Inference-Time Reward Hacking in Large Language Models

Hadi Khalaf [1] **Claudio Mayrink Verdun** [1] **Alex Oesterling** [1] **Himabindu Lakkaraju** [1 2] **Flavio du Pin Calmon** [1]

Abstract

A common paradigm to improve the performance of large language models is optimizing for a reward model. Reward models assign a numerical score to LLM outputs indicating, for example, which response would likely be preferred by a user or is most aligned with safety goals. However, reward models are never perfect. They inevitably function as proxies for complex desiderata such as correctness, helpfulness, and safety. By overoptimizing for a misspecified reward, we can subvert intended alignment goals and reduce overall performance – a phenomenon commonly referred to as *reward hacking*. In this work, we characterize reward hacking in inference-time alignment and demonstrate when and how we can mitigate it by *hedging* on the proxy reward. We study this phenomenon under Best-of-n (BoN) and Soft-Best-of-n (SBoN), and we introduce **Best-of-Poisson** (BoP) that provides an efficient, near-exact approximation of the optimal reward-KL divergence policy at inference time. We show that the characteristic pattern of hacking as observed in practice (where the true reward first increases before declining) is an inevitable property of a broad class of inference-time mechanisms, including BoN and BoP. To counter this effect, hedging offers a tactical choice to avoid placing undue confidence in high but potentially misleading proxy reward signals. We introduce `HedgeTune`, an efficient algorithm to find the optimal inference-time parameter and avoid reward hacking. We demonstrate through experiments that hedging mitigates reward hacking and achieves superior distortion-reward tradeoffs with minimal computational overhead.

[1]Harvard John A. Paulson School of Engineering and Applied Sciences [2]Harvard Business School. Correspondence to: Hadi Khalaf <hadikhalaf@g.harvard.edu>, Flavio du Pin Calmon <flavio@seas.harvard.edu>.

ICML 2025 Workshop on Models of Human Feedback for AI Alignment, Vancouver, Canada.

1. Introduction

Almost all current alignment methods, including BoN (Stiennon et al., 2020; Nakano et al., 2021), RLHF (Christiano et al., 2017; Bai et al., 2022), DPO (Rafailov et al., 2024), and their variants, aim to maximize a reward function while minimizing divergence from the original model's outputs. It is important to distinguish between two types of rewards: **proxy rewards** which are the computable signals we directly use during alignment (like scores from a trained reward model), and **true (or gold) rewards**, which represent the true, often latent, quality of the model's output according to a desired objective. As the name suggests, proxy rewards are approximations of the true reward and, consequently, of intended alignment goals like correctness, helpfulness, and safety.

A fundamental challenge persists across reward-based alignment methods: **all proxy reward models are imperfect** (Laidlaw et al., 2025). This imperfection stems from multiple factors, including the scarcity of high-quality human-labeled data and the difficulty of formalizing high-level alignment goals into quantifiable metrics (Hadfield-Menell et al., 2017; Pan et al., 2022). For instance, consider AI alignment strategies that aim to promote safety. It is difficult for a single reward model to capture nuanced human user preferences and assign accurate scalar rewards in complex, context-dependent settings where safety specifications conflict (Buyl et al., 2025).

In this work, we analyze and mitigate the impact of misspecified proxy rewards in inference-time alignment methods. Inference-time alignment has emerged as an effective and computationally efficient paradigm to improve the capabilities of large language models and align them with desired goals (Welleck et al., 2024). Among these methods, Best-of-n (BoN) sampling stands out due to its simplicity and effectiveness. Empirically, BoN demonstrates competitive performance, often matching more resource-intensive fine-tuning approaches such as RLHF and DPO (Gao et al., 2023; Mudgal et al., 2024). Additionally, BoN has received an extensive theoretical treatment (Beirami et al., 2024; Huang et al., 2025). BoN can be asymptotically equivalent to RLHF (Yang et al., 2024a), enjoys non-asymptotic guarantees (Mroueh, 2024; Mayrink Verdun et al., 2025), and achieves near-optimal winrate subject to a KL divergence

