Theoretical Limits of Language Model Alignment

Lucas Monteiro Paes* Apple
Natalie Mackraz* Apple
Barry-John Theobald Apple
Federico Danieli Apple

{lucasmp, natalie_mackraz, barryjohn_theobald, f_danieli}@apple.com

Abstract

Language model (LM) alignment improves model outputs to reflect human preferences while preserving the capabilities of the base model. The most common alignment approaches are (i) reinforcement learning, which maximizes the expected reward under a KL divergence constraint, and (ii) best-of-N alignment, which selects the highest-reward output among N candidates. Despite their widespread use, the fundamental limits of reward improvement under a KL budget remain poorly understood. We characterize the limits of LM alignment by deriving the maximum *achievable* reward gain for a fixed KL budget. Our first result provides a closed-form expression for the optimal reward improvement, governed by a Jeffreys divergence term rather than the $\sqrt{\mathrm{KL}}$ used in prior analyses. We reformulate this expression as a covariance under the base model, yielding a practical estimator that predicts achievable alignment gains from base model samples alone. We extend our analysis to the proxy reward setting, showing that the gap between ideal and proxy alignment (*reward hacking*) grows exponentially with the magnitude of reward error and the inverse of the KL penalty. We then prove that reward ensembling mitigates reward hacking, providing a theoretical justification for this technique used in practice. Empirically, we compute the KL–reward Pareto frontier for two alignment tasks for LMs, safety and summarization, and show that best-of-N closely approaches the theoretical limit, while PPO, GRPO, and DPO remain substantially suboptimal. Our theoretical results shed light on several empirically observed phenomena in the alignment literature and suggest that algorithmic improvements are needed to achieve optimal alignment without high inference costs.

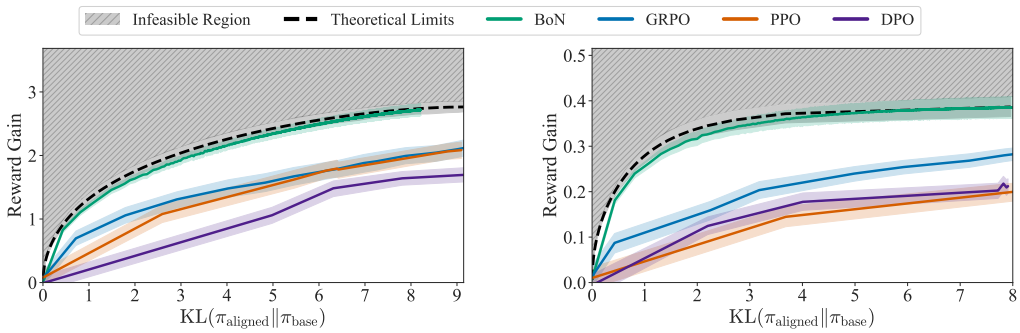
1 Introduction

Large Language Models (LLMs) perform complex tasks on behalf of users: such as solving mathematical problems [48, 16], moderating online content [33, 37], writing code [13, 4], and shopping [57, 31]. While these LLMs solve multiple problems without fine-tuning, their output may be toxic, deceiving, or otherwise harmful [28, 40, 10]. To ensure that models perform in a safe manner and avoid producing harmful outputs, they usually undergo a *safety alignment* step to ensure their outputs follow human preferences that ideally reflect ethical values and social norms [38, 6, 7, 47].

Approaches for aligning language models (LMs) include, (i) KL-regularized reinforcement learning (RL) [6, 47], (ii) direct preference optimization [42], (iii) best-of-N finetuning [50], among others [36, 59, 23, 53]. These methods aim to solve a KL-regularized RL problem shown in Equation (1). This approach begins by defining a reward function r which scores model outputs according to how well they align with user preferences. A base LLM $\pi_{\texttt{base}}$ is then fine-tuned to maximize the expected reward while penalizing (proportionally to a parameter λ) the KL-divergence from the base, leading to the aligned model $\pi_{r,\lambda}$ [14].

*Core contributors. Code available at: github.com/apple-aiml-research/ml-limits-of-alignment

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Zephyr 7B with Beavertails

(b) Pythia 1B with TL;DR

Figure 1: **Fundamental Limits of Alignment.** Reward gain vs. KL from $\pi_{\texttt{base}}$ for best-of-N (BoN), GRPO, PPO, DPO, and the theoretical limit computed using Def. 1 and Def. 2. Each PPO/GRPO/DPO point is a checkpoint and each BoN point is a value of N. In both cases, BoN closely track the theoretical limits, whereas GRPO, PPO, and DPO remain sub-optimal. The x-axis goes up to the maximum KL measured by the theoretical limit, which varies across tasks.

Despite the rapid proliferation of alignment algorithms, their theoretical limits remain largely unaddressed. Progress in alignment is typically measured by achieving higher expected reward for a fixed KL budget [21, 42, 36, 23], where the KL acts as a proxy for how much of the base model's general capabilities we retain after alignment [39]. While alignment methods continuously push the empirical Pareto frontier outward, the true theoretical limit (the maximum achievable reward for a given KL budget) remains generally unknown — see Sec. 2 for a discussion about existing results on the limits of alignment. Without characterizing this fundamental limit, it is impossible to determine whether current techniques are approaching optimality.

Fundamental Limits of Alignment. We characterize the fundamental limit[2] of LM alignment, shown in Fig. 1. First, we show that the reward gain after *optimal alignment* is given by $\lambda \texttt{JD}(\pi_{\texttt{base}}||\pi_{r,\lambda})$, where λ is the KL penalty used to restrict the KL (see Equation (1)) and $\texttt{JD}$ is the Jeffreys divergence given by: $\texttt{JD}(P||Q) \triangleq \texttt{KL}(P||Q) + \texttt{KL}(Q||P)$ [26]. This improves upon previous work that provided only an upper bound characterized by the square root of the KL [6, 21, 35]. While our characterization is exact, the limit expression depends on the aligned model $\pi_{r,\lambda}$, which is available only after *optimal alignment* — undermining its usefulness as a predictive tool. We resolve this circularity by showing that the Jeffreys divergence term can be rewritten as a covariance under $\pi_{\texttt{base}}$ alone, thus making it estimable before any post-training is performed. This provides a principled way to measure how close an alignment method comes to the optimal trade-off.

Proxy Rewards and Hacking. The limits of alignment rely on ideal (*gold*) reward models (r^*). In practice, rewards are learned from human annotations and are noisy approximations, called *proxy rewards* [21], denoted by $\tilde{r}$. We extend our characterization to proxy rewards and show that the gap in reward gain between alignment under the gold reward and under a proxy reward grows exponentially with the preference residual ($\tilde{r} - r^*$) and with the inverse of the KL penalty term (λ^{-1}). By isolating these exact dependencies, our formulation complements recent theoretical advances on the existence of reward hacking [2, 24, 32]. Together, these results *explain why aggressive alignment (smaller λ) using a proxy reward leads to hacking, as observed by Gao et al. [21, Figure 1].*

Reward Ensembling and Hacking. We further leverage our framework to justify the effectiveness of *reward ensembling* [18, 17, 58] as a practice to mitigate reward hacking. We show that, assuming unbiased proxy errors, aligning to an ensemble of n rewards converges to the performance of the gold-aligned model, thus providing a theoretical justification for reward ensembling.

Experimental Results. We evaluate our limits in two settings: safety alignment [27] and summarization from human feedback [47]. First, we empirically validate the estimators derived using our framework, observing fast convergence when predicting the expected reward gain directly from the base model. Second, we benchmark widely used alignment algorithms—specifically PPO [44], GRPO [45], DPO [42], and best-of-N sampling—against the theoretical limits.

Our **main contributions** include:

[2]We use the term "fundamental limit" to represent an algorithm-agnostic limit for a fixed reward and base LM.

- **Fundamental limit of alignment.** We prove that the optimal reward gain after alignment is exactly $\lambda\, \mathtt{JD}(\pi_{\mathtt{base}} \| \pi_{r,\lambda})$, where JD is the Jeffreys divergence—Thm. 1. This improves the previous scaling of $\sqrt{\mathtt{KL}}$ empirically observed in [6, 21] and proved to be an upper bound for reward gain by Mroueh and Nitsure [35].
- **Fundamental limit estimator.** We show that the optimal reward gain after alignment can be rewritten as a covariance term under the base model alone, making it computable *before alignment*—Thm. 2. We also derive an estimator of the KL between the aligned model and the base model in Prop. 2, allowing us to estimate the reward–KL Pareto frontier.
- **How Overoptimization leads to Reward Hacking.** We characterize how proxy rewards degrade alignment, showing that the post-alignment reward deviates from ideal when reward error grows and the KL penalty decreases—Prop. 3. This explains how aggressive alignment using a proxy reward leads to reward hacking, as observed by Gao et al. [21, Figure 1]
- **Why Reward Ensembling Improves Reward Hacking.** We leverage our limit covariance form to explain the impact of ensembling on reward hacking in Thm. 3. We show that "good" ensembles provably mitigate reward hacking, but when reward models share errors, the hacking persists—explaining the "herding" phenomenon observed by Eisenstein et al. [18].
- **Empirical verification & Optimality of best-of-N (BoN).** We validate our estimators and benchmark alignment methods against it (PPO, GRPO, DPO, and BoN), demonstrating that BoN closely matches our derived theoretical reward—KL Pareto frontier (Fig. 1). Showing why BoN was previously observed to be a strong baseline for LM alignment.

2 Background

Language model. An LLM is a conditional probability distribution $\pi_{\mathtt{base}}$ over sequences of text $\mathcal{V}$. Given a user query $\mathbf{x} \in \mathcal{V}$, the model generates a response $\mathbf{y}$ according to said distribution, $\mathbf{y} \sim \pi_{\mathtt{base}}(\cdot \mid \mathbf{x})$. In practice, $\pi_{\mathtt{base}}$ is obtained through pretraining and supervised fine-tuning to follow user instructions [41, 49]. This training procedure yields a highly capable model often referred to as a *base model*, but its responses may not align with human preferences, motivating the need for an alignment step [7, 40, 10, 6].

Reward models. Several alignment procedures rely on a *reward model* $r : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ that assigns a scalar score $r(\mathbf{x}, \mathbf{y})$ measuring the quality of alignment of response $\mathbf{y}$ to query $\mathbf{x}$. Ideally, the reward reflects user preferences: if a user prefers $\mathbf{y}^w$ over $\mathbf{y}^l$ in response to $\mathbf{x}$, then the associated reward should also be larger. If a reward model perfectly reflects user preferences it is termed *gold reward* and we denote it by r^*. Reward models are typically learned from preference data [38, 6, 42, 7]. However, due to annotation noise and the stochastic nature of training a reward model, the resulting model is an imperfect proxy for the true user preferences, often called a *proxy reward* [21], which we denote by $\tilde{r}$. We analyze the difference in models aligned with a gold and a proxy reward in Sec. 4.

KL-regularized alignment. At the heart of several alignment techniques [38, 42, 36] lies the KL-regularized RL problem (1). Specifically, to nudge the base model $\pi_{\mathtt{base}}$ toward user preferences, these approaches seek an aligned policy over the vocabulary $\mathcal{V}$ that maximizes the expected reward while penalizing deviation from the base model $\pi_{\mathtt{base}}$. Formally, this amounts to finding the aligned model $\pi_{r,\lambda}$ that optimizes the following objective:

$$\pi_{r,\lambda} \triangleq \arg\max_{\pi} \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathbb{E}_{\mathbf{y} \sim \pi(\cdot|\mathbf{x})} \left[r(\mathbf{x}, \mathbf{y}) \right] - \lambda \mathtt{KL}(\pi || \pi_{\mathtt{base}}). \tag{1}$$

The KL penalty term in Equation 1 prevents the model from drifting too far from $\pi_{\mathtt{base}}$, with $\lambda > 0$ directly controlling the strength of this regularization. In fact, we show in Prop. 3 that smaller KL penalties are related to more reward hacking. A fundamental property of Equation (1) is that it admits a closed-form solution given by the following tilted distribution [56, 43]:

$$\pi_{r,\lambda}(\mathbf{y} \mid \mathbf{x}) = \frac{\pi_{\mathtt{base}}(\mathbf{y} \mid \mathbf{x})\, e^{r(\mathbf{x},\mathbf{y})/\lambda}}{Z_{r,\lambda}(\mathbf{x})}, \qquad Z_{r,\lambda}(\mathbf{x}) = \mathbb{E}_{\mathbf{y} \sim \pi_{\mathtt{base}}(\cdot|\mathbf{x})} \left[e^{r(\mathbf{x},\mathbf{y})/\lambda} \right], \tag{2}$$

where $Z_{r,\lambda}$ is the *partition function* and normalizes $\pi_{\mathtt{base}}(\mathbf{y} \mid \mathbf{x})\, e^{r(\mathbf{x},\mathbf{y})/\lambda}$ into a probability measure. We use the result in Equation (2) to derive exact characterizations of the reward gain achievable under Equation (1), and to study how proxy rewards and reward ensembling affect the said gain.

The alignment problem (1) can be expressed in a constrained form, that is

$$\max_{\pi} \mathbb{E}_{\pi}[r] \quad \text{subject to} \quad \mathtt{KL}(\pi \| \pi_{\mathtt{base}}) \leq \varepsilon. \tag{3}$$

These two forms are equivalent: by treating the KL constraint via Lagrange, we can obtain (1) with λ playing the role of the multiplier. Since the constrained problem is convex and strictly feasible (at $\pi = \pi_{\mathtt{base}}$), the equivalence follows from strong duality [12]. This implies that by sweeping λ, the closed form in (2) can be used to recover the maximum reward achievable at any KL budget, which is the Pareto frontier we set out to estimate in Fig. 1. Strong duality establishes that the *fundamental limit applies to any alignment algorithm*, including ones that do not directly optimize (1) like BoN.

Limits of Alignment & The Optimality of Best-of-N. Best-of-N (BoN) has been shown to be a strong baseline for LM alignment [42, 36]. A separate line of work has attempted to explain BoN's performance by demonstrating its near-optimality for the win-rate–KL Pareto frontier [9, 23]. However, while win-rate is an interesting relative signal, it can be insufficient for measuring true alignment. For instance, in a pairwise comparison, a response that prepends a superficial refusal (e.g., "I cannot help with that...") before ultimately providing malicious instructions might strictly win against a direct malicious response. Even though it wins the comparison, its absolute safety magnitude remains unacceptably low. Optimizing for absolute expected reward circumvents this flaw and is particularly relevant for modern alignment methodologies utilizing rubric-based reward models, where magnitude serves as an interpretable signal of alignment performance [5, 34].

Despite the importance of expected reward, theoretical investigations into BoN's optimality on this metric have faced limitations. Verdun et al. [53] established that BoN can not achieve the reward–KL optimal frontier at finite N. While Yang et al. [56] showed that this optimality gap vanishes when $N \to \infty$, their proof required strict assumptions, namely that the reward model is linear and the base model is memoryless. To provide a definitive benchmark for absolute alignment, we provide the exact, assumption-free analytical solution for the optimal expected reward–KL Pareto frontier. This exact formulation allows us to directly verify the optimality gap of BoN in practice. In our experiments, we find that despite the theoretical separation described by Verdun et al. [53], BoN closely tracks the exact theoretical optimum across the entire optimal frontier.

Reward Ensembling. Reward ensembling has emerged as a practical strategy for mitigating reward hacking. Coste et al. [17] show that ensembles of reward models reduce reward hacking. Eisenstein et al. [18] compared different ensembling strategies showing that some are more effective than others on reducing hacking. In contrast, Zhang et al. [58] and Ahmed et al. [1] focused on improving the efficiency of ensembling by building light-weight reward models for the ensemble. While these works show empirically that reward ensembling mitigates reward hacking in practice, they do not provide a theoretical explanation for the observation, which we establish in Thm. 3.

