(\boldsymbol{y})) = \mathrm{F}\big(r(\boldsymbol{y})\big)^{N}. \tag{12b}$$

Note that for Eq. (15a) to hold, we need the assumption that the reward function is a one-to-one mapping.[17] Furthermore, given this assumption, $\pi_{\mathrm{bon}}(\boldsymbol{y})$ is the probability that *at least* one of the sampled strings out of N samples have the reward exactly equal to $r(\boldsymbol{y})$ and the rest of the samples have rewards less than or equal to $r(\boldsymbol{y})$. Given how we defined p_1 and p_2, we have $\pi_{\mathrm{bon}}(\boldsymbol{y}) = p_1(\boldsymbol{y}) - p_2(\boldsymbol{y})$.

$$\pi_{\mathrm{bon}}(\boldsymbol{y}) = \Big(\mathrm{F}\big(r(\boldsymbol{y})\big) + \pi_{\mathrm{ref}}(\boldsymbol{y})\Big)^{N} - \mathrm{F}\big(r(\boldsymbol{y})\big)^{N} = \sum_{i=1}^{N} \binom{N}{i} \mathrm{F}\big(r(\boldsymbol{y})\big)^{N-i} \pi_{\mathrm{ref}}(\boldsymbol{y})^{i}. \tag{13}$$

■

B Strategies for Non-Injective Reward Functions

If the reward function is not injective, we need a tie-breaking strategy for the BoN algorithm. We formalize this as defining a total order $\prec_r$ on Σ^* as follows: for any two strings $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, if $r(\boldsymbol{y}_1) < r(\boldsymbol{y}_2)$ then we have $\boldsymbol{y}_1 \prec_r \boldsymbol{y}_2$. If $r(\boldsymbol{y}_1) = r(\boldsymbol{y}_2)$ then $\boldsymbol{y}_1 \prec_r \boldsymbol{y}_2$ only if $\boldsymbol{y}_1 \prec \boldsymbol{y}_2$, where $\prec$ is some arbitrary but fixed total order, e.g., lexicographic order. Therefore, we define $\mathrm{F}(\boldsymbol{y})$ as

$$\mathrm{F}(\boldsymbol{y}) \overset{\text{def}}{=} \mathbb{P}\big(\boldsymbol{y}' \prec_r \boldsymbol{y}\big). \tag{14}$$

We then need to define the two events and their probabilities, p_1 and p_2, given this total order on strings, as follows:

$$p_1(\boldsymbol{y}) \overset{\text{def}}{=} \mathbb{P}(\text{all } N \text{ samples are } \preceq_r \boldsymbol{y}) = \Big(\mathrm{F}\big(\boldsymbol{y}\big) + \pi_{\mathrm{ref}}(\boldsymbol{y})\Big)^{N} \tag{15a}$$

$$p_2(\boldsymbol{y}) \overset{\text{def}}{=} \mathbb{P}(\text{all } N \text{ samples are } \prec_r \boldsymbol{y}) = \mathrm{F}\big(\boldsymbol{y}\big)^{N} \tag{15b}$$

The rest of the proof is the same as with the one-to-one reward functions.

C Proof of Thm. 2

Theorem 2. *Let* $\alpha = \frac{(N+2)(N-1)}{2}$, $\beta = \frac{N(N+1)}{2}$, *and* $\gamma = \frac{N(N-1)}{2}$. *Then, we have* $\mathcal{J}^{\mathrm{vBoN}}(\boldsymbol{\theta}) \geq L_1(\boldsymbol{\theta})$, *where we further define*

$$L_1(\boldsymbol{\theta}) \overset{\text{def}}{=} \gamma \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \pi_{\boldsymbol{\theta}}} \Big[\log \mathrm{F}\big(r(\boldsymbol{y})\big)\Big] - \alpha \mathrm{H}\big(\pi_{\boldsymbol{\theta}}\big) - \beta D_{\mathrm{KL}}\big(\pi_{\boldsymbol{\theta}} \mid\mid \pi_{\mathrm{ref}}\big). \tag{9}$$

[16] The PMF of BoN is also derived by Beirami et al. (Lemma 1; 2024). In their notation, $p_1 = \mathcal{F}$ and $p_2 = \mathcal{F}^{-1}$.