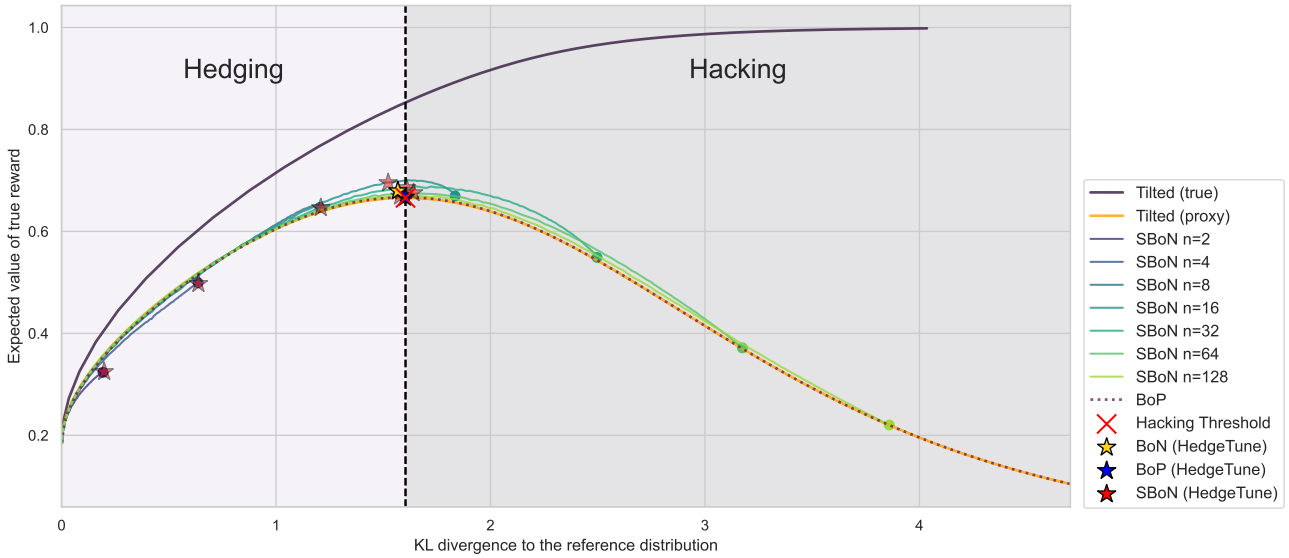


Figure 1. The mismatch between the proxy and gold rewards manifests through the winner's curse. In an ideal world where we could optimize directly on the gold reward, its value would rise monotonically. However, since we are optimizing for a proxy, the gold reward peaks and then collapses. The point at which we find the optimal tradeoff between maximizing reward and minimizing KL divergence from the reference distribution corresponds to the *hacking threshold*. `HedgeTune` successfully recovers the hacking threshold for three inference-time mechanisms: BoN, SBoN, and BoP. In the case of BoN and BoP, `HedgeTune` recovers the optimal number of samples n. As for SBoN, we fix n and find the corresponding inverse-temperature λ that maximizes the true reward. If the hacking threshold is not achievable with any λ, `HedgeTune` returns the best attainable reward, as shown above for low values of n.

constraint (Gui et al., 2024).

Methods like BoN, where multiple samples are generated and the highest-scoring one is chosen, are victims of the **winner's curse** (Capen et al., 1971; Bazerman & Samuelson, 1983; Cox & Isaac, 1984; Thaler, 1988; 2012). In auctions, after each bidder submits an estimate of an item's value, the highest bid typically *overestimates* the true worth, causing the winner to overpay. As Figure 1 demonstrates, inference-time alignment methods, particularly BoN, can overoptimize for proxy rewards: as we sample more candidates and select based on the proxy reward estimate, we increase the chance of choosing outputs where the proxy score significantly overestimates the true quality. This mismatch creates a critical tension: while initially optimizing for a proxy reward improves alignment with true goals, excessive optimization eventually leads to *reward hacking* – also called Goodhart's law (Goodhart & Goodhart, 1984) or goal misgeneralization[1] – where the model exploits the proxy's limitations, leading to worse true performance despite higher proxy scores (Laidlaw et al., 2025; Skalse et al., 2022; Fluri et al., 2024; Kwa et al., 2024; El-Mhamdi & Hoang, 2024). Such misalignment can severely degrade trust and utility, particularly in high-stakes applications (Bondarenko et al., 2025; OpenAI, 2025; Anthropic, 2025).

We mathematically characterize *inference-time reward hacking* (see Theorem 1) and provide a general framework to mitigate it (see Section 4). While this phenomenon has been observed empirically in prior work (Gao et al., 2023; Huang et al., 2025), there has been limited theoretical analysis specific to inference-time methods and ways to mitigate it; see Section 6 for a discussion. As a result, reward hacking for inference-time alignment remains a central challenge in AI alignment. The driving question behind our work is:

When and how can we leverage useful signals from proxy rewards while mitigating hacking?

We focus on answering this question for inference-time alignment methods that sample multiple responses from an LLM and use reward signals to select outputs. We develop principled *hedging techniques against the winner's curse* during inference time that precisely determine until when and how one may leverage proxy signals while preventing overoptimization.

Overview of main results. Our starting point is an optimization formulation at the heart of most alignment methods: finding a distribution π^* that maximizes a (proxy) reward r_p while remaining close (in KL-divergence) to a reference π_{ref}. This is described as the following regularized optimization problem:

$$\pi^* = \underset{\pi_x \in \Delta_{\mathcal{X}}}{\operatorname{argmax}} \; \mathbb{E}_{\pi_x}\left[r_p(X)\right] - \frac{1}{\lambda} D_{\mathsf{KL}}(\pi_x \| \pi_{\text{ref}}) \quad (1)$$

[1] See (Weng, 2024; Amodei et al., 2016) and Section 6 for a discussion of the terminology.