3 Fundamental Limits of LLM Alignment

In this section, we compute the limit of reward gain for a KL penalty and provide an estimator to compute the limit with convergence guarantees.

3.1 Information-Theoretic Limits of Alignment

Previous work has devoted interest to understanding the dependencies of reward improvement and the distance between the base and aligned models (measured as KL). Specifically, it has been empirically observed that $\mathbb{E}_{\pi_{r,\lambda}}[r] - \mathbb{E}_{\pi_{\mathtt{base}}}[r] \approx \sqrt{\sigma \mathtt{KL}(\pi_{r,\lambda} || \pi_{\mathtt{base}})}$ for a constant σ [21]. Recently, Mroueh and Nitsure [35] showed that, for a sub-gaussian reward, this empirically observed approximation was actually an upper bound to the reward gain after alignment, i.e.,

$$\mathbb{E}_{\pi_{r,\lambda}}[r] - \mathbb{E}_{\pi_{\mathtt{base}}}[r] \leq \sqrt{\sigma \mathtt{KL}(\pi_{r,\lambda} || \pi_{\mathtt{base}})}. \tag{4}$$

The result from [35] in Equation (4) has three main limitations: (i) it does not provide a closed-form expression for the limits of reward gain, only an upper-bound; (ii) it is not directly computable (we do not have access to $\pi_{r,\lambda}$); (iii) it relies on the assumption that the reward distribution is sub-Gaussian.

We move away from upper bounds and show a closed-form solution for the improvement in reward after alignment. Specifically, we prove that the optimal reward improvement is completely characterized by the Jeffreys divergence, i.e., the sum of the forward and reverse KL.

Theorem 1 (Information-Theoretic Limits of Alignment)**.** *If $\pi_{r,\lambda}$ is the solution of Equation (1) that aligns π_{base} using r, then for every fixed query $\mathbf{x}$ we have*

$$\mathbb{E}_{\mathbf{y}\sim\pi_{r,\lambda}(\cdot|\mathbf{x})}[r(\mathbf{y},\mathbf{x})] - \mathbb{E}_{\mathbf{y}\sim\pi_{base}(\cdot|\mathbf{x})}[r(\mathbf{y},\mathbf{x})] = \lambda JD(\pi_{base}||\pi_{r,\lambda}). \tag{5}$$

Theorem 1 shows that the reward improvement after alignment is characterized by the Jeffreys divergence, the sum of the reverse and forward KL, and not only the forward KL as previously believed [21, 35]. Our identity was inspired by the work of Fuentes and Gonçalves [20], who explored the characterization of Rényi entropy in the context of statistical mechanics. Notably, our result does not impose restrictions on the distribution of the reward or LM.

Remark 1 ($\sqrt{\text{KL}}$ scale as an equality when rewards are Gaussian)**.** *Mroueh and Nitsure [35] proved that the $\sqrt{KL}$ inequality was tight when the reward model was (i) one-to-one and (ii) Gaussian with equal variance (σ) under the base and the optimally aligned models. They also note that usual rewards are not Gaussian. Theorem 1 recovers this case. When the reward is Gaussian, the forward and reverse KLs coincide, hence*

$$\mathbb{E}_{\pi_{r,\lambda}}[r] - \mathbb{E}_{\pi_{base}}[r] = \lambda \times (KL(\pi_{base}||\pi_{r,\lambda}) + KL(\pi_{r,\lambda}||\pi_{base})) = \lambda \times 2\frac{\sigma^2}{2\lambda^2} = \sqrt{2\sigma^2 KL(\pi_{r,\lambda}||\pi_{base})}.$$

A caveat of Thm. 1 is that, for fixed λ, it is not straightforward to compute the gain in reward. This is the case because the computation of the Jeffreys divergence requires access to samples from the optimally aligned model $\pi_{r,\lambda}$, which *we do not have access to*. Next, we solve this problem.

For generality, we measure performance after alignment in our theoretical analysis using a reward r' that *may* be different from the reward we are aligning to r. This generality is useful to analyze reward gain when we have access to only proxy rewards in Sec. 4. To this end, we denote the reward improvement measured with r' after aligning the base model using a reward r as

$$\Delta(r, r') \triangleq \mathbb{E}_{\pi_{r,\lambda}(\cdot|\mathbf{x})}[r'(\mathbf{y},\mathbf{x})] - \mathbb{E}_{\pi_{\text{base}}(\cdot|\mathbf{x})}[r'(\mathbf{y},\mathbf{x})]. \tag{6}$$

Next, we provide a characterization of reward gain after alignment that requires access to samples from the base policy π_{base} and scores from the reward model.

Theorem 2 (Computable Limit of Alignment)**.** *Let r' be the reward model with respect to which we are measuring performance after alignment using Equation (1)—which may be equal or different from the reward model we align the model to (r)—then for every fixed query $\mathbf{x}$, we have*

$$\Delta(r, r') = \mathbb{E}_{\pi_{r,\lambda}}[r'(\mathbf{y},\mathbf{x})] - \mathbb{E}_{\pi_{base}}[r'(\mathbf{y},\mathbf{x})] = Cov_{\pi_{base}(\cdot|\mathbf{x})}\left(r', \frac{e^{r(\mathbf{x},\mathbf{y})/\lambda}}{\mathbb{E}_{\pi_{base}(\cdot|\mathbf{x})}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\right]}\right). \tag{7}$$

Theorem 2 shows that the expected gain in reward after alignment can be written as a function of π_{base} and the reward model, not requiring samples from the optimally aligned model $\pi_{r,\lambda}$. However, the covariance computation requires access to all possible samples $\mathbf{y} \sim \pi_{\text{base}}$, which is infeasible due to the number of responses $\mathbf{y}$ being too large. We next address this.

3.2 Estimating the Limits of Alignment from Data

Having established that the optimal expected reward gain can be exactly expressed as a covariance under the base model in Thm. 2, we now operationalize this result. The critical advantage of the covariance formulation is that it allows us to estimate information-theoretic optimal post-alignment performance without any training, using only generations from the unaligned base model. We formalize the estimator for the optimal reward gain $\Delta(r, r')$ Equation (6) next.

Definition 1 (Monte Carlo Estimator for $\Delta(r, r')$)**.** *Let $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}$ be n i.i.d. samples from π_{base} for the query $\mathbf{x}$, r a gold reward model, r' a proxy reward model, and $\lambda > 0$. The Monte Carlo estimator for the expected gain in reward is:*

$$\widehat{\Delta}_n(r, r') = \frac{\widehat{Cov}}{\widehat{Z}_r} = \frac{\widehat{C} - \hat{\mu}_{r'}\widehat{Z}_r}{\widehat{Z}_r}, \tag{8}$$

where the empirical components over the n samples are the reward mean $\hat{\mu}_{r'} = \frac{1}{n}\sum_{i=1}^{n} r'(\mathbf{y}^{(i)})$ (and similarly $\hat{\mu}_r$ for r), the partition function $\widehat{Z}_r = \frac{1}{n}\sum_{i=1}^{n} \exp(r(\mathbf{y}^{(i)})/\lambda)$, and the joint term $\widehat{C} = \frac{1}{n}\sum_{i=1}^{n}\left(r(\mathbf{y}^{(i)}) \cdot \exp(r'(\mathbf{y}^{(i)})/\lambda)\right)$.

While the estimator in Def. 1 provides a computationally tractable way to estimate the reward gain, its practical utility fundamentally depends on its convergence. Next, we formally ensure that as the number of base model samples n grows, our estimate faithfully converges to the theoretical limit.

Proposition 1 (Convergence of Estimator)**.** *If the expectations $\mathbb{E}[|r'|]$, Z_r, and $\mathbb{E}[|r' \cdot \exp(r/\lambda)|]$ are finite, then $\widehat{\Delta}_n(r, r')$ is a consistent estimator for $\mathbb{E}_{\pi_{r,\lambda}(\cdot|\mathbf{x})}[r'(\mathbf{y}, \mathbf{x})] - \mathbb{E}_{\pi_{\texttt{base}}(\cdot|\mathbf{x})}[r'(\mathbf{y}, \mathbf{x})]$. That is, as $n \to \infty$:*

$$\widehat{\Delta}_n(r, r') \xrightarrow{p} \Delta(r, r') = \mathbb{E}_{\pi_{r,\lambda}(\cdot|\mathbf{x})}[r'(\mathbf{y}, \mathbf{x})] - \mathbb{E}_{\pi_{\texttt{base}}(\cdot|\mathbf{x})}[r'(\mathbf{y}, \mathbf{x})],$$

where $\xrightarrow{p}$ denotes convergence in probability.

The theoretical Pareto frontier is a trade-off curve, mapping expected reward against base model deviation (KL). Predicting this frontier requires estimating not only the reward gain, but also its corresponding KL for any given penalty λ. Fortunately, the exact same empirical components required to estimate the expected reward are sufficient to construct an estimator for the KL between the optimally aligned model $\pi_{r,\lambda}$ and the base $\pi_{\texttt{base}}$.

Definition 2 (Monte Carlo Estimator for KL Divergences)**.** *Under the notation as Def. 1, we define the KL estimator to be*

$$\widehat{KL}(\pi_{r,\lambda} \| \pi_{\texttt{base}}) = \frac{\hat{\mu}_r + \widehat{\Delta}_n(r, r)}{\lambda} - \log \widehat{Z}_r. \tag{9}$$

Just as with the reward estimator, we must formally prove that the KL estimator is convergent. By mirroring the asymptotic arguments applied to the reward gain, we can similarly guarantee the convergence of the KL estimator.

Proposition 2 (Convergence of KL Estimators)**.** *If the averages Z_r and $\mathbb{E}[|r \cdot \exp(r/\lambda)|]$ are finite, then $\widehat{KL}(\pi_{r,\lambda} \| \pi_{\texttt{base}}) \xrightarrow{p} KL(\pi_{r,\lambda} \| \pi_{\texttt{base}})$ as $n \to \infty$.*

Beyond consistency, both estimators (Def. 1 and 2) converge at rate $n^{-1/2}$: this allows us to compute variances which can be used to build confidence intervals for the frontier (see Prop. 5 in App. B.1). The information-theoretic limits derived in this section, alongside the proposed estimators, provide a complete framework to compute the theoretical KL and reward gain post-alignment for any KL penalty value. *By evaluating these estimators across a range of λ, we can estimate the optimal trade-off curve and plot the entire Pareto front using only samples from the base model.*

Our results allow practitioners to directly compare the empirical performance of various alignment algorithms against our derived theoretical Pareto frontier. By establishing this exact theoretically achievable limit, we can now quantify how close current methods are to information-theoretic optimality. Ultimately, this framework provides a definitive benchmark to determine whether substantial headroom for algorithmic improvement remains, or if further refinements to existing alignment methods will inevitably lead to diminishing returns.

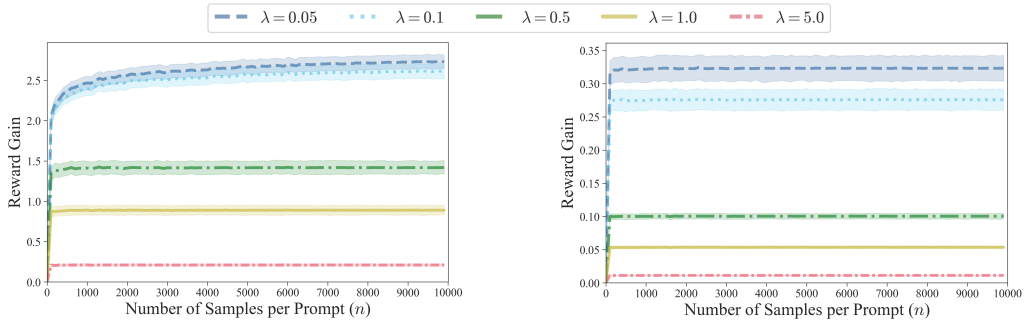
The limits established in this section apply to the setting where the reward optimized during alignment is the same reward used for evaluation or we have access to the gold reward that will be used at the time of evaluation. In practice, however, we lack direct access to a "gold" reward model r^*. Instead, alignment relies on proxy reward models $\tilde{r}$ trained from finite and noisy human preference data. Next, we connect our results to proxy rewards used in practice.

4 Proxy Rewards and a Justification for Reward Ensembling

In this section, we leverage our covariance characterization (Thm. 2) to (i) quantify how aligning to proxy rewards degrades alignment performance on the gold reward (reward hacking) [3, 21] and to (ii) theoretically explain why reward ensembling works as a mechanism to mitigate reward hacking.

4.1 The Impact of Proxy Rewards in Reward Hacking

To analyze the impact of the proxy reward on alignment, we first formalize the discrepancy between the ideal evaluation metric (gold reward) and the imperfect training signal (proxy reward).



(a) Zephyr 7B with BeaverTails

(b) Pythia 1B with TL;DR

Figure 2: **Convergence of the reward-gain estimator.** Estimated reward gain ($\widehat{\Delta}_n(r, r)$ — Def. 1) in the y-axis as a function of the number of base model samples used to estimate it (n) in the x-axis — $\lambda \in \{0.05, 0.1, 0.5, 1.0, 5.0\}$. Shaded bands show 95% confidence intervals using bootstrap [55].

Definition 3 (Preference Residual)**.** *A reward model r^* is a* gold reward *if it perfectly quantifies the preferences of a user. A proxy reward $\tilde{r}$ is an approximation for r^* that we usually have access to (e.g., model trained from pairwise preference). We denote the* ***preference residual*** *between the gold and proxy rewards by $\mathcal{N}(\mathbf{x}, \mathbf{y}) \triangleq \tilde{r}(\mathbf{x}, \mathbf{y}) - r^*(\mathbf{x}, \mathbf{y})$.*

The penalty incurred by aligning to an imperfect proxy is quantified by the performance gap between a model aligned to the gold reward and one aligned to the proxy. Proposition 3 computes the difference in expected true reward between these two policies.

Proposition 3 (The Impact of Proxy Rewards)**.** *Let r^* be the gold reward, $\tilde{r}$ the proxy reward, and $\mathcal{N}(\mathbf{x}, \mathbf{y})$ the preference residual. Denote by $\pi_{r^*,\lambda}$ the aligned model using the gold reward and, $\pi_{\tilde{r},\lambda}$ the aligned model using the proxy reward — both optimally aligned as per Thm. 2. The difference in reward is given by:*

$$\mathbb{E}_{\pi_{r^*,\lambda}(\cdot|\mathbf{x})}\left[r^*(\mathbf{x}, \mathbf{y})\right] - \mathbb{E}_{\pi_{\tilde{r},\lambda}(\cdot|\mathbf{x})}\left[r^*(\mathbf{x}, \mathbf{y})\right] = \texttt{Cov}\left(r^*(\mathbf{y}), \frac{e^{r^*(\mathbf{y})/\lambda}}{\mathbb{E}\left[e^{r^*(\mathbf{y})/\lambda}\right]}\left(1 - \frac{e^{\mathcal{N}(\mathbf{x},\mathbf{y})/\lambda}}{\mathbb{E}_{\pi^*}\left[e^{\mathcal{N}(\mathbf{x},\mathbf{y})/\lambda}\right]}\right)\right) \tag{10}$$

Proposition 3 characterizes the sensitivity that drives reward hacking. The reward degradation is governed by the exponential term $e^{\mathcal{N}(\mathbf{x},\mathbf{y})/\lambda}$, i.e., it is governed by the exponential tilt of the preference residual. Moreover, as the penalty λ decreases (allowing the model to drift further from the base policy), the impact of the proxy reward increases exponentially.