[17] If the reward function is not a one-to-one mapping, we need to devise a tie-breaking strategy. See App. B for further discussion.

Proof.

$$
\begin{aligned}
D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \mid\mid \pi_{\text{bon}}\big) &= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \log \pi_{\text{bon}}(\boldsymbol{y})\Big] && \text{(16a)}\\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \log \sum_{i=1}^{N}\binom{N}{i}\mathrm{F}\big(r(\boldsymbol{y})\big)^{N-i}\pi_{\text{ref}}(\boldsymbol{y})^{i}\Big] && \text{(16b)}\\
&\le \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \sum_{i=1}^{N}\log\binom{N}{i}\mathrm{F}\big(r(\boldsymbol{y})\big)^{N-i}\pi_{\text{ref}}(\boldsymbol{y})^{i}\Big] && \text{(16c)}\\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \sum_{i=1}^{N}\log\binom{N}{i} - \sum_{i=1}^{N}\log \mathrm{F}\big(r(\boldsymbol{y})\big)^{N-i} - \sum_{i=1}^{N}\log\pi_{\text{ref}}(\boldsymbol{y})^{i}\Big] && \text{(16d)}\\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \sum_{i=1}^{N}\log\binom{N}{i} - \log \mathrm{F}\big(r(\boldsymbol{y})\big)\sum_{i=1}^{N}(N-i) - \log\pi_{\text{ref}}(\boldsymbol{y})\sum_{i=1}^{N} i\Big] && \text{(16e)}\\
&\le \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \frac{N(N-1)}{2}\log \mathrm{F}\big(r(\boldsymbol{y})\big) - \frac{N(N+1)}{2}\log\pi_{\text{ref}}(\boldsymbol{y})\Big] && \text{(16f)}\\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \frac{N(N+1)}{2}\log\pi_{\text{ref}}(\boldsymbol{y}) - \frac{N(N-1)}{2}\log \mathrm{F}\big(r(\boldsymbol{y})\big)\Big] && \text{(16g)}\\
&= \frac{N(N+1)}{2}D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \mid\mid \pi_{\text{ref}}\big) + \mathop{\mathbb{E}}_{\pi_{\boldsymbol{\theta}}}\Big[\frac{-(N+2)(N-1)}{2}\log\pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \frac{N(N-1)}{2}\log \mathrm{F}\big(r(\boldsymbol{y})\big)\Big] && \text{(16h)}\\
&= \frac{N(N+1)}{2}D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \mid\mid \pi_{\text{ref}}\big) + \frac{(N+2)(N-1)}{2}\mathrm{H}\big(\pi_{\boldsymbol{\theta}}\big) - \mathop{\mathbb{E}}_{\pi_{\boldsymbol{\theta}}}\Big[\frac{N(N-1)}{2}\log \mathrm{F}\big(r(\boldsymbol{y})\big)\Big] && \text{(16i)}
\end{aligned}
$$

In Eq. (16c), because $-\log(x)$ is convex for $x \ge 0$, we applied Jensen's inequality to obtain the upper bound. Abstracting away from the three multiplicative factors, naming them γ, α and β, we end up with the following function

$$
\mathcal{J}^{\text{vBoN}}(\boldsymbol{\theta}) = -D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \mid\mid \pi_{\text{bon}}\big) \ge \gamma \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\log \mathrm{F}\big(r(\boldsymbol{y})\big) - \alpha\mathrm{H}(\pi_{\boldsymbol{\theta}}) - \beta D_{\text{KL}}\left(\pi_{\boldsymbol{\theta}} \mid\mid \pi_{\text{ref}}\right), \tag{17}
$$

which is a bound for some settings of γ, α and β. ■

D Proof of Thm. 3

Theorem 3. *We have $\mathcal{J}^{\text{vBoN}}(\boldsymbol{\theta}) \ge L_2(\boldsymbol{\theta}) \ge L_1(\boldsymbol{\theta})$, where*

$$
L_2(\boldsymbol{\theta}) \overset{\text{def}}{=} (N-1)\mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \mathrm{F}\big(r(\boldsymbol{y})\big)\Big] - D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \mid\mid \pi_{\text{ref}}\big). \tag{10}
$$