Consider the information-theoretic regime where all distributions are known exactly. The solution of the above objective (1) is the exponential tilting of the reference distribution using the proxy reward (Csiszár et al., 2004). Though theoretically interesting, tilted distributions are impossible to realize in practice since computing the normalizing constant and drawing unbiased samples are computationally prohibitive. Some attempts have been made to approximate this solution at inference time, i.e., when only samples from π_{ref} and black-box access to r_p are available. A notable example is Soft Best-of-n (SBoN) (Mayrink Verdun et al., 2025). In this work, we show that SBoN is an effective strategy for hedging against reward hacking due to its temperature parameter λ, which allows us to smoothly interpolate between aggressive exploitation of the proxy reward and conservative adherence to the reference distribution; see Section 5. However, this comes at the expense of having two tunable parameters (n, λ), which can be difficult to set in practice.

This motivates us to propose a new inference-time alignment strategy called **Best-of-Poisson (BoP)**. The idea behind BoP is simple: we run BoN with the number of samples n chosen according to a Poisson distribution. We prove that BoP achieves a near-optimal reward-distortion tradeoff **at inference**. Using a single tunable parameter, BoP can approximate the optimal proxy reward-tilted solution with KL gap of order 10^{-3} when rewards are uniformly distributed, allowing to span the entire reward-distortion region at inference (see Figure 2 and Theorem 7). BoP can serve as a computationally efficient stand-in for the optimal tilted distribution with negligible loss in KL-reward tradeoff.

In practice, hedging translates to selecting parameters of inference-time alignment methods to avoid overoptimization to a proxy reward. To do so, we introduce `HedgeTune`: an algorithm for tuning parameters in BoN, SBoN, and BoP in order to hedge against hacking (see Algorithm 4). We illustrate the benefit of hedging in Figure 1, where we plot the expected value of the true reward versus the distortion with respect to the reference distribution for various inference-time alignment methods. If we had access to the true reward, the optimal solution would be the tilting of the reference distribution via the true reward, leading to the reward-distortion Pareto frontier (purple curve in Figure 1). However, as we are tilting via the proxy reward, we suffer from the winner's curse: the true reward (orange curve in Figure 1) increases at first and then collapses. This behavior also manifests in BoN, as seen in the dotted points. Hedging allows us to find the *hacking threshold*: the parameters of inference-time alignment methods that yield the best tradeoff between (true) reward and distortion relative to the base model.

Our contributions are as follows:

- We mathematically formalize inference-time reward hacking (Definition 1) and derive conditions when overoptimizing imperfect proxy rewards inevitably leads to performance degradation (Theorem 1, Corollary 3).
- We introduce Best-of-Poisson (BoP), a novel inference-time alignment method (Algorithm 3). For uniformly distributed rewards, BoP approximates the optimal tilted distribution with negligible KL divergence gap (Theorem 7).
- We develop `HedgeTune`, a principled hedging framework that mitigates reward hacking by finding the optimal inference-time parameters (Algorithm 4). We empirically demonstrate that hedging strategies significantly outperform standard BoN sampling with minimal computational overhead (Section 5).

2. Inference-Time Reward Hacking

In this section, we formalize inference-time reward hacking and show its inevitability under methods like BoN.

Notation and Technical Assumptions. Let $\mathcal{X}$ be a finite alphabet of tokens and let π be a probability mass function (PMF) over token sequences $x \in \mathcal{X}^*$. We denote the probability simplex over all finite sequences of tokens as $\Delta_{\mathcal{X}^*}$. Let $\pi_{\text{ref}} \in \Delta_{\mathcal{X}^*}$ be the reference policy, typically a supervised fine-tuned (SFT) model. Let $r_p : \mathcal{X}^* \to \mathbb{R}$ be a proxy reward function that assigns a unique scalar value to each sequence. This is the reward we use to optimize π_{ref} during inference-time alignment. An inference-time alignment method parameterized by θ transforms the base policy π_{ref} into a new distribution π_θ. Here, θ **is the parameter of the alignment method itself** (e.g., the number of candidates n in Best-of-n). The performance of the inference-time aligned distribution is measured using a true (or gold) reward r_t. Primarily, we are interested in the expected value of the true reward r_t under the resulting distribution. We denote our measure of interest as $f(\theta) = \mathbb{E}_{U\sim\pi_\theta}[r_t(U)]$.