Explaining Hacking. Recent theoretical advances have formalized the manifestation of reward hacking. Khalaf et al. [32] prove that the true reward inevitably declines when aggressively optimizing a proxy reward. Aminian et al. [2] establish finite-sample regret bounds to limit the worst-case impact of alignment using a proxy reward. Huang et al. [24] characterize how such overoptimization scales with inference compute in Best-of-N. While these works establish the existence, scaling dynamics, and regret bounds of reward deterioration, our formulation complements them by providing the exact closed-form equation that drives reward degradation—the exponential tilt of the preference residual. Putting together the inevitability of reward decline established by Khalaf et al. [32] with our quantification provides a complete theoretical understanding of both the occurrence and the mechanics of reward hacking. Consequently, they yield a mathematically grounded explanation for the reward hacking curve empirically observed by Gao et al. [21, Figure 1].

4.2 Why Reward Ensembling Mitigates Reward Hacking

Given the reward degradation caused by preference residuals, practitioners need strategies to counter reward hacking. A common heuristic consists of using an ensemble of rewards [17, 18].

While ensembling has been shown to effectively mitigate hacking in practice, its effectiveness lacks formal guarantees. To solve this problem, Prop. 4 connects the difference in expected reward between two aligned models directly to the distance between their underlying rewards under the base model.

Proposition 4 (Deterministic Lipschitz Bound). *Let r^* be a bounded gold reward, and let r and s be two bounded reward models (proxy or not). Let $\pi(r, \lambda)$ and $\pi(s, \lambda)$ denote the optimally aligned models using r and s, respectively. There exists a constant $\mathrm{C}_\lambda > 0$ such that*

$$\left|\mathbb{E}_{\pi(r,\lambda)}[r^*] - \mathbb{E}_{\pi(s,\lambda)}[r^*]\right| \leq \mathrm{C}_\lambda \times \mathtt{Var}(r^*)^{1/2} \|r - s\|_{L^2(\pi_{base})}. \tag{11}$$

In the worst case, the constant C_λ is at most $\exp\left(2\max(\|r\|_\infty, \|s\|_\infty)/\lambda\right)/\lambda$.

When comparing a proxy ($r = \tilde{r}$) to the gold reward ($s = r^*$), the term $r - s$ is the preference residual $\mathcal{N} \triangleq \tilde{r} - r^*$. Hence, Prop. 4 guarantees that if a proxy reward has a small preference residual, *the post-alignment performance of the proxy-aligned model will generalize to the gold reward.*

We can also apply the bound in Equation (11) to an ensemble to derive its exact rate of convergence, demonstrating *why* reward ensembling may mitigate hacking when we can build "good" ensembles, complementing previous work that have empirically observed this result [17, 1, 18, 58].

Theorem 3 (Rate of Convergence for Reward Ensemble). *Let $r_1, ..., r_n$ be proxy rewards sampled i.i.d. from a distribution P (e.g., different reward training runs). Define R_n as the mean of the proxy rewards, i.e., $R_n(\mathbf{x}, \mathbf{y}) \triangleq \frac{1}{n}\sum_{i=1}^n r_i(\mathbf{x}, \mathbf{y})$, and let $\bar{r}(\mathbf{x}, \mathbf{y}) \triangleq \mathbb{E}_{r_i \sim P}[r_i(\mathbf{x}, \mathbf{y})]$ denote the expected (potentially biased) mean reward. If r_i and r^* are bounded for each $i \in [n]$, then*

$$\mathbb{E}_{\pi(R_n,\lambda)}\left[r^*\right] \to \mathbb{E}_{\pi(\bar{r},\lambda)}\left[r^*\right], \text{ therefore if } \bar{r} = r^* \Rightarrow \mathbb{E}_{\pi(R_n,\lambda)}\left[r^*\right] \to \mathbb{E}_{\pi_{r^*,\lambda}}\left[r^*\right], \tag{12}$$

at a rate of $O\left(\frac{1}{\sqrt{n}}\right)$ in expectation over the randomness of P.

Mitigating hacking with "good" ensembles. Theorem 3 shows that a "good" reward ensemble consists of proxy rewards that are unbiased estimators of the gold reward, i.e., $\bar{r}(\mathbf{x}, \mathbf{y}) \triangleq \mathbb{E}_{r_i \sim P}[r_i(\mathbf{x}, \mathbf{y})] = r^*$. In this setting, as the ensemble size n grows, the expected gold reward of the model aligned using the ensemble converges to the gold reward of the optimally aligned model $\left(\mathbb{E}_{\pi(r^*,\lambda)}[r^*]\right)$ at a rate of $O(1/\sqrt{n})$. This demonstrates that when we can build "good" ensembles using unbiased proxy rewards, ensembling mitigates reward hacking with theoretical guarantees.

Herding not helping. In practice, however, building unbiased proxy rewards may be challenging. Models in an ensemble are often fine-tuned from the same base model [1, 18, 58] and therefore may develop correlated errors [22]. When the ensemble shares a systematic bias ($\bar{r} \neq r^*$), Theorem 3 establishes that the ensemble still converges, but it converges to the policy aligned to this biased mean reward. This provides a justification for the "herding" phenomenon empirically observed by Eisenstein et al. [18], where ensembling successfully eliminates independent errors but cannot resolve shared mistakes. Therefore, proxy models herd toward the same flawed preferences, limiting the maximum gold reward the ensemble can recover and failing to eliminate reward hacking.

5 Experiments

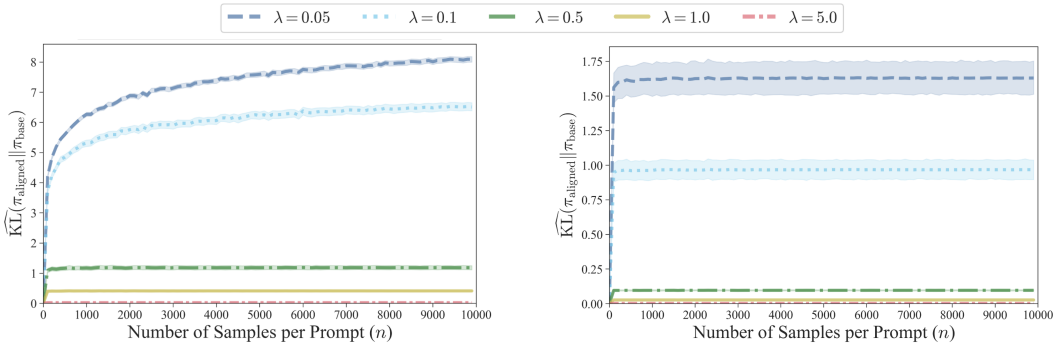
In this section, we present experiments demonstrating that (i) estimators proposed in Prop. 1 and Prop. 2 quickly converge, (ii) best-of-N is close to the theoretical limits, and (iii) GRPO, PPO, and DPO are suboptimal in terms of the KL vs. Reward gain. We omit evaluations of reward hacking and ensembling, as these are well-established phenomena [21, 17, 18].

Dataset & Tasks. We evaluate our theoretical limits on two alignment tasks: summarization and safety. For summarization, we use the `Reddit TL;DR` dataset [46], which provides Reddit posts paired with two candidate summaries and human preference labels; the task is to generate post summaries (TL;DRs) aligned with annotator preferences — during RL we use the preprocessed variant of the dataset [51]. For safety, we use `BeaverTails` [27], which consists of prompts designed to elicit harmful completions; the task is to align the model to produce safe responses.

Base & Reward Models. For summarization, we use a `Pythia-1B` [11] base model and reward model, both fine-tuned on `Reddit TL;DR` [19, 15], reproducing the setup of Huang et al. [25]. For safety on `BeaverTails`, we use `Zephyr-7B` [52] as the base model and train a corresponding reward model on pairwise preferences from `PKU-SafeRLHF` [29] using the TRL library [54].

Alignment Algorithms. We compare the theoretical limit against GRPO [45], PPO [44], and DPO [42], which are KL-regularized RL algorithms, and best-of-N (BoN), an inference-time method that has proven to be a strong baseline [36, 42, 56, 23]. Since GRPO, PPO, and DPO[3] are sensitive to

[3] DPO does not use a reward model, instead, it directly uses the preference datasets used to train the reward models.



(a) Zephyr 7b with BeaverTails

(b) Pythia 1B with TL;DR

Figure 3: **Convergence of the KL Estimator.** Estimated KL between the optimally aligned and the base model ($\widehat{\mathrm{KL}}(\pi_{r,\lambda} \| \pi_{\texttt{base}})$ — Def. 2) in the y-axis as a function of the number of samples used to estimate it in the x-axis. The KL penalty varies across $\lambda \in \{0.05, 0.1, 0.5, 1.0, 5.0\}$. Shaded bands show 95% confidence intervals using bootstrap [55].

hyperparameter choices, we performed an extensive sweep (App. E) and report the best run, measured by the KL-reward trade-off, in the main paper. For best-of-N, we vary N between 1 and 10^4 ($10k$) and compute its KL using the estimator proposed by Beirami et al. [9].

5.1 Experimental Results

Estimator convergence. Before benchmarking algorithms against the fundamental limit, we analyze how quickly our estimators converge to their asymptotic values. Fig. 2 reports the reward-gain estimator $\widehat{\Delta}_n(r, r)$ (Def. 1) and Fig. 3 reports the KL estimator $\widehat{\mathrm{KL}}(\pi_{r,\lambda} | \pi_{\texttt{base}})$ (Def. 2), as functions of the number of generations per query and across varying penalties ($\lambda \in \{0.1, 0.5, 1, 5\}$).

In Fig. 2, the reward-gain estimator converges as predicted by Prop. 1, typically requiring fewer than one thousand samples per input query. Fig. 3 shows analogous convergence behavior for the KL estimator, as predicted by Prop. 2. These results confirm that reward gain and KL are computable from base model samples alone, without access to the aligned policy. In both estimations, convergence is faster for larger λ and slower for small λ, where the exponential tilt $e^{r/\lambda}$ amplifies the variance of high-reward samples. Together, these results validate the estimators used for both Fig. 1 and Fig. 3, and show that the full theoretical Pareto frontier can be reliably estimated from a few thousand base model samples, before any alignment training.

KL–reward Pareto frontier. Fig. 1 compares the theoretical KL-reward frontier computed using $\widehat{\Delta}_n$ and $\widehat{\mathrm{KL}}(\pi_{r,\lambda} | \pi_{\texttt{base}})$ against the empirical KL-reward gain achieved by BoN, GRPO, PPO, and DPO. The theoretical frontier is computed using $10k$ samples per query from base-model and $\lambda \in [0.01, 10]$— Fig. 2 and Fig. 3 show that $1k$ samples suffice to estimate the frontier—see App. D. The black curve represents the information-theoretic achievable limit on the reward gain at each KL budget (Thm. 1), while the gray region marks *unachievable* rewards. The vertical gap between an algorithm's empirical point and the frontier quantifies its suboptimality.

best-of-N is almost Pareto optimal. Fig. 1 shows that BoN's KL vs. reward gain points lie on or near the Pareto curve at *every* KL value, equivalently, every number of samples N. Prior work established that BoN exhibits near-optimality in terms of win-rates [9, 23] but not in terms of expected reward. In fact, [53] established that BoN remains separated from the exact reward–KL optimal at finite N while [56] proved that BoN's reward converges to the optimal when $N \to \infty$. We complement these works by showing that despite theoretical separation, BoN is near-optimal across the *entire trade-off curve*. Therefore, BoN is not merely a strong baseline that approaches optimality when $N \to \infty$, it is close to the best any alignment algorithm can achieve at every KL budget.

GRPO, PPO, and DPO are suboptimal. Fig. 1 shows that GRPO, PPO, and DPO are substantially below the theoretical frontier at every KL budget tested. This points to a structural limitation of these RL methods, which operate far from the KL-optimal frontier. Recent work supports this finding, demonstrating that the RL–BoN performance gap can be closed with explicit reward calibration [8].

6 Final Discussion

Conclusion. We characterized the information-theoretic limits of LM alignment in terms of KL vs. reward gain. First, we derived the optimal reward improvement after alignment showing that the gain in expected reward is governed by the Jeffreys divergence $\lambda\,\mathtt{JD}(\pi_{r,\lambda}\|\pi_{\mathtt{base}})$, improving upon the previously conjectured $\sqrt{\mathtt{KL}(\pi_{r,\lambda}\|\pi_{\mathtt{base}})}$ scaling which was empirically observed [6, 21] and proven to be an upper bound for optimal alignment by Mroueh and Nitsure [35].

Second, we rewrite the Jeffreys divergence term as a covariance under the base model, yielding a characterization of the optimal reward gain that does not require access to the aligned model. Building on this, we proposed estimators that recover the full KL–reward Pareto frontier from base model samples alone. This enables practitioners to predict reward gains and benchmark alignment algorithms against an *achievable limit*. Empirically, we found that BoN tracks the alignment frontier, while GRPO, PPO, and DPO are suboptimal, pointing to significant room for algorithmic improvements.

Third, we leveraged our covariance characterization to analyze the discrepancy between optimally aligned LMs using the gold and proxy rewards. We prove that this discrepancy grows with the preference residual and the inverse KL penalty λ^{-1}. By isolating these exact dependencies, we complement recent theoretical advances on reward deterioration [2, 24, 32]. Together with these results, we explain why aggressive alignment with proxy rewards leads to reward hacking. Building on the same characterization, we show that ensembling provably mitigates reward hacking, providing a theoretical guarantee for this widely used practice.

Implications and Future Work. Practically, our estimators for the KL vs. reward gain Pareto frontier allow principled hyperparameter selection and algorithm evaluation: practitioners can use the Pareto frontier to identify the KL budget required for a target reward level, and to compare the absolute performance of alignment algorithms. Estimating the frontier is also more cost effective, requiring only a one-time, parallelizable inference cost. In contrast, training-based methods require model updates and hyper-parameter sweeps and fail to reach the theoretical limit. This could be due to reasons such as a smaller number of on-policy samples per query compared to BoN or reward miscalibration [8]—we leave a deeper understanding of the pitfalls to future work.

We find that BoN closely tracks the theoretical limit, but its inference cost scales linearly with N. In our experiments, approaching the theoretical limit at the high-KL end required an N of almost $10{,}000$, which needs N generations per query and N reward model calls, making this approach impractical for deployment. Developing methods that (i) reduce the inference cost of near-optimal inference-time methods [8, 53, 36] and (ii) achieve optimal reward–KL trade-off without high inference overhead [23] therefore remains an important open direction for future work.

On the modeling side, our analysis on the use of proxy rewards exposes a structural limitation of the dominant reward-modeling paradigm. Currently, rewards are trained under the Bradley-Terry (BT) model. This approach captures only the *relative preference* of responses, but not *how much* one is preferred over another. As a consequence, the preference residual $\mathcal{N}$ between learned proxy and gold reward can grow unboundedly, even with a perfectly minimized BT loss — see App. C for a more detailed justification. However, Prop. 3 shows that the preference residual is precisely the culprit behind the degradation of alignment quality. These findings suggest that a *new reward training paradigm*, capable of directly constraining the preference residual (for instance, by incorporating preference *magnitudes*) could mitigate reward hacking at its source, rather than relying on post-hoc corrections such as ensembling. We believe that designing such feedback protocols, and understanding how they interact with the fundamental limits derived here, are important directions for future work.

Zero-Training Reward Model Selection. Our covariance formulation in Prop. 3 also provides an efficient method for evaluating proxy reward models. Rather than incurring the prohibitive cost of aligning a model to test candidate proxy rewards ($\tilde{r}$), practitioners can evaluate $\tilde{r}$ alongside a gold signal (r^*) directly on base model generations. By computing the expected gold reward gain via our covariance identity (or BoN), practitioners can select the best reward models without any training.

Limitations. Our work has limitations we hope future work will address. First, our results characterize the optimal solution to the KL-regularized objective in Equation (1); extending the analysis to alternative objectives (e.g., f-divergence regularization, win-rate maximization, per-token KL) is an interesting direction. Second, our proposed estimators converge slowly for smaller λ (large $\mathtt{KL}$); future work could focus on designing better estimators for this regime.