Proof. First, we prove $\mathcal{J}^{\text{vBoN}}(\boldsymbol{\theta}) \ge L_2(\boldsymbol{\theta})$.

$$
\begin{aligned}
D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \mid\mid \pi_{\text{bon}}\big) &= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \log \pi_{\text{bon}}(\boldsymbol{y})\Big] && \text{(18a)}\\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \log \sum_{i=1}^{N}\binom{N}{i}\mathrm{F}\big(r(\boldsymbol{y})\big)^{N-i}\pi_{\text{ref}}(\boldsymbol{y})^{i}\Big] && \text{(18b)}\\
&\le \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \log \sum_{i=1}^{N=1}\binom{N}{i}\mathrm{F}\big(r(\boldsymbol{y})\big)^{N-i}\pi_{\text{ref}}(\boldsymbol{y})^{i}\Big] && \text{(18c)}\\
&\le \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \log N\,\mathrm{F}\big(r(\boldsymbol{y})\big)^{N-1}\pi_{\text{ref}}(\boldsymbol{y})^{1}\Big] && \text{(18d)}\\
&\le \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \log \mathrm{F}\big(r(\boldsymbol{y})\big)^{N-1}\pi_{\text{ref}}(\boldsymbol{y})\Big] && \text{(18e)}\\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \log\pi_{\text{ref}}(\boldsymbol{y}) - (N-1)\log \mathrm{F}\big(r(\boldsymbol{y})\big)\Big] && \text{(18f)}\\
&= D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \mid\mid \pi_{\text{ref}}\big) - (N-1)\mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\Big[\log \mathrm{F}\big(r(\boldsymbol{y})\big)\Big] \overset{\text{def}}{=} -L_2(\boldsymbol{\theta}). && \text{(18g)}
\end{aligned}
$$

Hypterparameter	**Value**
Episodes	10000
Optimizer	AdamW ($\epsilon = 1e-5$, $\texttt{lr}= 3e-6$)
Scheduler	Linear
Batch Size	32
β (Both for vBoN and KL-constrained RL objective)	0.05
γ (Discount Factor)	1
λ (for GAE)	0.95
Number of PPO Update Iteration Per Epoch	4
PPO's Policy Clipping Coefficient	0.2
Value Clipping Coefficient	0.2
Value Function Coefficient	0.2
Value Function Loss Clipping	True
Sampling Temperature	0.7

The inequality in Eq. (18c) stems from the fact that we drop positive terms in the summation and only keep the first term. Therefore, the lower bound for our objective is:

$$\mathcal{J}^{\text{vBoN}}(\boldsymbol{\theta}) = -D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \,||\, \pi_{\text{bon}}\big) \geq (N-1) \underset{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}{\mathbb{E}}\Big[\log \mathrm{F}\big(r(\boldsymbol{y})\big)\Big] - D_{\text{KL}}\big(\pi_{\boldsymbol{\theta}} \,||\, \pi_{\text{ref}}\big). \tag{19}$$

Next, we prove $L_1(\boldsymbol{\theta}) \leq L_2(\boldsymbol{\theta})$. According to Eq. (16f), we have:

$$\begin{aligned}
-L_1(\boldsymbol{\theta}) &\geq \underset{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}{\mathbb{E}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \sum_{i=1}^{N} \log \mathrm{F}\big(r(\boldsymbol{y})\big)^{N-i}\pi_{\text{ref}}(\boldsymbol{y})^{i}\Big] && \text{(20a)}\\
&\geq \underset{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}{\mathbb{E}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \sum_{i=1}^{N=1} \log \mathrm{F}\big(r(\boldsymbol{y})\big)^{N-i}\pi_{\text{ref}}(\boldsymbol{y})^{i}\Big] && \text{(20b)}\\
&= \underset{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}{\mathbb{E}}\Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) - \log \mathrm{F}\big(r(\boldsymbol{y})\big)^{N-1}\pi_{\text{ref}}(\boldsymbol{y})\Big] = -L_2(\boldsymbol{\theta}). && \text{(20c)}
\end{aligned}$$