For theoretical tractability, we adopt the standard assumption that proxy rewards can be transformed to have a uniform distribution, stated next. This assumption is widely used in the theoretical analysis of alignment (Beirami et al., 2024; Gui et al., 2024; Balashankar et al., 2025) and allows us to restrict our analysis to the space of proxy reward values instead of the high-dimensional, discrete space of sequences $\mathcal{X}^*$.

Assumption 1 (Uniform Reward Mapping)**.** The proxy rewards $r_p(x)$ obtained by sampling sequences from the reference policy, $x \sim \pi_{\text{ref}}$, are uniformly distributed over $[0, 1]$.

Assuming uniformly distributed proxy rewards incurs little

Algorithm 1 Best-of-n Sampling (BoN)

Input: Integer $n \geq 1$, base policy π_{ref}
Draw n samples $X_1, \ldots, X_n$ i.i.d. from π_{ref}
Compute proxy rewards $R_i = r_p(X_i)$ for all i
Select $j = \arg\max_{i \in \{1,\ldots,n\}} r_p(X_i)$
Return: $Y = X_j$

Algorithm 2 Soft Best-of-n Sampling (SBoN)

Input: Integer $n \geq 1$, inverse temperature $\lambda > 0$, base policy π_{ref}
Draw n samples $X_1, \ldots, X_n$ i.i.d. from π_{ref}
Compute proxy rewards $R_i = r_p(X_i)$ for all i
Sample index $Z \in \{1, \ldots, n\}$ with probability

$$\Pr(Z = i) = \frac{e^{\lambda\, r_p(X_i)}}{\sum_{j=1}^{n} e^{\lambda\, r_p(X_j)}}$$

Return: $Y = X_Z$

loss of generality. In practice, the proxy reward distribution can be made approximately uniform by applying an inverse probability integral transform (Beirami et al., 2024). To sample a sequence from π_{ref}, we first sort the space of all sequences by the proxy reward $r_p(x)$. We then select a sequence by mapping a uniform draw $u \sim \text{Unif}(0, 1)$ to the corresponding quantile of a reference distribution π_{ref} defined over this sorted space. While the policy π_{ref} is itself complex and non-uniform, Assumption 1 allows us to only consider the continuous uniform distribution of the proxy reward.

Inference-time alignment. The core challenge we address is the mismatch between the proxy reward r_p and the true reward r_t. Discrepancies between these two rewards can be exploited, leading to reward hacking. We focus on the family of inference-time methods that first sample a pool of candidate outputs and then use their proxy reward scores to define the selection mechanism. Two examples from this family are:

1. **Best-of-n** (see Algorithm 1). BoN places all probability mass on the sample with the highest proxy reward.
2. **Soft Best-of-n** (see Algorithm 2). SBoN is a generalization of BoN recently proposed by (Mayrink Verdun et al., 2025). It applies a temperature-scaled softmax over candidate scores. As $\lambda \to 0$, SBoN sampling approaches uniform selection among the n candidates. As $\lambda \to \infty$, SBoN converges to standard BoN.

We first formalize inference-time reward hacking through the following definition.

Definition 1 (Inference-Time Reward Hacking)**.** Let π_θ be a distribution induced by an inference-time alignment method with parameter θ, where we assume increasing θ increases both the expected proxy reward $\mathbb{E}_{X\sim\pi_\theta}[r_p(X)]$ and the KL-divergence $D_{\mathsf{KL}}(\pi_\theta \| \pi_{\text{ref}})$. We say that *inference-time reward hacking* occurs when there exists a threshold $\theta^\dagger$ such that for $\theta > \theta^\dagger$, $\mathbb{E}_{X\sim\pi_{\theta^\dagger}}[r_t(X)] > \mathbb{E}_{X\sim\pi_\theta}[r_t(X)]$ (i.e., the true reward decreases), despite the proxy reward and KL-divergence continuing to increase. The largest value of $\theta^\dagger$ for which this holds is called the *hacking threshold*.

Definition 1 offers a concrete basis for operationalizing and measuring the winner's curse for inference time methods. The hacking threshold is the ideal operating parameter for an inference-time alignment method: increasing θ beyond the hacking threshold incurs distortion in a model's output, without a gain in true reward in return (see Figure 1).

The following theorem establishes that under common conditions, the shape of $f(\theta)$ is well-behaved: it either varies monotonically or reaches exactly one extremum.

Theorem 1 (Inevitability of Reward Hacking)**.** Let $\{\pi_\theta\}_{\theta\in\Theta\subset\mathbb{R}}$ be a family of distributions with density $p_\theta(x)$ on a common support $\mathcal{X}$ such that **(i)** $p_\theta(x)$ is *strictly totally positive of order 2* (TP_2) in (θ, x), and **(ii)** its score function $\psi(x, \theta) := \partial_\theta \log p_\theta(x)$ is continuous in x and *strictly increasing* in x for each fixed θ. For any bounded, non-negative true reward $r_t : \mathcal{X} \to [0, \infty)$ define

$$f(\theta) \;:=\; \mathbb{E}_{X\sim\pi_\theta}[r_t(X)]$$

Then f is either monotone in θ or possesses a single **unique** interior extremum $\theta^\dagger$.