Acknowledgments and Disclosure of Funding

Acknowledgment. We thank Parikshit Gopalan and Andrew Silva for their valuable feedback on earlier versions of the paper. We also thank Ahmad Beirami and Flavio Calmon for pointing us to relevant literature on the optimality of best-of-N and reward hacking. Finally, we thank the NeurIPS 2026 reviewers for their valuable feedback.

Funding Disclosure. This work was supported entirely by Apple, the primary affiliation of all authors. No third-party funding, hardware donations, or external support was received. The authors declare no competing financial interests or additional revenues outside of their primary employment during the 36 months prior to this submission.

References

[1] Ahmed M. Ahmed, Rafael Rafailov, Stepan Sharkov, Xuechen Li, and Sanmi Koyejo. Scalable ensembling for mitigating reward overoptimisation, 2024. URL https://arxiv.org/abs/2406.01013.

[2] Gholamali Aminian, Idan Shenfeld, Amir R. Asadi, Ahmad Beirami, and Youssef Mroueh. Best-of-n through the smoothing lens: KL divergence and regret analysis. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=tCv1D3M7Lb.

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016. URL https://arxiv.org/abs/1606.06565.

[4] Anthropic. Claude opus 4.6 system card. Technical report, Anthropic, February 2026. URL https://www-cdn.anthropic.com/14e4fb01875d2a69f646fa5e574dea2b1c0ff7b5.pdf.

[5] Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. Healthbench: Evaluating large language models towards improved human health, 2025. URL https://arxiv.org/abs/2505.08775.

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

[7] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. URL https://arxiv.org/abs/2212.08073.

[8] Ananth Balashankar, Ziteng Sun, Jonathan Berant, Jacob Eisenstein, Michael Collins, Adrian Hutter, Jong Lee, Chirag Nagpal, Flavien Prost, Aradhana Sinha, Ananda Theertha Suresh, and Ahmad Beirami. Infalign: Inference-aware language model alignment. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=hInfvt7c4p.

[9] Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander Nicholas D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=u3U8qzFV7w.

[10] Yoshua Bengio, Geoffrey Hinton, Andrew Yao, Dawn Song, Pieter Abbeel, Trevor Darrell, Yuval Noah Harari, Ya-Qin Zhang, Lan Xue, Shai Shalev-Shwartz, et al. Managing extreme ai risks amid rapid progress. *Science*, 384(6698):842–845, 2024.

[11] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023. URL https://arxiv.org/abs/2304.01373.

[12] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[13] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

[14] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4302–4310, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[15] CleanRL. Eleutherai_pythia-1b-deduped__reward__tldr (reward model), 2023. URL https://huggingface.co/cleanrl/EleutherAI_pythia-1b-deduped__reward__tldr.

[16] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

[18] Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alexander Nicholas D'Amour, Krishnamurthy Dj Dvijotham, Adam Fisch, Katherine A Heller, Stephen Robert Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=5u1GpUkKtG.

[19] TRL (Hugging Face). pythia-1b-deduped-tldr-sft, 2024. URL https://huggingface.co/trl-lib/pythia-1b-deduped-tldr-sft.

[20] Jesús Fuentes and Jorge M. Gonçalves. Rényi entropy in statistical mechanics. *Entropy*, 24, 2022. URL https://api.semanticscholar.org/CorpusID:251454034.

[21] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[22] Juan Felipe Gomez, Caio Machado, Lucas Monteiro Paes, and Flavio Calmon. Algorithmic arbitrariness in content moderation. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, page 2234–2253, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704505. doi: 10.1145/3630106.3659036. URL https://doi.org/10.1145/3630106.3659036.

[23] Lin Gui, Cristina Garbacea, and Victor Veitch. BoNBon alignment for large language models and the sweetness of best-of-n sampling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=haSKMlrbX5.

[24] Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J Foster. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=QnjfkhrbYK.

[25] 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, 2024. URL https://arxiv.org/abs/2403.17031.

[26] Harold Jeffreys. *The theory of probability*. OuP Oxford, 1998.

[27] Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Chi Zhang, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset, 2023. URL https://arxiv.org/abs/2307.04657.

[28] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*, 2023.

[29] Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Juntao Dai, Boren Zheng, Tianyi Qiu, Jiayi Zhou, Kaile Wang, Boxuan Li, Sirui Han, Yike Guo, and Yaodong Yang. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference, 2025. URL https://arxiv.org/abs/2406.15513.

[30] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[31] Yilun Jin, Zheng Li, Chenwei Zhang, Tianyu Cao, Yifan Gao, Pratik Sridatt Jayarao, Mao Li, Xin Liu, Ritesh Sarkhel, Xianfeng Tang, Haodong Wang, Zhengyang Wang, Wenju Xu, Jingfeng Yang, Qingyu Yin, Xian Li, Priyanka Nigam, Yi Xu, Kai Chen, Qiang Yang, Meng Jiang, and Bing Yin. Shopping MMLU: A massive multi-task online shopping benchmark for large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=D3jyWDBZTk.

[32] Hadi Khalaf, Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio Calmon. Inference-time reward hacking in large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=hSX7Dd8dxy.

[33] Deepak Kumar, Yousef Anees AbuHashem, and Zakir Durumeric. Watch your language: Investigating content moderation with large language models. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 18, pages 865–878, 2024.

[34] Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. Openrubrics: Towards scalable synthetic rubric generation for reward modeling and llm alignment, 2026. URL https://arxiv.org/abs/2510.07743.

[35] Youssef Mroueh and Apoorva Nitsure. Information theoretic guarantees for policy alignment in large language models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Uz9J77Riul.

[36] 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*, ICML'24. JMLR.org, 2024.

[37] OpenAI. Using gpt-4 for content moderation, 2023. URL https://openai.com/index/using-gpt-4-for-content-moderation. Accessed: 2024-05-01.

[38] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[39] Lucas Monteiro Paes, Nivedha Sivakumar, Yinong Oliver Wang, Masha Fedzechkina, Barry-John Theobald, Luca Zappella, and Nicholas Apostoloff. Dso: Direct steering optimization for bias mitigation, 2026. URL https://arxiv.org/abs/2512.15926.

[40] Peter S Park, Simon Goldstein, Aidan O'Gara, Michael Chen, and Dan Hendrycks. Ai deception: A survey of examples, risks, and potential solutions. *Patterns*, 5(5), 2024.

[41] XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, September 2020. ISSN 1869-1900. doi: 10.1007/s11431-020-1647-3. URL http://dx.doi.org/10.1007/s11431-020-1647-3.

[42] 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 *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[43] Salman Salamatian, Litian Liu, Ahmad Beirami, and Muriel Médard. Mismatched guesswork, 2019. URL https://arxiv.org/abs/1907.00531.

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

[45] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[46] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[47] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[48] Terence Tao. Machine-assisted proof. *Notices of the American Mathematical Society*, 72(1): 6–13, 2025.

[49] Guiyao Tie, Zeli Zhao, Dingjie Song, Fuyang Wei, Rong Zhou, Yurou Dai, Wen Yin, Zhejian Yang, Jiangyue Yan, Yao Su, Zhenhan Dai, Yifeng Xie, Yihan Cao, Lichao Sun, Pan Zhou, Lifang He, Hechang Chen, Yu Zhang, Qingsong Wen, Tianming Liu, Neil Zhenqiang Gong, Jiliang Tang, Caiming Xiong, Heng Ji, Philip S. Yu, and Jianfeng Gao. A survey on post-training of large language models, 2025. URL https://arxiv.org/abs/2503.06072.

[50] 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, 2023. URL https://arxiv.org/abs/2307.09288.

[51] TRL Team. Tl;dr dataset for trl. https://huggingface.co/datasets/trl-lib/tldr, 2025.

[52] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023. URL https://arxiv.org/abs/2310.16944.

[53] Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio P. Calmon. Soft best-of-n sampling for model alignment, 2025. URL https://arxiv.org/abs/2505.03156.

[54] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL https://github.com/huggingface/trl.

[55] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

[56] Joy Qiping Yang, Salman Salamatian, Ziteng Sun, Ananda Theertha Suresh, and Ahmad Beirami. Asymptotics of language model alignment. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 2027–2032, 2024. doi: 10.1109/ISIT57864.2024.10619456.

[57] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: towards scalable real-world web interaction with grounded language agents. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

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

[59] Yao Zhao, Mikhail Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. Calibrating sequence likelihood improves conditional language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0qSOodKmJaN.

Supplementary Material: Theoretical Limits of Language Model Alignment

A Preliminaries and Notation

Preliminaries. In the supplementary material we provide the following information:

- App. B provides the proofs for all theoretical results presented in the paper.
- App. C discusses the limitations of the Bradley-Terry (BT) reward-modeling paradigm and provides a path to improve reward hacking with a new ideation for reward modeling that accounts for preference residuals.
- App. D presents additional empirical results that validate the convergence of the estimated Reward-KL Pareto frontier.
- App. E provides the full details for the extensive hyper-parameter sweeps performed for the GRPO and PPO baseline experiments.
- App. G details the licenses for all public datasets, model checkpoints, and software libraries utilized in our evaluation.

Notation Table. We show the notation used through the paper in Tab. 1.

Symbol	Meaning
$\mathcal{V}$	vocabulary of the language model
$\mathbf{x}$	user prompt
$\mathbf{y}$	model response
$\pi_{\texttt{base}}$	base language model (before alignment)
λ	KL penalty parameter in Equation (1)
$\pi_{r,\lambda}$	optimally aligned language model policy for reward r and penalty λ
r	general reward model function
r^*	ideal (gold) reward model
$\tilde{r}$	proxy reward model
$\texttt{KL}(P\|\|Q)$	Kullback-Leibler divergence between distributions P and Q
$\texttt{JD}(P\|\|Q)$	Jeffreys divergence, i.e., $\texttt{JD}(P\|\|Q) \triangleq \texttt{KL}(P\|\|Q) + \texttt{KL}(Q\|\|P)$
$\Delta(r, r')$	expected reward gain measured with r' after aligning with r
$\widehat{\Delta}_n$	Monte Carlo estimator for the expected reward gain
$\mathcal{N}(\mathbf{x}, \mathbf{y})$	preference residual between gold and proxy rewards, i.e., $\mathcal{N}(\mathbf{x}, \mathbf{y}) \triangleq \tilde{r} - r^*$
$Z_{r,\lambda}(\mathbf{x})$	partition function for the tilted distribution
N	number of samples per query generated for best-of-N
n	number of base model samples per query used for Monte Carlo estimation

Table 1: Notation used throughout the main paper and appendix.

B Theoretical Results and Proofs

B.1 Proof of Theoretical results in Section 3

Theorem 1 (Information-Theoretic Limits of Alignment)**.** *If $\pi_{r,\lambda}$ is the solution of Equation (1) that aligns π_{base} using r, then for every fixed query $\mathbf{x}$ we have*

$$\mathbb{E}_{\mathbf{y}\sim\pi_{r,\lambda}(\cdot|\mathbf{x})}[r(\mathbf{y},\mathbf{x})] - \mathbb{E}_{\mathbf{y}\sim\pi_{base}(\cdot|\mathbf{x})}[r(\mathbf{y},\mathbf{x})] = \lambda JD(\pi_{base}||\pi_{r,\lambda}). \tag{5}$$

Proof. As per (2), the following relationship holds between aligned and base models:

$$\frac{\pi(r,\lambda)(\mathbf{y}\mid\mathbf{x})}{\pi_{\texttt{base}}(\mathbf{y}\mid\mathbf{x})} = \frac{e^{r(\mathbf{x},\mathbf{y})/\lambda}}{Z_{r,\lambda}(\mathbf{x})}. \tag{13}$$

Taking logarithms both sides gives

$$\begin{aligned} &\log\pi(r,\lambda)(\mathbf{y}\mid\mathbf{x}) - \log\pi_{\texttt{base}}(\mathbf{y}\mid\mathbf{x}) = \frac{r(\mathbf{x},\mathbf{y})}{\lambda} - \log Z_{r,\lambda}(\mathbf{x}) \\ \implies\quad & r(\mathbf{x},\mathbf{y}) = \lambda\left(\log\frac{\pi(r,\lambda)(\mathbf{y}\mid\mathbf{x})}{\pi_{\texttt{base}}(\mathbf{y}\mid\mathbf{x})} + \log Z_{r,\lambda}(\mathbf{x})\right). \end{aligned} \tag{14}$$

Taking the expectation of (14) under $\pi(r,\lambda)$ gives

$$\mathbb{E}_{\mathbf{y}\sim\pi(r,\lambda)}[r(\mathbf{x},\mathbf{y})] = \lambda\texttt{KL}\left(\pi(r,\lambda)||\pi_{\texttt{base}}\right) + \lambda\log Z_{r,\lambda}(\mathbf{x}), \tag{15}$$

while taking the expectation of (14) under $\pi_{\texttt{base}}$ similarly yields

$$\mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}[r(\mathbf{x},\mathbf{y})] = -\lambda\texttt{KL}\big(\pi_{\texttt{base}}||\pi(r,\lambda)\big) + \lambda\log Z_{r,\lambda}(\mathbf{x}). \tag{16}$$

Finally, subtracting the two expectations we get

$$\begin{aligned} \mathbb{E}_{\mathbf{y}\sim\pi(r,\lambda)}[r(\mathbf{x},\mathbf{y})] - \mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}[r(\mathbf{x},\mathbf{y})] &= \lambda\left(\texttt{KL}(\pi(r,\lambda)||\pi_{\texttt{base}}) + \texttt{KL}(\pi_{\texttt{base}}||\pi(r,\lambda))\right) \\ &= \lambda\texttt{JD}(\pi(r,\lambda)||\pi_{\texttt{base}}), \end{aligned}$$

with JD denoting Jeffreys divergence, which completes the proof. □

Theorem 2 (Computable Limit of Alignment)**.** *Let r' be the reward model with respect to which we are measuring performance after alignment using Equation (1)—which may be equal or different from the reward model we align the model to (r)—then for every fixed query $\mathbf{x}$, we have*

$$\Delta(r,r') = \mathbb{E}_{\pi_{r,\lambda}}[r'(\mathbf{y},\mathbf{x})] - \mathbb{E}_{\pi_{base}}[r'(\mathbf{y},\mathbf{x})] = \texttt{Cov}_{\pi_{base}(\cdot|\mathbf{x})}\left(r', \frac{e^{r(\mathbf{x},\mathbf{y})/\lambda}}{\mathbb{E}_{\pi_{base}(\cdot|\mathbf{x})}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\right]}\right). \tag{7}$$

Proof. First, note that using (2) we can write the expectation under the aligned model as

$$\mathbb{E}_{\mathbf{y}\sim\pi_{r,\lambda}}\left[r'(\mathbf{x},\mathbf{y})\right] = \sum_{\mathbf{y}\in\mathcal{V}}\frac{\pi_{\texttt{base}}(\mathbf{y}\mid\mathbf{x})e^{r(\mathbf{x},\mathbf{y})/\lambda}}{\mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\right]}r'(\mathbf{x},\mathbf{y}) = \frac{\mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\,r'(\mathbf{x},\mathbf{y})\right]}{\mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\right]}. \tag{17}$$

Plugging this into (7), we have

$$\begin{aligned} &\mathbb{E}_{\mathbf{y}\sim\pi_{r,\lambda}}\left[r'(\mathbf{x},\mathbf{y})\right] - \mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[r'(\mathbf{x},\mathbf{y})\right] \\ =&\frac{\mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\,r'(\mathbf{x},\mathbf{y})\right]}{\mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\right]} - \mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[r'(\mathbf{x},\mathbf{y})\right] \\ =&\frac{\mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\,r'(\mathbf{x},\mathbf{y})\right] - \mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\right]\mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[r'(\mathbf{x},\mathbf{y})\right]}{\mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\right]} \end{aligned} \tag{18}$$

Note that the numerator in (18) is in the form $\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$, which is exactly the covariance between X and Y. This allows us to rewrite (18) as

$$\begin{aligned} \mathbb{E}_{\mathbf{y}\sim\pi_{r,\lambda}}\left[r'(\mathbf{x},\mathbf{y})\right] - \mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[r'(\mathbf{x},\mathbf{y})\right] &= \frac{\texttt{Cov}_{\pi_{\texttt{base}}}\left(r'(\mathbf{x},\mathbf{y}), e^{r(\mathbf{x},\mathbf{y})/\lambda}\right)}{\mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\right]} \\ &= \texttt{Cov}_{\pi_{\texttt{base}}}\left(r'(\mathbf{x},\mathbf{y}), \frac{e^{r(\mathbf{x},\mathbf{y})/\lambda}}{\mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{r(\mathbf{x},\mathbf{y})/\lambda}\right]}\right), \end{aligned} \tag{19}$$

thus concluding the proof. □

Proposition 1 (Convergence of Estimator)**.** *If the expectations* $\mathbb{E}[|r'|]$*,* Z_r*, and* $\mathbb{E}[|r' \cdot \exp(r/\lambda)|]$ *are finite, then* $\widehat{\Delta}_n(r, r')$ *is a consistent estimator for* $\mathbb{E}_{\pi_{r,\lambda}(\cdot|\mathbf{x})}[r'(\mathbf{y}, \mathbf{x})] - \mathbb{E}_{\pi_{base}(\cdot|\mathbf{x})}[r'(\mathbf{y}, \mathbf{x})]$*. That is, as* $n \to \infty$*:*

$$\widehat{\Delta}_n(r, r') \xrightarrow{p} \Delta(r, r') = \mathbb{E}_{\pi_{r,\lambda}(\cdot|\mathbf{x})}[r'(\mathbf{y}, \mathbf{x})] - \mathbb{E}_{\pi_{base}(\cdot|\mathbf{x})}[r'(\mathbf{y}, \mathbf{x})],$$

where $\xrightarrow{p}$ *denotes convergence in probability.*

Proof. The proof relies on the Weak Law of Large Numbers (WLLN) and the Continuous Mapping Theorem.