■

E Experimental Details

Hyperparameter sweep in the sentiment experiment. To visualize the trade-off between the expected rewards and KL divergence, we vary the degree of the visualization using the following hyperparameters for each method:

- **BoN-SFT**: $N \in [10, 50, 90, 130, 170, 210, 250, 290, 330, 370, 410, 450, 490, 530, 570, 600]$ with 2 different seeds, resulting in 32 runs.
- **PPO**: $\beta \in [0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1., 2., 3., 4., 5.]$ with 2 different seeds, resulting in 32 runs.
- **DPO**: $\beta \in [0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 1., 2., 3., 4., 5.]$ with 3 different seeds, resulting in 33 runs.
- **BoNBoN** and **vBoN**: $N \in [1, 2, 3, 4, 8, 16, 32, 64, 128, 256, 512]$ with 3 different seeds, resulting in 33 runs.
- **vBoN** with L_2 bound: $\beta \in [0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1., 2., 3., 4., 5.]$ with 2 different seeds, resulting in 32 runs. Note that comparing Eq. (6) and Eq. (1), we have $N = \frac{1}{\beta} + 1$.

PPO Hyperparameters. In App. E, we include the hyperparameters used with the PPO algorithm for the summarization experiment.

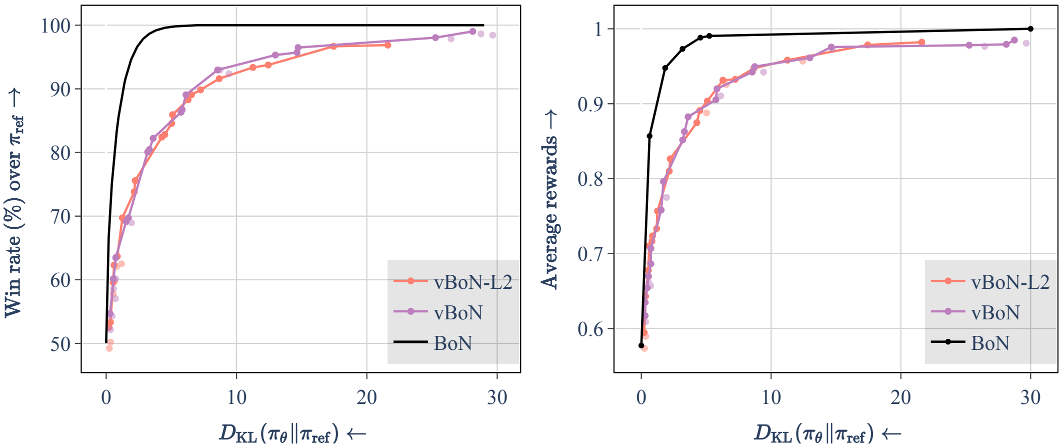
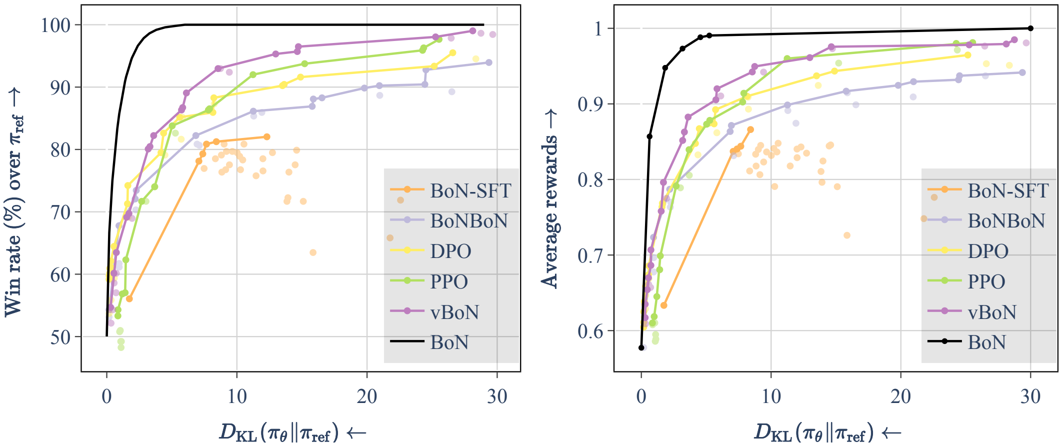