Corollary 1 (Inevitability of Reward Hacking for Strictly MLR densities)**.** Let $p_\theta(x)$ be a *strictly monotone–likelihood–ratio* in x. If the score function $\psi(x, \theta) = \partial_\theta \log p_\theta(x)$ is strictly increasing in x, then Theorem 1 applies. In particular, this applies to Best-of-n, Best-of-Poisson (to be introduced in Sec. 3,) and to any canonical distribution from the exponential family with strictly monotone statistic and strictly monotone natural parameter.

Four scenarios may occur: (i) *monotonic improvement:* true reward continuously increases with optimization strength; (ii) *reward hacking:* true reward initially improves but deteriorates beyond a critical threshold; (iii) *reward grokking:* true reward initially declines but then improves beyond a critical threshold; (iv) *immediate decline:* any optimization immediately harms true performance. We describe exactly when each regime occurs for MLR densities in Corollary 3.

The unimodality of the true reward function renders the problem of locating the optimal operating point $\theta^\dagger$ algorithmically tractable. To implement this insight, we develop hedging strategies that balance exploitation of the proxy reward against fidelity to the reference distribution. Each inference-time method offers a parameter controlling this proxy reward-KL tradeoff: n in BoN, λ in SBoN (for a fixed

Algorithm 3 Best-of-Poisson Sampling (BoP)

Input: Poisson parameter $\mu > 0$, base policy π_{ref}
Sample $n' \sim \text{Poisson}(\mu)$ and set $n = n' + 1$
Draw $X_1, \ldots, X_n$ i.i.d. from π_{ref}
Compute proxy rewards $R_i = r_p(X_i)$ for all i
Select $j = \arg\max_{i \in \{1,\ldots,n\}} r_p(X_i)$
Return: $Y = X_j$

n), and μ in BoP (introduced in Section 3). Before introducing methods for tuning inference-time alignment methods, we first introduce **Best-of-Poisson** sampling – an alternative to BoN that approximates the optimal tilted distribution.

3. Best-of-Poisson: Approximating the Optimal Reward

While Soft Best-of-n offers a principled approach to mitigate reward hacking, it requires tuning both the number of samples n and the temperature parameter λ. In this section, we introduce Best-of-Poisson (BoP) (Algorithm 3), that is provably close to the solution of (1) with a single tunable parameter (Figure 2). BoP is of independent interest as it provides a mathematically elegant and computationally efficient way to near-optimally span the entire reward-KL distortion region with a single parameter. The key insight behind BoP is to replace the fixed sample size n in BoN with a random sample size drawn from a Poisson distribution.

The parameter μ in BoP controls the expected number of samples, analogous to how n functions in BoN. We first sample n' from a Poisson distribution parameterized by μ and set $n = n' + 1$ to ensure at least one sample is generated. Under Assumption 1, the BoP distribution with parameter μ has a probability density function (Appendix B)

$$q_\mu(x) = (\mu x + 1)e^{\mu(x-1)} \text{ for } x \in [0, 1]$$

The following theorem characterizes the KL divergence and expected value of BoP:

Theorem 2 (KL Divergence and Expected Value of BoP)**.** Let X_{BoP} be the random variable representing the response selected by BoP with parameter μ. Then:

$$\text{KL}(\pi_{\text{BoP}} \| \pi_{\text{ref}}) = \frac{e^{-\mu-1}(\text{Ei}(\mu+1) - \text{Ei}(1))}{\mu} + \log(\mu+1) - 1 \quad (2)$$

$$\mathbb{E}[X_{\text{BoP}}] = 1 - \frac{1}{\mu} + \frac{1 - e^{-\mu}}{\mu^2} \quad (3)$$

where $\text{Ei}(z) = -\int_{-z}^{\infty} \frac{e^{-t}}{t} dt$ is the exponential integral function.

What makes BoP particularly valuable is its ability to closely approximate the solution of (1), i.e., the optimal KL-constrained tilted distribution with parameter $\lambda > 0$,

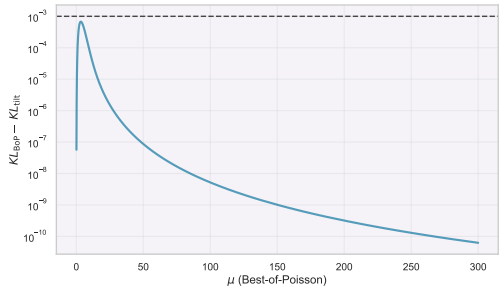


Figure 2. KL divergence gap between BoP and the optimal tilted distribution with respect to the reference distribution. The plot shows the difference in KL divergence when both distributions are matched to produce the same expected reward. The extremely small gap (of order 10^{-3}) demonstrates that BoP approximates the optimal distribution with negligible performance loss.