1. Convergence of Component Estimators (WLLN):

The WLLN states that for a sequence of i.i.d. random variables $X_1, \ldots, X_n$ with a finite expectation $\mathbb{E}[X]$, the sample mean $\bar{X}_n = \frac{1}{n}\sum X_i$ converges in probability to the expectation: $\bar{X}_n \xrightarrow{p} \mathbb{E}[X]$.

We apply this to our three primary estimators (from Definition 1), which are all sample means of i.i.d. variables (since each $y^{(i)}$ is drawn i.i.d. from π_{base}):

- $\hat{\mu}_{r'} \xrightarrow{p} \mathbb{E}_{\pi_{\text{base}}}[r'(y)] = \mu_{r'}$
- $\hat{Z}_r \xrightarrow{p} \mathbb{E}_{\pi_{\text{base}}}[\exp(r(y)/\lambda)] = Z_r$
- $\hat{C} \xrightarrow{p} \mathbb{E}_{\pi_{\text{base}}}[r'(y) \cdot \exp(r(y)/\lambda)] = C$

This holds because we assumed these expectations are finite.

2. Convergence of Combined Estimators (Continuous Mapping):

The Continuous Mapping Theorem states that for a sequence of random variables $X_n \xrightarrow{p} c$, and a function g that is continuous at c, we have $g(X_n) \xrightarrow{p} g(c)$.

The covariance estimator $\widehat{\text{Cov}}$ is a continuous function g_1 of our three sample means:

$$\widehat{\text{Cov}} = g_1(\hat{C}, \hat{\mu}_{r'}, \hat{Z}_r) = \hat{C} - (\hat{\mu}_{r'} \cdot \hat{Z}_r)$$

Since multiplication and subtraction are continuous functions, and all arguments converge in probability to their respective limits, the estimator $\widehat{\text{Cov}}$ converges in probability to the true covariance:

$$\widehat{\text{Cov}} \xrightarrow{p} C - \mu_{r'} Z_r = \text{Cov}_{\pi_{\text{base}}}(r', \exp(r/\lambda))$$

Finally, the gain estimator $\widehat{\Delta}_n(r, r')$ is a continuous function g_2 of $\widehat{\text{Cov}}$ and $\hat{Z}_r$:

$$\widehat{\Delta}_n(r, r') = g_2(\widehat{\text{Cov}}, \hat{Z}_r) = \frac{\widehat{\text{Cov}}}{\hat{Z}_r}$$

Since division is a continuous function (given our assumption that the denominator $Z_r \neq 0$), $\widehat{\Delta}_n(r, r')$ converges in probability to the true gain $\Delta(r, r')$:

$$\widehat{\Delta}_n(r, r') \xrightarrow{p} \frac{\text{Cov}_{\pi_{\text{base}}}(r', \exp(r/\lambda))}{Z_r} = \Delta(r, r')$$

Showing that $\widehat{\Delta}_n(r, r')$ converges to $\Delta(r, r')$. □

Proposition 2 (Convergence of KL Estimators)**.** *If the averages* Z_r *and* $\mathbb{E}[|r \cdot \exp(r/\lambda)|]$ *are finite, then* $\widehat{KL}(\pi_{r,\lambda} \| \pi_{base}) \xrightarrow{p} KL(\pi_{r,\lambda} \| \pi_{base})$ *as* $n \to \infty$*.*

Proof. Taking logarithms in (2) gives $\log \frac{\pi_{r,\lambda}(\mathbf{y}|\mathbf{x})}{\pi_{\text{base}}(\mathbf{y}|\mathbf{x})} = r(\mathbf{x}, \mathbf{y})/\lambda - \log Z(\mathbf{x})$, so taking expectation under $\pi_{r,\lambda}$ yields the population quantity

$$\text{KL}(\pi_{r,\lambda} \| \pi_{\text{base}}) = \frac{\mathbb{E}_{\pi_{r,\lambda}}[r]}{\lambda} - \log Z = \frac{\mu_r + \Delta(r, r)}{\lambda} - \log Z, \tag{20}$$

where the second equality uses $\mathbb{E}_{\pi_{r,\lambda}}[r] = \mathbb{E}_{\pi_{\text{base}}}[r] + \Delta(r, r)$ from Thm. 2. The estimator $\widehat{\text{KL}}(\pi_{r,\lambda} \| \pi_{\text{base}})$ replaces each population quantity with its sample counterpart:

1. $\hat{\mu}_r \xrightarrow{p} \mu_r$ by the WLLN.
2. $\widehat{\Delta}_n(r,r) \xrightarrow{p} \Delta(r,r)$ by Prop. 1, finiteness conditions are satisfied by hypothesis.
3. $\widehat{Z}_r \xrightarrow{p} Z_r$ by the WLLN, since $\mathbb{E}[\exp(r/\lambda)] = Z_r < \infty$ by hypothesis; then $\log \widehat{Z}_r \xrightarrow{p} \log Z_r$ by the Continuous Mapping Theorem, since $Z_r > 0$.

The estimator is a continuous function of these three convergent quantities, so by a further application of the Continuous Mapping Theorem, $\widehat{\mathtt{KL}}(\pi_{r,\lambda}\|\pi_{\mathtt{base}}) \xrightarrow{p} \mathtt{KL}(\pi_{r,\lambda}\|\pi_{\mathtt{base}})$. □

Proposition 5 (Variance of Estimators)**.** *Under the notation of Def. 1, let* $C_r \triangleq \mathbb{E}_{\pi_{base}}[re^{r/\lambda}]$ *and* $C \triangleq \mathbb{E}_{\pi_{base}}[r'e^{r/\lambda}]$*, and assume that*

$$\mathbb{E}_{\pi_{base}}[r'^2] < \infty, \qquad \mathbb{E}_{\pi_{base}}[(1 + r^2 + r'^2)e^{2r/\lambda}] < \infty. \tag{21}$$

Then, as $n \to \infty$*, we have*

$$\begin{aligned} \sqrt{n}\left(\widehat{KL}(\pi_{r,\lambda}\|\pi_{base}) - KL(\pi_{r,\lambda}\|\pi_{base})\right) &\xrightarrow{d} \mathcal{N}(0, \sigma^2_{KL}), \\ \textit{and} \qquad \sqrt{n}\left(\widehat{\Delta}_n(r,r') - \Delta(r,r')\right) &\xrightarrow{d} \mathcal{N}(0, \sigma^2_{\Delta}), \end{aligned} \tag{22}$$

where $\xrightarrow{d}$ *denotes convergence in distribution, with*

$$\sigma^2_{KL} = \mathit{Var}_{\pi_{base}}\left[\frac{e^{r/\lambda}}{Z_r}\left(\frac{r}{\lambda} - \frac{C_r}{\lambda Z_r} - 1\right)\right], \qquad \sigma^2_{\Delta} = \mathit{Var}_{\pi_{base}}\left[\frac{e^{r/\lambda}}{Z_r}\left(r' - \frac{C}{Z_r}\right) - r'\right]. \tag{23}$$

Proof. We start by proving the result for $\widehat{\mathtt{KL}}$, with the argument for $\widehat{\Delta}_n$ following a similar procedure. By substituting Def. 1 into Def. 2, the sample mean cancels out:

$$\hat{\mu}_r + \widehat{\Delta}_n(r,r) = \frac{\widehat{C}_r}{\widehat{Z}_r}, \quad \text{with} \quad \widehat{C}_r \triangleq \frac{1}{n}\sum_{i=1}^{n} r(\mathbf{y}^{(i)})e^{r(\mathbf{y}^{(i)})/\lambda}. \tag{24}$$

It follows that

$$\widehat{\mathtt{KL}}(\pi_{r,\lambda}\|\pi_{\mathtt{base}}) = g(\widehat{Z}_r, \widehat{C}_r) = g(Z_r, C_r), \qquad \text{where} \qquad g(z,c) \triangleq \frac{c}{\lambda z} - \log z. \tag{25}$$

The quantity $(\widehat{Z}_r, \widehat{C}_r)$ represents the sample mean of the i.i.d. vectors $V_i \triangleq \left(e^{r(\mathbf{y}^{(i)})/\lambda}, r(\mathbf{y}^{(i)})e^{r(\mathbf{y}^{(i)})/\lambda}\right)$, whose covariance matrix Σ is finite by assumption. By the (multivariate) Central Limit Theorem, we have

$$\sqrt{n}\left((\widehat{Z}_r, \widehat{C}_r) - (Z_r, C_r)\right) \xrightarrow{d} \mathcal{N}(0, \Sigma). \tag{26}$$

We also have that g is differentiable at (Z_r, C_r) (since $Z_r > 0$), with gradient

$$\nabla g(Z_r, C_r) = \left(-\frac{C_r}{\lambda Z_r^2} - \frac{1}{Z_r}, \frac{1}{\lambda Z_r}\right). \tag{27}$$

Applying the delta method allows us to recover

$$\sqrt{n}\left(\widehat{\mathtt{KL}}(\pi_{r,\lambda}\|\pi_{\mathtt{base}}) - \mathtt{KL}(\pi_{r,\lambda}\|\pi_{\mathtt{base}})\right) \xrightarrow{d} \mathcal{N}\left(0, \nabla g^\top \Sigma \nabla g\right). \tag{28}$$

The first of the sought variances in (23) is then given by

$$\sigma^2_{\mathtt{KL}} = \nabla g^\top \Sigma \nabla g = \mathtt{Var}_{\pi_{\mathtt{base}}}[\nabla g^\top V_i] = \mathtt{Var}_{\pi_{\mathtt{base}}}\left[\frac{e^{r/\lambda}}{Z_r}\left(\frac{r}{\lambda} - \frac{C_r}{\lambda Z_r} - 1\right)\right]. \tag{29}$$

The other variance is recovered similarly. We have:

$$\widehat{\Delta}_n(r,r') = h(\hat{\mu}_{r'}, \widehat{Z}_r, \widehat{C}), \qquad \text{with} \qquad h(\mu, z, c) \triangleq \frac{c}{z} - \mu. \tag{30}$$

Following the same steps with $W_i = \left(r'(\mathbf{y}^{(i)}),\ e^{r(\mathbf{y}^{(i)})/\lambda}, r'(\mathbf{y}^{(i)})e^{r(\mathbf{y}^{(i)})/\lambda}\right)$ and $\nabla h(\mu_{r'}, Z_r, C) = \left(-1, -\frac{C}{Z_r^2}, \frac{1}{Z_r}\right)$ gives

$$\sigma^2_{\Delta} = \mathtt{Var}_{\pi_{\mathtt{base}}}[\nabla h^\top W_i] = \mathtt{Var}_{\pi_{\mathtt{base}}}\left[\frac{e^{r/\lambda}}{Z_r}\left(r' - \frac{C}{Z_r}\right) - r'\right], \tag{31}$$

terminating the proof. □

B.2 Proof of Theoretical results in Section 4

Proposition 3 (The Impact of Proxy Rewards). *Let r^* be the gold reward, $\tilde{r}$ the proxy reward, and $\mathcal{N}(\mathbf{x}, \mathbf{y})$ the preference residual. Denote by $\pi_{r^*,\lambda}$ the aligned model using the gold reward and, $\pi_{\tilde{r},\lambda}$ the aligned model using the proxy reward — both optimally aligned as per Thm. 2. The difference in reward is given by:*

$$\mathbb{E}_{\pi_{r^*,\lambda}(\cdot|\mathbf{x})}\left[r^*(\mathbf{x},\mathbf{y})\right] - \mathbb{E}_{\pi_{\tilde{r},\lambda}(\cdot|\mathbf{x})}\left[r^*(\mathbf{x},\mathbf{y})\right] = \texttt{Cov}\left(r^*(\mathbf{y}), \frac{e^{r^*(\mathbf{y})/\lambda}}{\mathbb{E}\left[e^{r^*(\mathbf{y})/\lambda}\right]}\left(1 - \frac{e^{\mathcal{N}(\mathbf{x},\mathbf{y})/\lambda}}{\mathbb{E}_{\pi^*}\left[e^{\mathcal{N}(\mathbf{x},\mathbf{y})/\lambda}\right]}\right)\right) \tag{10}$$

Proof. Fix a query $\mathbf{x}$ and work conditionally on $\mathbf{x}$. Define the two partition functions

$$Z_* \triangleq \mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{r^*(\mathbf{y})/\lambda}\right], \qquad Z_\sim \triangleq \mathbb{E}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left[e^{\tilde{r}(\mathbf{y})/\lambda}\right], \tag{32}$$

and recall that $\mathcal{N}(\mathbf{y}) = \tilde{r}(\mathbf{y}) - r^*(\mathbf{y})$ is the preference residual. Applying Thm. 2 with alignment reward $r = r^*$ and evaluation reward $r' = r^*$, and again with $r = \tilde{r}$ and $r' = r^*$, gives

$$\mathbb{E}_{\pi_{r^*,\lambda}}[r^*] - \mathbb{E}_{\pi_{\texttt{base}}}[r^*] = \texttt{Cov}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left(r^*(\mathbf{y}), \tfrac{e^{r^*(\mathbf{y})/\lambda}}{Z_*}\right), \tag{33}$$

$$\mathbb{E}_{\pi_{\tilde{r},\lambda}}[r^*] - \mathbb{E}_{\pi_{\texttt{base}}}[r^*] = \texttt{Cov}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left(r^*(\mathbf{y}), \tfrac{e^{\tilde{r}(\mathbf{y})/\lambda}}{Z_\sim}\right). \tag{34}$$