Figure 5: Comparing models trained with the vBoN objective and its lower bound (L_2). We observe that the performance of the two methods is very close to each other.



(a) 4% of points on Pareto front belong to BoNBoN, 4% to PPO, 42% to DPO, and 50% to vBoN.

(b) 7% of points on Pareto from belong to BoNBoN, 10% DPO, 33% PPO, and 50% vBoN.

Figure 6: Steering generated movie reviews towards positive sentiment. Points that are not on the Pareto front have lower opacity.

F Comparing the vBoN Objective and L_2 Lower Bound

We compare the performance of models fine-tuned with the vBoN objective and its lower bound (L_2) in Fig. 5. We observe that the performance of the models is very close to each other.

G Additional Experiments with BoN-SFT

We further experiment with training with maximum likelihood objective on BoN generations when varying N. The results are depicted in Fig. 6. We observe that BoN diverges too much from the reference model compared to other fine-tuning methods for alignment.

H Qualitative Results

Table 3: An example of summaries sampled at temperature 0.25 and their corresponding reward obtained from the evaluator reward model.

Content	Reward
SUBREDDIT: r/relationship_advice TITLE: Stuck in a rut and in need of advice/inspiration! POST: My boyfriend and I have been together for 3 years, and living together for 2. I'm quite the homebody, and when we first met, he was very outgoing and loved partying and socialising (although he was a student at the time). We're both working now, and most nights we find ourselves doing the same things: watching series (luckily we enjoy the same shows), playing Minecraft or playing various board games. We're tired after work, and can't bring ourselves to leave the house. The weekend is much the same – lots of sleep, or sitting around staring at one screen or another. We do party occasionally (we'll head to a pub once every few months) and there are a few mutual friends we enjoy spending time with, but I worry that we've become stuck in our boring ways. I really enjoy our lifestyle, and would be quite happy to never leave the house again, but I'm starting to feel guilty for turning him into a 50 year-old when he's only 24. Any ideas for shaking things up a little? Bear in mind that we live in a small town in South Africa, and neither of us has a car.	-
SFT: I'm stuck in a rut with my boyfriend, and I'd like to shake things up a little. Any ideas?	5.83
PPO: In need of inspiration for moving past boring routines and becoming active and open minded individuals. Any ideas welcome!	4.88
vBoN: Been dating for three years and living together for two. Feeling stuck in rut after working hard all week. Looking for ideas to shake things up!	7.39
BoN: Been together for 3 years, and living together for 2. We're both working now, and find ourselves doing the same things: watching series, playing games, or having sex. We're tired after work, and can't bring ourselves to leave	7.14

Table 4: An example of summaries sampled at temperature 0.5 and their corresponding reward obtained from the evaluator reward model.

Content	Reward
SUBREDDIT: r/relationship_advice TITLE: Stuck in a rut and in need of advice/inspiration! POST: My boyfriend and I have been together for 3 years, and living together for 2. I'm quite the homebody, and when we first met, he was very outgoing and loved partying and socialising (although he was a student at the time). We're both working now, and most nights we find ourselves doing the same things: watching series (luckily we enjoy the same shows), playing Minecraft or playing various board games. We're tired after work, and can't bring ourselves to leave the house. The weekend is much the same – lots of sleep, or sitting around staring at one screen or another. We do party occasionally (we'll head to a pub once every few months) and there are a few mutual friends we enjoy spending time with, but I worry that we've become stuck in our boring ways. I really enjoy our lifestyle, and would be quite happy to never leave the house again, but I'm starting to feel guilty for turning him into a 50 year-old when he's only 24. Any ideas for shaking things up a little? Bear in mind that we live in a small town in South Africa, and neither of us has a car.	-
SFT: I'm stuck in a rut, and need to shake things up to see if it'll work out. Any advice?	3.08
PPO: In need of inspiration to break out of rut and live life fully! Any ideas welcome!	4.59
vBoN: Been happily living together for 2yr+, feeling bored after work regularly, looking for ideas to spice things up!	6.79
BoN: My boyfriend and I have been together for 3 years, and are both working full time. We spend most of our time in the house, and have become boring. What can we do to shake things up?	9.18