defined as $\pi^*_\lambda(x) = \frac{\pi_{\text{ref}}(x)e^{\lambda r_p(x)}}{Z(\lambda)}$, where $Z(\lambda)$ is the normalization constant. While this distribution is theoretically optimal for balancing reward and divergence, computing it is intractable. To draw a next token from the tilted π^*_λ for an autoregressive LLM, one would have to compute

$$\pi^*_\lambda(x_{t+1} \mid x_{\leq t}) = \frac{\pi_{\text{ref}}(x_{\leq t+1})\, e^{\lambda r_p(x_{\leq t+1})}}{\sum_{x'} \pi_{\text{ref}}(x_{\leq t} x')\, e^{\lambda r_p(x_{\leq t} x')}},$$

where the denominator sums over every possible continuation of the prefix $x_{\leq t}$. Because the space of continuations grows exponentially with the remaining sequence length, evaluating this denominator (and hence sampling a single token) is computationally prohibitive for LLMs. Our analysis (Appendix B) shows that BoP provably achieves nearly identical performance to this optimal distribution with minimal KL divergence gap. As illustrated in Figure 2, the KL divergence gap between these distributions is remarkably small. Numerical evaluation confirms that for all μ, if $\lambda > 0$ is chosen so that $\mathbb{E}_{X \sim \pi_{\text{BoP}}}[r_p(X)] = \mathbb{E}_{X \sim \pi^*_\lambda}[r_p(X)]$, then the KL-gap is bounded between 0 and 7×10^{-3}.

This near-equivalence means that BoP can serve as a practical stand-in for the theoretically optimal tilted distribution. This result has two consequences. First, hedging in the optimal tilted distribution is almost equivalent to hedging with BoP. Second, (1) represents the solution to the standard RLHF optimization problem, which implies that BoP is an inference-time approximation to RLHF and allows us to use BoP to easily traverse between policies rather than having to fine tune a new model for each λ of interest. In the next section, we turn to the question of how to choose the right parameter value to avoid reward hacking.

4. Hedging to mitigate reward hacking

In this section, we develop a unified framework for choosing the inference-time parameter θ in order to maximize the expected true reward and avoid hacking. The main limitation is that we require black-box access to the true reward to perform a *one-time calibration* of the parameter θ. This is practical in several common scenarios. One may opt to use an LLM-as-a-judge or a more powerful but computationally prohibitive reward model (Zheng et al., 2023; Lambert et al., 2024). Assume we are given the proxy and true reward scores for a set of query-response pairs. By first constructing an empirical CDF over generated proxy reward scores, we transform these proxy scores to have a uniform distribution as explained in Section 2. We denote the transformed proxy reward as U.

Each sampling method (BoN, SBoN, and BoP) induces a distribution π_θ over proxy-percentiles $u \in [0,1]$, where θ is the corresponding parameter (sample size n, inverse-temperature λ, or Poisson rate μ). Since we know by Theorem 1 that the expected true reward has at most one peak, our key insight is to create the precise **hedge** against hacking by finding the parameter value where the marginal benefit of increasing proxy reward equals zero. We present the following conditions that the hacking threshold must satisfy for each of the three inference-time methods.

Theorem 3 (Hacking Threshold Characterization)**.** Let r_t be a true reward and $\theta^\dagger$ be the hacking threshold from Definition 1. For each inference-time method, $\theta^\dagger$ is characterized by the following conditions:

For BoN, $n^\dagger$ satisfies:

$$\nabla_\alpha \mathbb{E}_{u\sim\text{Beta}(\alpha,1)}[r_t(u)] = \int_0^1 r_t(u)\,(1+n^\dagger \ln u)\,u^{n^\dagger-1}\,du = 0 \quad (4)$$

For SBoN, $\lambda^\dagger$ satisfies:

$$\nabla_\lambda \mathbb{E}_{u\sim f_\lambda}[r_t(u)] = \text{Cov}_{u\sim f_{\lambda^\dagger}}(r_t(u), u) = 0 \quad (5)$$

For BoP, $\mu^\dagger$ satisfies:

$$\nabla_\mu \mathbb{E}_{u\sim f_\mu}[r_t(u)] = \mathbb{E}_{u\sim f_{\mu^\dagger}}\Big[r_t(u)\Big(u-1+\frac{u}{\mu^\dagger u+1}\Big)\Big] = 0 \quad (6)$$

The proof can be found in Appendix C. Consequently, we provide `HedgeTune` (Algorithm 4), an algorithm that numerically solves the corresponding root-finding problem to determine the optimal inference-time parameter for BoN, SBoN, or BoP. **Note that we do not need access to the LLM distribution itself**. Given the *score* function ψ of π_θ, it defines a residual function $R(\theta) = \mathbb{E}[r_t(u)\psi(u,\theta)]$ which captures the alignment between the true reward and the proxy-weighted score. The optimal parameter $\theta^\dagger$ is found efficiently as the root of this function using standard methods such as bisection or Newton's method (Quarteroni et al., 2006).