Subtracting the second from the first, the $\mathbb{E}_{\pi_{\texttt{base}}}[r^*]$ terms cancel, and linearity of the covariance in the second argument yields

$$\mathbb{E}_{\pi_{r^*,\lambda}}[r^*] - \mathbb{E}_{\pi_{\tilde{r},\lambda}}[r^*] = \texttt{Cov}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left(r^*(\mathbf{y}), \frac{e^{r^*(\mathbf{y})/\lambda}}{Z_*} - \frac{e^{\tilde{r}(\mathbf{y})/\lambda}}{Z_\sim}\right). \tag{35}$$

Algebraic factorization of the second argument and the substitution $\mathcal{N} = \tilde{r} - r^*$ give

$$\frac{e^{r^*/\lambda}}{Z_*} - \frac{e^{\tilde{r}/\lambda}}{Z_\sim} = \frac{e^{r^*/\lambda}}{Z_*}\left(1 - \frac{Z_*}{Z_\sim}e^{(\tilde{r}-r^*)/\lambda}\right) = \frac{e^{r^*/\lambda}}{Z_*}\left(1 - \frac{Z_*}{Z_\sim}e^{\mathcal{N}(\mathbf{y})/\lambda}\right). \tag{36}$$

Substituting back into (35) yields

$$\mathbb{E}_{\pi_{r^*,\lambda}}[r^*] - \mathbb{E}_{\pi_{\tilde{r},\lambda}}[r^*] = \texttt{Cov}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left(r^*(\mathbf{y}), \frac{e^{r^*(\mathbf{y})/\lambda}}{Z_*}\left(1 - \frac{Z_*}{Z_\sim}e^{\mathcal{N}(\mathbf{y})/\lambda}\right)\right). \tag{37}$$

Next, we express $Z_\sim/Z_*$ under $\pi_{r^*,\lambda}$:

$$\frac{Z_\sim}{Z_*} = \frac{\mathbb{E}_{\pi_{\texttt{base}}}\left[e^{(r^*+\mathcal{N})/\lambda}\right]}{\mathbb{E}_{\pi_{\texttt{base}}}\left[e^{r^*/\lambda}\right]} \tag{38}$$

$$= \frac{1}{Z_*}\sum_{\mathbf{y}\in\mathcal{V}} \pi_{\texttt{base}}(\mathbf{y})\, e^{r^*(\mathbf{y})/\lambda}\, e^{\mathcal{N}(\mathbf{y})/\lambda} \tag{39}$$

$$= \mathbb{E}_{\mathbf{y}\sim\pi_{r^*,\lambda}}\left[e^{\mathcal{N}(\mathbf{y})/\lambda}\right]. \tag{40}$$

Hence

$$\frac{Z_*}{Z_\sim} = \frac{1}{\mathbb{E}_{\mathbf{y}\sim\pi_{r^*,\lambda}}\left[e^{\mathcal{N}(\mathbf{y})/\lambda}\right]}. \tag{41}$$

Substituting (41) into (37),

$$\mathbb{E}_{\pi_{r^*,\lambda}}[r^*] - \mathbb{E}_{\pi_{\tilde{r},\lambda}}[r^*] = \texttt{Cov}_{\mathbf{y}\sim\pi_{\texttt{base}}}\left(r^*(\mathbf{y}), \frac{e^{r^*(\mathbf{y})/\lambda}}{Z_*}\left(1 - \frac{e^{\mathcal{N}(\mathbf{y})/\lambda}}{\mathbb{E}_{\mathbf{y}\sim\pi_{r^*,\lambda}}\left[e^{\mathcal{N}(\mathbf{y})/\lambda}\right]}\right)\right), \tag{42}$$

which completes our proof. □

Proposition 4 (Deterministic Lipschitz Bound). *Let r^* be a bounded gold reward, and let r and s be two bounded reward models (proxy or not). Let $\pi(r,\lambda)$ and $\pi(s,\lambda)$ denote the optimally aligned models using r and s, respectively. There exists a constant $\mathrm{C}_\lambda > 0$ such that*

$$\left|\mathbb{E}_{\pi(r,\lambda)}[r^*] - \mathbb{E}_{\pi(s,\lambda)}[r^*]\right| \leq \mathrm{C}_\lambda \times \texttt{Var}(r^*)^{1/2}\|r - s\|_{L^2(\pi_{base})}. \tag{11}$$

In the worst case, the constant C_λ is at most $\exp\left(2\max(\|r\|_\infty, \|s\|_\infty)/\lambda\right)/\lambda$.

Proof. Let's fix a query $\mathbf{x}$ and work conditionally on $\mathbf{x}$. By the tilted-form characterization of the aligned model,

$$\pi(r,\lambda)(\mathbf{y} \mid \mathbf{x}) = \frac{\pi_{\texttt{base}}(\mathbf{y} \mid \mathbf{x}) e^{r(\mathbf{x},\mathbf{y})/\lambda}}{\mathbb{E}_{\pi_{\texttt{base}}}[e^{r(\mathbf{x},\mathbf{y})/\lambda}]}.$$

Define the partition function and the normalized density weights as

$$Z_r \triangleq \mathbb{E}_{\pi_{\texttt{base}}}[e^{r/\lambda}], \qquad w_r(\mathbf{x},\mathbf{y}) \triangleq \frac{e^{r(\mathbf{x},\mathbf{y})/\lambda}}{Z_r},$$

and define Z_s and w_s similarly for s (dependence on $\mathbf{x}$ is implicit).

The fundamental limit covariance identity applied with alignment reward r and evaluation reward r^* gives

$$\mathbb{E}_{\pi(r,\lambda)}[r^*] - \mathbb{E}_{\pi_{\texttt{base}}}[r^*] = \texttt{Cov}_{\pi_{\texttt{base}}}(r^*, w_r).$$

Similarly, for alignment reward s:

$$\mathbb{E}_{\pi(s,\lambda)}[r^*] - \mathbb{E}_{\pi_{\texttt{base}}}[r^*] = \texttt{Cov}_{\pi_{\texttt{base}}}(r^*, w_s).$$

Subtracting the second equation from the first, the $\mathbb{E}_{\pi_{\texttt{base}}}[r^*]$ terms cancel, and the linearity of covariance in the second argument yields

$$\mathbb{E}_{\pi(r,\lambda)}[r^*] - \mathbb{E}_{\pi(s,\lambda)}[r^*] = \texttt{Cov}_{\pi_{\texttt{base}}}(r^*, w_r - w_s).$$

By Cauchy–Schwarz, we can bound this absolute difference:

$$\begin{aligned}
\left|\mathbb{E}_{\pi(r,\lambda)}[r^*] - \mathbb{E}_{\pi(s,\lambda)}[r^*]\right| &= \left|\texttt{Cov}_{\pi_{\texttt{base}}}(r^*, w_r - w_s)\right| \\
&\le \texttt{Var}(r^*)^{1/2} \texttt{Var}(w_r - w_s)^{1/2} \\
&\le \texttt{Var}(r^*)^{1/2} \mathbb{E}_{\pi_{\texttt{base}}}[(w_r - w_s)^2]^{1/2} \\
&= \texttt{Var}(r^*)^{1/2} \|w_r - w_s\|_{L^2(\pi_{\texttt{base}})}. \qquad (43)
\end{aligned}$$

We now bound $\|w_r - w_s\|_{L^2(\pi_{\texttt{base}})}$ in terms of $\|r - s\|_{L^2(\pi_{\texttt{base}})}$. Define an interpolation w_t for $t \in [0,1]$:

$$w_t = \frac{e^{(s+(r-s)t)/\lambda}}{Z_t}, \quad \text{where} \quad Z_t = \mathbb{E}_{\pi_{\texttt{base}}}[e^{(s+(r-s)t)/\lambda}].$$

By the fundamental theorem of calculus,

$$w_r - w_s = \int_0^1 \frac{d}{dt} w_t \, dt = \frac{1}{\lambda} \int_0^1 w_t\big((r-s) - \mathbb{E}_{\pi_{\texttt{base}}}[w_t(r-s)]\big) \, dt.$$

Substituting this into $\mathbb{E}_{\pi_{\texttt{base}}}[(w_r - w_s)^2]$ and applying Cauchy–Schwarz to swap the square and integral gives:

$$\begin{aligned}
\mathbb{E}_{\pi_{\texttt{base}}}[(w_r - w_s)^2] &= \frac{1}{\lambda^2} \mathbb{E}_{\pi_{\texttt{base}}}\left[\left(\int_0^1 w_t\big((r-s) - \mathbb{E}_{\pi_{\texttt{base}}}[w_t(r-s)]\big) \, dt\right)^2\right] \\
&\le \frac{1}{\lambda^2} \mathbb{E}_{\pi_{\texttt{base}}}\left[\int_0^1 w_t^2\big((r-s) - \mathbb{E}_{\pi_{\texttt{base}}}[w_t(r-s)]\big)^2 \, dt\right]. \qquad (44)
\end{aligned}$$

Let $W = \sup_{t\in[0,1]} \|w_t\|_\infty$. Because r and s are bounded, W is finite. Extracting one factor of W from the integral yields:

$$\begin{aligned}
\mathbb{E}_{\pi_{\texttt{base}}}[(w_r - w_s)^2] &\le \frac{W}{\lambda^2} \int_0^1 \mathbb{E}_{\pi_{\texttt{base}}}\left[w_t\big((r-s) - \mathbb{E}_{\pi_{\texttt{base}}}[w_t(r-s)]\big)^2\right] dt \\
&= \frac{W}{\lambda^2} \int_0^1 \texttt{Var}_{w_t}[r-s] \, dt. \qquad (45)
\end{aligned}$$

A property of the exponential family of distributions is that the second derivative of the log-partition function relates directly to the variance:

$$\frac{1}{\lambda^2} \texttt{Var}_{w_t}[r-s] = \frac{d}{dt} \mathbb{E}_{\pi_{\texttt{base}}}[w_t(r-s)].$$

Substituting this back and integrating:

$$\mathbb{E}_{\pi_{\texttt{base}}}[(w_r - w_s)^2] \le \frac{W}{\lambda}\int_0^1 \frac{d}{dt}\mathbb{E}_{\pi_{\texttt{base}}}[w_t(r-s)]\,dt$$
$$= \frac{W}{\lambda}\mathbb{E}_{\pi_{\texttt{base}}}[(w_r - w_s)(r-s)]. \tag{46}$$

Applying Cauchy–Schwarz once more:

$$\mathbb{E}_{\pi_{\texttt{base}}}[(w_r - w_s)^2] \le \frac{W}{\lambda}\mathbb{E}_{\pi_{\texttt{base}}}[(w_r - w_s)^2]^{1/2}\mathbb{E}_{\pi_{\texttt{base}}}[(r-s)^2]^{1/2}.$$

Dividing both sides by $\mathbb{E}_{\pi_{\texttt{base}}}[(w_r - w_s)^2]^{1/2}$ isolates the norm:

$$\mathbb{E}_{\pi_{\texttt{base}}}[(w_r - w_s)^2]^{1/2} \le \frac{W}{\lambda}\mathbb{E}_{\pi_{\texttt{base}}}[(r-s)^2]^{1/2} = \frac{W}{\lambda}\|r-s\|_{L^2(\pi_{\texttt{base}})}.$$

Combining this with Equation (43) and defining the constant $C = W/\lambda$ completes the proof:

$$\left|\mathbb{E}_{\pi(r,\lambda)}[r^*] - \mathbb{E}_{\pi(s,\lambda)}[r^*]\right| \le C \times \texttt{Var}(r^*)^{1/2}\|r-s\|_{L^2(\pi_{\texttt{base}})}. \qquad \square$$

Theorem 3 (Rate of Convergence for Reward Ensemble)**.** *Let $r_1, ..., r_n$ be proxy rewards sampled i.i.d. from a distribution P (e.g., different reward training runs). Define R_n as the mean of the proxy rewards, i.e., $R_n(\mathbf{x},\mathbf{y}) \triangleq \frac{1}{n}\sum_{i=1}^n r_i(\mathbf{x},\mathbf{y})$, and let $\bar{r}(\mathbf{x},\mathbf{y}) \triangleq \mathbb{E}_{r_i\sim P}[r_i(\mathbf{x},\mathbf{y})]$ denote the expected (potentially biased) mean reward. If r_i and r^* are bounded for each $i \in [n]$, then*

$$\mathbb{E}_{\pi(R_n,\lambda)}[r^*] \to \mathbb{E}_{\pi(\bar{r},\lambda)}[r^*], \text{ therefore if } \bar{r} = r^* \Rightarrow \mathbb{E}_{\pi(R_n,\lambda)}[r^*] \to \mathbb{E}_{\pi_{r^*,\lambda}}[r^*], \tag{12}$$

at a rate of $O\left(\frac{1}{\sqrt{n}}\right)$ in expectation over the randomness of P.

Proof. Let $\Delta(r)$ denote the post-RLHF true reward when we align with reward r, i.e.,

$$\Delta(r) \triangleq \mathbb{E}_{\pi(r,\lambda)}[r^*].$$

In particular,

$$\Delta(R_n) = \mathbb{E}_{\pi(R_n,\lambda)}[r^*], \qquad \Delta(\bar{r}) = \mathbb{E}_{\pi(\bar{r},\lambda)}[r^*].$$

By the deterministic Lipschitz bound (proved earlier), there exists a constant $C > 0$ such that

$$\left|\Delta(r) - \Delta(\bar{r})\right| \le C \times \texttt{Var}(r^*)^{1/2}\|r-\bar{r}\|_{L^2(\pi_{\texttt{base}})} \tag{47}$$

for all reward functions r, where $\|\cdot\|_{L^2(\pi_{\texttt{base}})}$ is the L^2 norm under the base model $\pi_{\texttt{base}}$.