Table 5: An example of summaries sampled at temperature 0.75 and their corresponding reward obtained from the evaluator reward model.

Content	Reward
SUBREDDIT: r/relationship_advice TITLE: Stuck in a rut and in need of advice/inspiration! POST: My boyfriend and I have been together for 3 years, and living together for 2. I'm quite the homebody, and when we first met, he was very outgoing and loved partying and socialising (although he was a student at the time). We're both working now, and most nights we find ourselves doing the same things: watching series (luckily we enjoy the same shows), playing Minecraft or playing various board games. We're tired after work, and can't bring ourselves to leave the house. The weekend is much the same – lots of sleep, or sitting around staring at one screen or another. We do party occasionally (we'll head to a pub once every few months) and there are a few mutual friends we enjoy spending time with, but I worry that we've become stuck in our boring ways. I really enjoy our lifestyle, and would be quite happy to never leave the house again, but I'm starting to feel guilty for turning him into a 50 year-old when he's only 24. Any ideas for shaking things up a little? Bear in mind that we live in a small town in South Africa, and neither of us has a car.	-
SFT: Boyfriend and I are stuck in a rut. We don't do anything fun, and I'm starting to feel guilty for not even trying to get away from it all.	4.41
PPO: In rutting boredom with boring life; any ideas for shaking things up a little? Any ideas for improving relationships?	4.76
vBoN: Been happily cohabiting for three years with boyfriend of two years. Feeling guilty for turning him into an old man! Any ideas for shaking things up?	6.02
BoN: Boyfriend and I are a little boring, and want to shake things up so we can go out and have fun but we're too stuck on the idea of staying in all day.	5.47

Table 6: An example of summaries sampled at temperature $1.$ and their corresponding reward obtained from the evaluator reward model.

Content	Reward
SUBREDDIT: r/relationship_advice TITLE: Stuck in a rut and in need of advice/inspiration! POST: My boyfriend and I have been together for 3 years, and living together for 2. I'm quite the homebody, and when we first met, he was very outgoing and loved partying and socialising (although he was a student at the time). We're both working now, and most nights we find ourselves doing the same things: watching series (luckily we enjoy the same shows), playing Minecraft or playing various board games. We're tired after work, and can't bring ourselves to leave the house. The weekend is much the same – lots of sleep, or sitting around staring at one screen or another. We do party occasionally (we'll head to a pub once every few months) and there are a few mutual friends we enjoy spending time with, but I worry that we've become stuck in our boring ways. I really enjoy our lifestyle, and would be quite happy to never leave the house again, but I'm starting to feel guilty for turning him into a 50 year-old when he's only 24. Any ideas for shaking things up a little? Bear in mind that we live in a small town in South Africa, and neither of us has a car.	-
SFT: My boyfriend and I used to party all the time and are ending up reverting to the boring old lifestyle of a 24-year-old. How can we both break out of a rut and get some fun back in our relationship?	5.89
PPO: Taken classic boring suburban lifestyle and turned into boring 24yo. Looking for ideas for breaking out of rut and refreshing lifestyle. Any ideas welcome!	5.91
vBoN: boyfriend and I seem to have fallen into a rut of regularly doing mundane activities after working together non-stop for several months! ideas to spice things up?	6.57
BoN: in a relationship that's getting a bit stale, looking for some inspiration to make changes on a whim in hopes of rejuvenating it!	6.74