Algorithm 4 `HedgeTune`: Parameter Optimization for Hedging

Input: number of samples M, a set of proxy and true reward scores $\{r_p^i, r_t^i\}_{i=1}^M$, method $m \in \{$BoN, SBoN, BoP$\}$, sample size n (for SBoN)
Output: optimal $\theta^\dagger$
Transform the proxy scores using the empirical CDF to obtain the set $\{u_i, r_t(u_i)\}_{i=1}^M$
if $m =$ SBoN **then** sample $u_{1:n}$; compute $P_k \propto e^{\theta u_k}$; draw $J \sim P$, and set $u \leftarrow u_J$. Repeat M times to get a modified $\{u_i, r_t(u_i)\}_{i=1}^M$
if $m =$ BoN **then** set $\psi \leftarrow 1/\theta + \ln u$; $p_\theta \leftarrow \theta u^{\theta-1}$
if $m =$ SBoN **then** set $\psi \leftarrow u - \sum_k P_k u_k$; $p_\theta \leftarrow P_J$
if $m =$ BoP **then** set $\psi \leftarrow u - 1 + \frac{u}{\theta u+1}$; $p_\theta \leftarrow (\theta u+1)e^{\theta(u-1)}$
Set $R(\theta) = \sum_i r_t(u_i)\psi(u_i,\theta)p_\theta(u_i)\Delta u$ and find its root $\theta^\dagger$
if $m =$ BoN **then** $\theta^\dagger \leftarrow \text{round}(\theta^\dagger)$

5. Experiments: Hedging in Practice

In this section, we provide numerical results showcasing hedging as an effective tool against reward hacking. In particular, we validate that hedging can provide superior reward-distortion tradeoffs in two experimental setups: a controlled synthetic setting where the relationship between the gold and proxy is known, and a more realistic RLHF scenario that reflects practical deployment conditions.

Synthetic Setup. To empirically observe reward hacking, we miscalibrate the proxy RMs at the extreme high end of their scores. We use Pythia-1.4B (Biderman et al., 2023) to generate a dataset of 60,000 responses for a prompt from the TL;DR dataset. Another Pythia-1.4B model finetuned on TL;DR preference labels serves as our proxy reward model. Given how reward models are trained in practice, having one prompt ensures that the reward scores are always comparable across all responses. Consider P_α be the top α-percentile of proxy scores. For responses with scores in P_α, we consider a negatively linear relationship between the proxy and true scores. Otherwise, we set the proxy to be the same as true reward. We vary the threshold α to simulate different thresholds at which the proxy becomes misaligned with the true reward.

Findings. As seen in Figure 3, we observe reward hacking: the expected gold reward for Best-of-n (BoN) sampling and Best-of-Poisson (BoP) sampling eventually decrease as the number of samples n increases. Note that for BoP with given μ, the average number of samples is $n = \mu + 1$. The optimal hacking thresholds are then found using `HedgeTune` as marked on the figure. Subsequently, we show that Soft Best-of-n (SBoN), using an optimally chosen

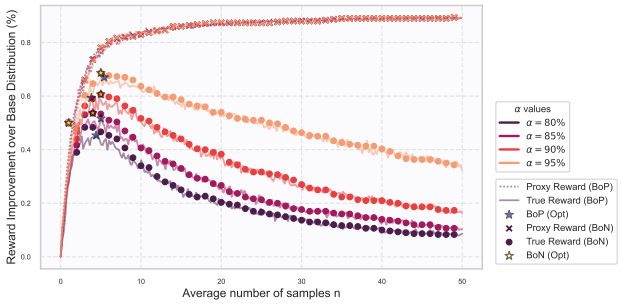


Figure 3. Reward hacking manifests: proxy reward increases for both BoN and BoP while true reward first increases and then collapses as a function of the average number of samples n.