Apply (47) with $r = R_n$:

$$\left|\Delta(R_n) - \Delta(\bar{r})\right| \le C \times \texttt{Var}(r^*)^{1/2}\|R_n-\bar{r}\|_{L^2(\pi_{\texttt{base}})}. \tag{48}$$

We now bound the right-hand side in expectation over the randomness of the ensemble $\{r_i\}_{i=1}^n$. We will work in the Hilbert space $L^2(\pi_{\texttt{base}})$ with inner product $\langle f, g\rangle = \mathbb{E}_{\pi_{\texttt{base}}}[fg]$. Define the centered random variables

$$\Delta_i \triangleq r_i - \bar{r}, \qquad R_n - \bar{r} = \frac{1}{n}\sum_{i=1}^n \Delta_i.$$

First compute the second moment of the $L^2(\pi_{\texttt{base}})$-norm:

$$\mathbb{E}\left\|R_n - \bar{r}\right\|^2_{L^2(\pi_{\texttt{base}})} = \mathbb{E}\Big\langle \frac{1}{n}\sum_{i=1}^n \Delta_i, \frac{1}{n}\sum_{j=1}^n \Delta_j \Big\rangle$$
$$= \frac{1}{n^2}\sum_{i,j=1}^n \mathbb{E}\langle \Delta_i, \Delta_j\rangle. \tag{49}$$

Since $r_1, \ldots, r_n$ are i.i.d. with mean $\bar{r}$ in $L^2(\pi_{\texttt{base}})$, we have $\mathbb{E}[\Delta_i] = 0$ and, for $i \ne j$,

$$\mathbb{E}\langle \Delta_i, \Delta_j\rangle = \langle \mathbb{E}\Delta_i, \mathbb{E}\Delta_j\rangle = 0.$$

Thus only the diagonal terms don not cancel (49):

$$\begin{aligned}\mathbb{E}\big\|R_n - \bar{r}\big\|^2_{L^2(\pi_{\text{base}})} &= \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\|\Delta_i\|^2_{L^2(\pi_{\text{base}})} \\ &= \frac{1}{n^2}\cdot n\,\mathbb{E}\|\Delta_1\|^2_{L^2(\pi_{\text{base}})} \\ &= \frac{1}{n}\,\mathbb{E}\|r_1 - \bar{r}\|^2_{L^2(\pi_{\text{base}})}. \end{aligned} \tag{50}$$

Set

$$\sigma^2 \triangleq \mathbb{E}\|r_1 - \bar{r}\|^2_{L^2(\pi_{\text{base}})} < \infty,$$

which is finite because each r_i is bounded. Then

$$\mathbb{E}\big\|R_n - \bar{r}\big\|^2_{L^2(\pi_{\text{base}})} = \frac{\sigma^2}{n}. \tag{51}$$

By Cauchy–Schwarz,

$$\mathbb{E}\big\|R_n - \bar{r}\big\|_{L^2(\pi_{\text{base}})} \le \Big(\mathbb{E}\big\|R_n - \bar{r}\big\|^2_{L^2(\pi_{\text{base}})}\Big)^{1/2} \le \frac{\sigma}{\sqrt{n}}. \tag{52}$$

Taking expectations in (48) and using (52) gives

$$\mathbb{E}\big|\Delta(R_n) - \Delta(\bar{r})\big| \le C \times \texttt{Var}(r^*)^{1/2}\mathbb{E}\big\|R_n - \bar{r}\big\|_{L^2(\pi_{\text{base}})} \le \frac{C\texttt{Var}(r^*)^{1/2}\sigma}{\sqrt{n}}. \tag{53}$$

In other words,

$$\mathbb{E}\big|\mathbb{E}_{\pi(R_n,\lambda)}[r^*] - \mathbb{E}_{\pi(\bar{r},\lambda)}[r^*]\big| = O\left(\frac{\texttt{Var}(r^*)^{1/2}}{\sqrt{n}}\right),$$

which yields the desired $O(n^{-1/2})$ rate. □

C On the need for a new paradigm for reward modeling

Proxy reward models are predominantly trained using the Bradley-Terry model, which frames human feedback strictly as binary pairwise comparisons over relative rankings, and thus fails to impose restrictions on preference residuals. Specifically, under the BT framework, the probability that a response $\mathbf{y}_w$ is preferred over a response $\mathbf{y}_l$ given a prompt $\mathbf{x}$ is modeled as $\Pr(\mathbf{y}_w \succ \mathbf{y}_l \mid \mathbf{x}) = \sigma(\tilde{r}(\mathbf{x}, \mathbf{y}_w) - \tilde{r}(\mathbf{x}, \mathbf{y}_l))$, where σ is the logistic function. However, because this objective captures only the relative ordering of preferences, it provides no structural guarantee that the learned proxy reward $\tilde{r}$ is quantitatively close to the unobserved gold reward r^* in terms of preference residual. Hence, a reward can perfectly minimize the BT loss with an arbitrarily large preference residual $\mathcal{N}(\mathbf{x}, \mathbf{y})$.

The result in Prop. 3 shows that preference residual —uncontrolled in the BT framework— causes an exponential decrease in the gold reward, leading to reward hacking. Conversely, Prop. 4 provides justification that if the preference residual is small, then reward hacking is mitigated, i.e., gains in performance are transferred to the evaluation using the gold reward.

Together, these results highlight *the necessity for a new reward modeling paradigm*: we **must** incorporate preference magnitude to directly anchor the proxy reward, allowing us to actively measure and minimize its absolute ℓ_2 distance to the gold reward. Incorporating the magnitude of a preference would anchor the scale of the proxy, providing a direct mechanism to measure and minimize the preference residual.

Example: BT-invariant but increases the proxy–gold gap. For a prompt $\mathbf{x}$ suppose that the set of candidate responses decomposes into two disjoint clusters

$$\mathcal{V} = \mathcal{V}_A(\mathbf{x}) \cup \mathcal{V}_B(\mathbf{x}), \qquad \mathcal{V}_A(\mathbf{x}) \cap \mathcal{V}_B(\mathbf{x}) = \varnothing, \tag{54}$$

where $\mathcal{V}$ is the set of all possible responses. Also, consider that BT preference dataset for $\mathbf{x}$ contains only within-cluster comparisons, i.e., every labeled pair $(\mathbf{y}^w, \mathbf{y}^l)$ satisfies either $\mathbf{y}^w, \mathbf{y}^l \in \mathcal{V}_A(\mathbf{x})$ or $\mathbf{y}^w, \mathbf{y}^l \in \mathcal{V}_B(\mathbf{x})$.

When do clustered preference clusters arise? This two-component structure naturally emerges in practice from mixture-of-sources candidate generation paired with within-source preference labeling. The following are examples of when this emerges:

1. **Helpfulness and Harmlessness** We can generate one batch of candidates using a "helpful" decoding regime (cluster $\mathcal{V}_A(\mathbf{x})$) and a second batch using a "safe/refusal" regime (cluster $\mathcal{V}_B(\mathbf{x})$), as in HH-RLHF [6], where it is useful to train a *helpful* and *harmless* reward.
2. **Subtile Safety Risks** To reduce annotation costs, alignment pipelines may use a *selective mechanism* where an AI labels simpler candidate pairs while humans evaluate only the most challenging ones. This inadvertently creates isolated data clusters.
3. **Reasoning** A model might generate one batch of candidates using Chain-of-Thought reasoning, and another batch using direct answers. If human raters compare only CoT responses to other CoT responses (to judge the logic) and direct answers to direct answers (to judge conciseness), we have two clusters.
4. **Tool Use** When training an LLM to use external tools (like web search), we may evaluate tool-use trajectories against other tool-uses to improve tool usage. These are never compared against a model simply answering from its own internal weights.

In these scenarios, the BT model learns accurate within-cluster rankings but leaves the relative reward offset between the clusters weakly identified. The shift preserves all observed BT pairwise differences while potentially inflating the absolute preference residual $\mathcal{N}$.

The theoretical Argument. For any $M > 0$, define a modified proxy reward $\tilde{r}^{(M)}(\mathbf{x}, \mathbf{y}) \triangleq r^*(\mathbf{x}, \mathbf{y}) + M\,\mathbf{1}\{\mathbf{y} \in \mathcal{V}_A(\mathbf{x})\} - M\,\mathbf{1}\{\mathbf{y} \in \mathcal{V}_B(\mathbf{x})\}$. For any observed preference pair $(\mathbf{y}^w, \mathbf{y}^l)$, the pair lies within a single cluster; hence the $\pm M$ offsets cancel. Therefore every BT likelihood term is unchanged, and consequently $\mathcal{L}_{\mathrm{BT}}(\tilde{r}^{(M)}) = \mathcal{L}_{\mathrm{BT}}(r^*)$, which is the "optimal".

However, note that the preference residual can be written as

$$\mathcal{N}(\mathbf{x}, \mathbf{y}) \triangleq \tilde{r}^{(M)}(\mathbf{x}, \mathbf{y}) - r^*(\mathbf{x}, \mathbf{y}) = M\,\mathbf{1}\{\mathbf{y} \in \mathcal{V}_A(\mathbf{x})\} - M\,\mathbf{1}\{\mathbf{y} \in \mathcal{V}_B(\mathbf{x})\}. \tag{55}$$

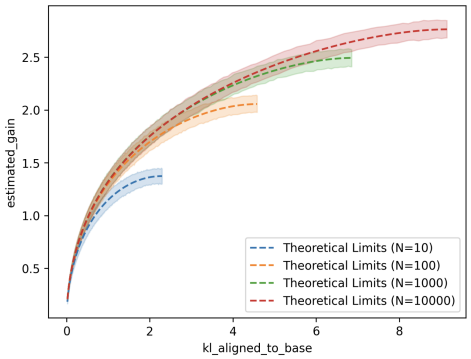
Hence $\|\mathcal{N}^{(M)}\|_{L^2(\pi_{\mathtt{base}})}$ (and $\sup_{\mathbf{y}} |\mathcal{N}^{(M)}(\mathbf{x}, \mathbf{y})|$) can be made arbitrarily large by increasing M, while the BT training objective remains identical. At the same time, Proposition 3 shows that the proxy-gold degradation is governed by the exponential tilt given as

$$\frac{e^{\mathcal{N}(\mathbf{x},\mathbf{y})/\lambda}}{\mathbb{E}_{\pi^*}\left[e^{\mathcal{N}(\mathbf{x},\mathbf{y})/\lambda}\right]} = \frac{e^{M/\lambda}\,\mathbf{1}\{\mathbf{y} \in \mathcal{V}_A(\mathbf{x})\} + e^{-M/\lambda}\,\mathbf{1}\{\mathbf{y} \in \mathcal{V}_B(\mathbf{x})\}}{\pi^*(\mathcal{V}_A(\mathbf{x}))e^{M/\lambda} + \pi^*(\mathcal{V}_B(\mathbf{x}))e^{-M/\lambda}} > 0, \tag{56}$$

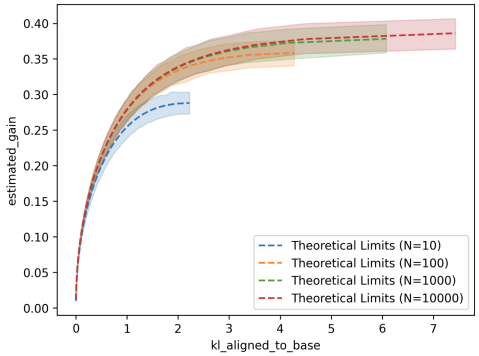
which amplifies the mismatch between the proxy-aligned model $\pi(\tilde{r}^{(M)}, \lambda)$ and the gold-aligned model $\pi_{r^*,\lambda}$, thereby increasing the gap δ in Proposition 3.

D Reward-KL Pareto Frontier

Fig. 4 validates the practicality of our proposed estimators for the Pareto frontier by showing the convergence of the entire theoretical Reward-KL Pareto frontier as a function of the number of base model samples n. Figure 4 illustrates the estimated frontiers using $n \in \{10, 100, 1000, 10000\}$ across both tasks (safety and summarization). We observe that while small sample sizes (e.g., $n = 10$ or 100) systematically underestimate the achievable reward and prematurely truncate in the high-KL regime, the frontier rapidly stabilizes as n increases. This behavior directly reflects the mechanics of the estimators in Def. 1 and Def. 2: in the high-KL regime (which corresponds to a small penalty λ), the exponential tilt $e^{r/\lambda}$ amplifies the variance of high-reward samples, thereby requiring a larger number of samples to achieve a tight estimate. Crucially, the curves for $n = 1000$ and $n = 10000$ are nearly overlapping across most of the KL spectrum. This macro view of the frontier's convergence complements the pointwise convergence guarantees of Prop. 1 and Prop. 2, as well as the fixed-λ empirical results in Fig. 2 and Fig. 3. Ultimately, Fig. 4 confirms that a few thousand samples (around one thousand) are sufficient to reliably delineate the fundamental limits of alignment from base model samples alone.



(a) Zephyr 7b with BeaverTails



(b) Pythia 1B with TL;DR

Figure 4: **Convergence of Pareto front.** Estimated Reward gain (Def. 1) vs. KL divergence (Def. 2) Pareto frontier using different number of samples per prompt. Shaded bands show 95% confidence intervals using bootstrap from Seaborn [55].

E Experimental Runs Hyper-parameter Sweeps

To achieve the experimental results presented in the paper for GRPO, PPO, and DPO, we ran an extensive hyper-parameter sweep. The reward gain per KL value for each run in the sweep is displayed in Figure 5 for training with Beavertails using GRPO, Figure 6 for Beavertails using PPO, Figure 7 for TL;DR using GRPO, Figure 8 for TL;DR using PPO, Figure 9 for Beavertails using DPO, and Figure 10 for TL;DR using DPO. The GRPO, PPO, and DPO results with the highest reward gains are displayed in the main body in Figure. 1. The parameters and maximum reward gain for the KL values across hyper-parameter sweeps can be found in Table 2 and Table 3 for tuning Zephyr with Beavertails and Pythia with TL;DR, respectively, with GRPO and PPO, and in Table 4 for DPO (Beavertails in Table 4a and TL;DR in Table 4b). The run from each fine-tuning method and model/dataset combination that are shown in the main body are in **bold**.

β	**LR**	**BS**	**Num Gens**	max $R\Delta$
2e-2	5e-7	256	16	2.101
2e-2	**3e-7**	**256**	**16**	**2.117**
1.5e-2	3e-7	256	16	2.103
1e-2	5e-7	256	64	2.083
1e-2	5e-7	256	16	2.066
1e-2	3e-7	256	16	2.088
1e-2	2e-7	256	16	2.104
5e-3	5e-7	256	16	2.034
3e-3	5e-7	256	16	2.051
1e-2	2e-7	256	16	1.909
1.5e-2	2e-7	256	16	1.886
1e-2	2e-7	256	8	1.899
2e-2	2e-7	256	16	1.889

(a) GRPO

β	**LR**	**BS**	T	**PPO Epochs**	max $R\Delta$
5e-2	3e-6	32	1.0	1	0.918
1e-3	1e-6	32	1.0	1	1.573
5e-2	3e-6	32	0.7	2	1.809
5e-2	1e-6	32	1.0	1	1.908
5e-2	1e-6	32	1.0	2	1.924
3e-2	1e-6	32	1.0	1	1.232
2e-2	**1e-6**	**32**	**0.7**	**1**	**2.084**
1e-2	1e-6	32	0.7	2	1.544
2e-2	1e-6	32	0.7	2	1.698
1e-2	3e-6	32	0.7	2	1.810
1e-2	1e-6	32	0.5	1	1.357
2e-2	3e-6	32	0.7	2	1.411
2e-2	3e-6	32	0.5	2	1.809

(b) PPO

Table 2: The hyper parameters swept for training Zephyr-7B-SFT-full on the BeaverTails dataset with **GRPO** and **PPO**. The maximum reward gain is reported within the KL range that the theoretical limits are calculated for [0, 9.14]. The hyper-parameter run that is used in Figure 1 is **bolded**. For comparison, the maximum reward gain for the theoretical limit and BoN are 2.764 and 2.722, respectively.

β	**LR**	**BS**	**Num Gens**	max $R\Delta$
1e-5	1e-6	128	16	0.175
1e-1	1e-6	128	4	0.251
1e-1	1e-6	128	4	0.253
1e-1	1e-6	128	4	0.256
1e-1	1e-6	128	4	0.249
1e-2	1e-6	128	8	0.243
1e-3	1e-6	128	8	0.220
1e-3	1e-6	128	4	0.196
1e-4	1e-6	128	8	0.225
1e-1	1e-6	128	4	0.277
5e-2	1e-6	128	8	0.276
7e-2	**1e-6**	**128**	**8**	**0.282**
5e-2	5e-7	128	8	0.272
3e-2	1e-6	128	8	0.257
5e-2	1e-6	128	16	0.273

(a) GRPO

β	**LR**	**BS**	T	**PPO Epochs**	max $R\Delta$
2e-2	3e-6	64	0.7	4	0.114
1.5e-2	3e-6	64	0.7	4	0.147
1e-2	3e-6	32	0.7	4	0.113
1e-2	**3e-6**	**64**	**0.7**	**2**	**0.187**
1e-2	3e-6	64	1.0	4	0.145
1e-2	3e-6	64	0.7	6	0.214
1.5e-2	3e-6	64	0.7	4	0.157
7e-3	3e-6	64	0.7	4	0.163
5e-3	3e-6	64	0.7	4	0.011
5e-3	5e-6	64	0.7	4	0.012
3e-3	3e-6	64	0.7	4	0.022
7e-3	3e-6	128	0.7	4	0.167
7e-3	3e-6	64	0.3	4	0.131
7e-3	5e-7	64	0.7	4	0.212
1e-2	1e-5	64	0.7	4	0.143

(b) PPO

Table 3: The hyper parameters swept for training Pythia 1B SFT on the TL;DR dataset with **GRPO** and **PPO**. The maximum reward gain is reported within the KL range that the theoretical limits are calculated for [0, 7.96]. The hyper-parameter run that is used in Figure 1 is **bolded**. For comparison, the maximum reward gain for the theoretical limit and BoN are 0.387 and 0.385, respectively. For PPO, the two highest estimated reward gain runs were not used in the main body due to how noisy they are. (See Fig. 8 for more details)

β	**LR**	**BS**	max $R\Delta$
1e-2	1e-7	128	1.690
1e-2	**1e-7**	**256**	**1.688**
1e-2	5e-7	128	1.002
1e-2	5e-7	256	0.819
1e-2	1e-6	128	0.516
1e-2	1e-6	256	0.512
5e-2	1e-7	128	1.536
5e-2	1e-7	256	1.541
5e-2	5e-7	128	1.442
5e-2	5e-7	256	1.469
5e-2	1e-6	128	1.014
5e-2	1e-6	256	1.036
1e-1	1e-7	128	1.327
1e-1	1e-7	256	1.315
1e-1	5e-7	128	1.517
1e-1	5e-7	256	1.364
1e-1	1e-6	128	1.358
1e-1	1e-6	256	1.411

(a) Zephyr 7B with Beavertails

β	**LR**	**BS**	max $R\Delta$
1e-2	1e-7	128	0.105
1e-2	1e-7	256	0.053
1e-2	5e-7	128	0.203
1e-2	5e-7	256	0.171
1e-2	1e-6	128	0.198
1e-2	1e-6	256	0.192
5e-2	1e-7	128	0.090
5e-2	1e-7	256	0.053
5e-2	5e-7	128	0.212
5e-2	5e-7	256	0.197
5e-2	1e-6	128	0.212
5e-2	1e-6	256	0.214
1e-1	1e-7	128	0.072
1e-1	1e-7	256	0.044
1e-1	5e-7	128	0.195
1e-1	5e-7	256	0.165
1e-1	1e-6	128	0.215
1e-1	**1e-6**	**256**	**0.218**

(b) Pythia 1B with TL;DR

Table 4: The hyper parameters swept for training Zephyr-7B-SFT-full on the BeaverTails dataset (a) and Pythia 1B SFT on the TL;DR dataset (b) with **DPO**. The maximum reward gain is reported within the KL range that the theoretical limits are calculated for, [0, 9.14] for BeaverTails and [0, 7.96] for TL;DR. The hyper-parameter run that is used in Figure 1 is **bolded**.

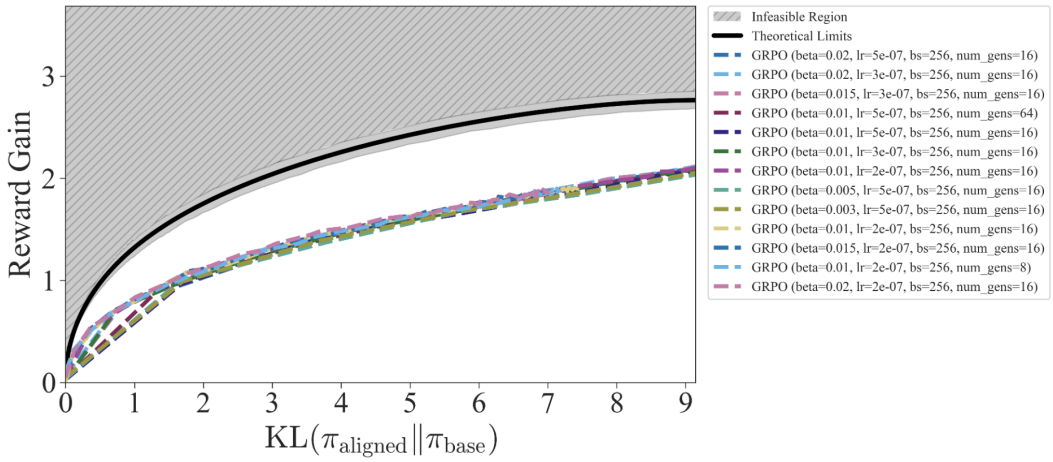


Figure 5: **Hyperparameter sweep for the fine-tuning Zephyr 7B SFT Full using GRPO on the Beavertails dataset.** The plot shows the reward gain as the trained deviates from the base model. Abnormalities in the plotted lines are caused by training runs either plateauing before a KL of 9.14 or from oscillating between a small range of KL values.

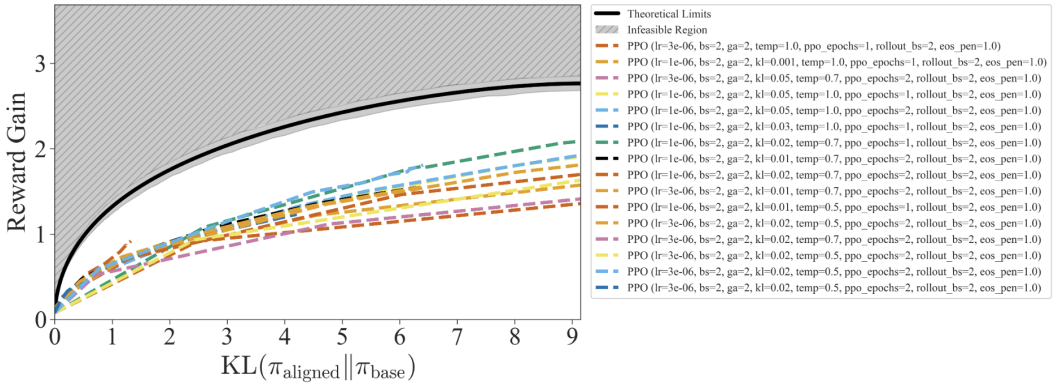


Figure 6: **Hyperparameter sweep for the fine-tuning Zephyr 7B SFT Full using PPO on the Beavertails dataset.** The plot shows the reward gain as the trained deviates from the base model. Abnormalities in the plotted lines are caused by training runs either plateauing before a KL of 9.14 or from oscillating between a small range of KL values.

F Stability of the Alignment Algorithms

The various alignment methods were run over five seeds to ensure that the observed results are stable and not an artifact of the seed. The results are shown in Table 5.

G Licenses of Used Assets

The existing assets used in this paper are listed in Table 6. The assets include publicly available datasets, model checkpoints, and software libraries used for the theoretical validation and empirical evaluation. We cite the original creators of each asset and report the corresponding licenses.

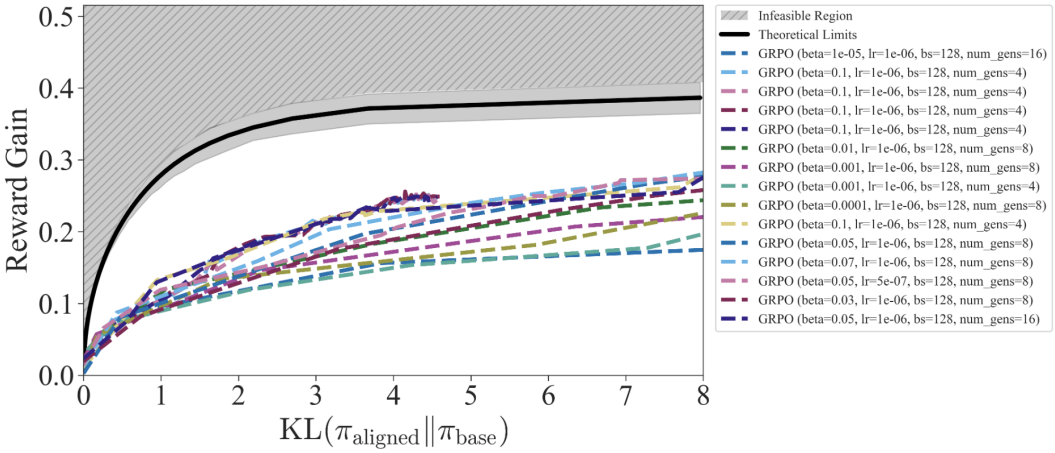


Figure 7: **Hyperparameter sweep for the fine-tuning Pythia 1B SFT using GRPO on the TLDR dataset.** The plot shows the reward gain as the trained deviates from the base model. Abnormalities in the plotted lines are caused by training runs either plateauing before a KL of 7.96 or from oscillating between a small range of KL values.

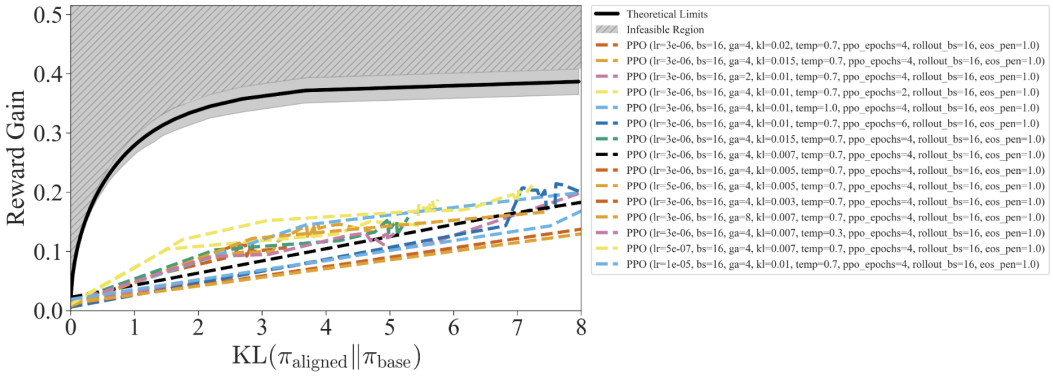


Figure 8: **Hyperparameter sweep for the fine-tuning Pythia 1B SFT using PPO on the TLDR dataset.** The plot shows the reward gain as the trained deviates from the base model. Abnormalities in the plotted lines are caused by training runs either plateauing before a KL of 7.96 or from oscillating between a small range of KL values.

Method	Seed 42	Seed 43	Seed 44	Seed 45	Seed 46
PPO — Beavertails	4.8	5.1	5.0	5.0	5.1
GRPO — Beavertails	3.8	4.1	4.2	4.1	4.2
PPO — TLDR	1.7	2.3	2.2	2.3	2.4
GRPO — TLDR	1.3	2.0	1.8	1.7	1.7

Table 5: **Area under the alignment curve** (AUAC) for PPO and GRPO in the Beavertails and TLDR datasets across five seeds. AUAC is the area under the KL vs. reward gain trade-off curve achieved by the specific algorithm. Results show that findings are not specific to a particular seed.

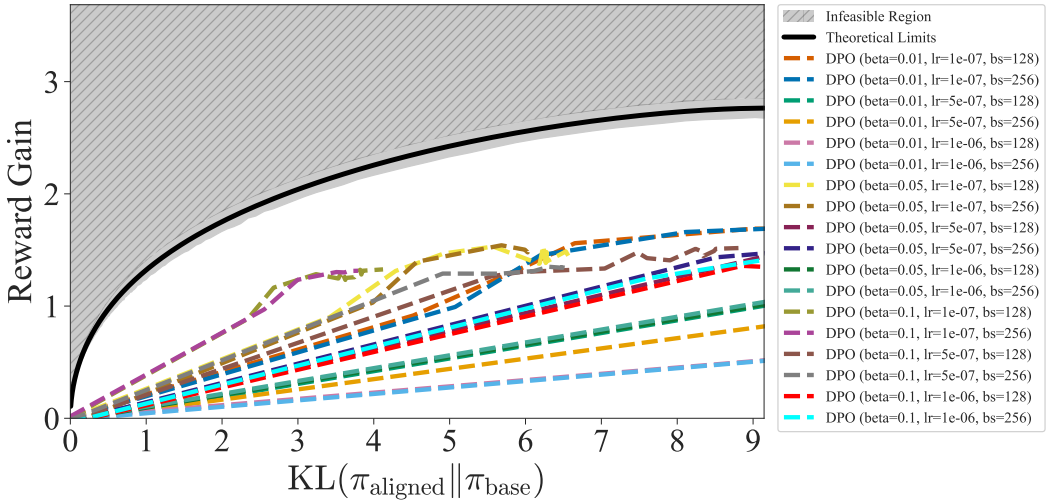


Figure 9: **Hyperparameter sweep for the fine-tuning Zephyr 7B SFT Full on the Beavertails dataset using DPO.** The plot shows the reward gain as the trained model deviates from the base model. Abnormalities in the plotted lines are caused by training runs either plateauing before a KL of 9.14 or from oscillating between a small range of KL values.

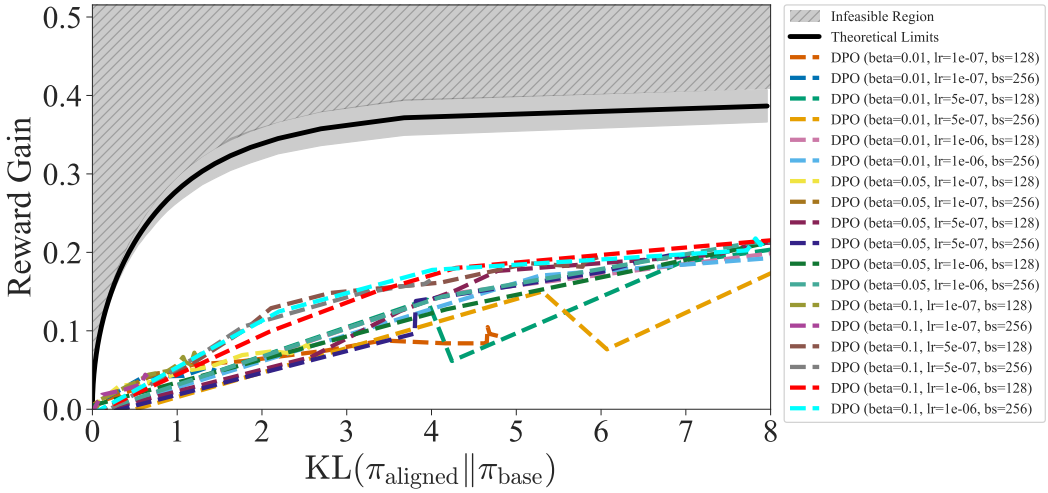


Figure 10: **Hyperparameter sweep for the fine-tuning Pythia 1B SFT on the TLDR dataset using DPO.** The plot shows the reward gain as the trained model deviates from the base model. Abnormalities in the plotted lines are caused by training runs either plateauing before a KL of 7.96 or from oscillating between a small range of KL values.

Table 6: Existing assets used in this work.

Asset	**Type**	**License / Terms**
Reddit TL;DR [51, 47]	Dataset	Modified MIT License
BeaverTails [27]	Dataset	CC BY-NC 4.0
PKU-SafeRLHF [29]	Dataset	CC BY-NC 4.0
Pythia 1B [19, 11]	Model	Apache 2.0
Zephyr-7B [52]	Model checkpoint	Apache 2.0
Mistral-7B-v0.1[30]	Model	Apache 2.0
TRL [54]	Code library	Apache 2.0

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Proof of theoretical results are in App. B.1 and B.2 and the empirical results are in Sec. 5 and App. E.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Sec. 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Proof of theoretical results are in App. B.1 and B.2

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, all hyperparameters are detailed in App. E, experimental details are discussed in Sec. 5, and code will be released with the publication.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code will be released with the paper. No data or models were introduced.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyperparameters are discussed in App. E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: 95% confidence error bars were computed using bootstrap from the python package Seaborn [55].

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Experimental resources were discussed in App. E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, the code of ethics was reviewed.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our paper characterize the fundamental limits of language model alignment. Therefore, our contribution is technical and does not involve societal implications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Assets are cited in the main paper. Additionally, we add licenses and citations for all used assets in App. G.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code accompanying this contribution is well documented.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.