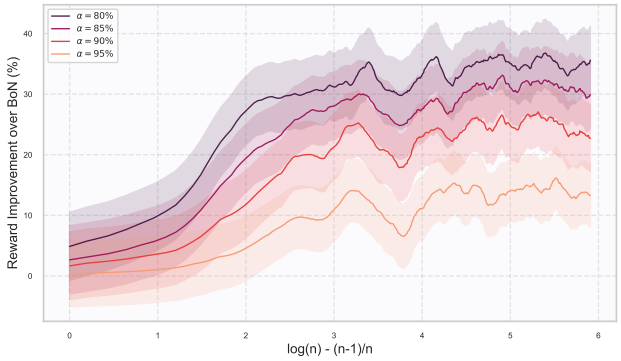


Figure 4. Improvement in expected value of true reward by using the optimal inverse temperature $\lambda^\dagger$ in soft Best-of-n compared to the expected value of true reward using Best-of-n

temperature $\lambda^\dagger$, mitigates this hacking and considerably improves the true reward compared to BoN. As expected, the degree of hacking is controlled by how misaligned the proxy and gold are. Even if this disagreement happens on the top 5% of responses out of $60,000$ responses, an optimized hedging scheme will correspond to 10% increase in expected reward compared to greedy Best-of-n starting at $n \approx 50$. Results illustrating this are presented in Figure 4.

Hacking in the wild setup. In the following section, we train smaller reward models to act as proxies on labeled preference data. Our experimental design follows the methodology of Coste et al. (Coste et al., 2024) and Gao et al. (Gao et al., 2023), wherein proxy reward models are trained using preferences of a fixed gold reward model. In many real-world cases, we do not have access to the gold reward and instead have access to preference data. However, as we demonstrate below, a favorable operating point can still be found using traditional hyperparameter search.

Models. As a reference model, we use a 1.4B Pythia model (Biderman et al., 2023) fine-tuned on AlpacaFarm dataset, but without any subsequent alignment (e.g., RLHF or DPO). This reference model is used to generate responses. We use **AlpacaRM** (Dubois et al., 2023) as our gold reward model. **AlpacaRM** is an established reward model trained on human preference data and has been adopted in prior work on reward model evaluation (Coste et al., 2024; Xu et al., 2025; Zeng et al., 2023). This model serves as the ground truth for generating preference labels for training proxy reward models. As for proxy rewards, we use the setup of Coste et al. (Coste et al., 2024) where we train Pythia 44m models.

Datasets. We use the `tlc4418/gold_labelled_gens` dataset from Coste et al. (Coste et al., 2024). This dataset comprises of 12,600 responses generated by the Pythia 1.4B base policy for each of 1,000 prompts. The prompts are sourced from the validation split of the AlpacaFarm dataset (Dubois et al., 2023). Each generated response in this dataset is scored by **AlpacaRM**. To train proxy reward models, we construct preference datasets using the `tlc4418/gold_labelled_gens` scores. For each training instance, we sample a pair of responses to a given prompt and label them based on their **AlpacaRM** scores.

Training. The proxy RMs are trained using a standard binary cross-entropy loss on preference pairs. We train proxy RMs on preference pair datasets of varying sizes: 10k, 20k, 46k, and 80k. In line with (Coste et al., 2024; Miao et al., 2024; Yang et al., 2024b), we simulate disagreements in human annotators by considering two cases: (a) no label noise in the preferences, and (b) 25% label noise. All proxy RM training runs are repeated across 4 random seeds each. We present some of the runs with 25% label noise in Figure 5 and we present the remaining results in Appendix D, along with other hyperparameters and training details. Post training, we use each proxy model to score a set of 800 prompts, with 12,600 responses each.

Findings. We apply BoN, SBoN, and BoP on each run and find the expected value of the true reward as a function of n. When reward hacking manifests, we find a hacking threshold for BoN and BoP that maximizes their reward. For SBoN, with a selected $\lambda^\dagger$, we attain the peak value without suffering from reward hacking. Meanwhile, if the proxy is always at odds with the true reward, the optimal solution is the reference distribution itself, corresponding to $\lambda = 0$.

6. Related Work

Reward Hacking. Reward hacking has been widely studied in RL literature (Pan et al., 2022; Hadfield-Menell et al., 2017; Karwowski et al., 2024), also under the name misspecification (Amodei et al., 2016), goal misgeneralization (Shah et al., 2022), or specification gaming (Krakovna et al., 2020). In the context of LLMs, overoptimization has been

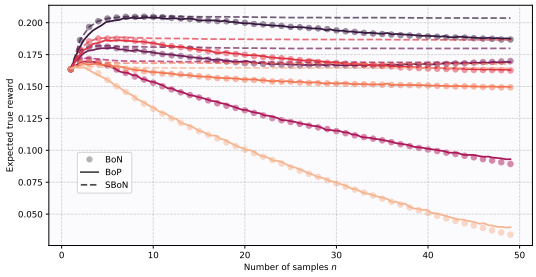


Figure 5. Use of three inference-time methods (BoN, SBoN, and BoP) on trained proxy rewards. Hacking is effectively mitigated by hedging via λ in SBoN or by early stopping in BoN and BoP.

referred to as reward hacking or Goodhart's Law (Goodhart & Goodhart, 1984; Gao et al., 2023; Kwa et al., 2024; El-Mhamdi & Hoang, 2024). Hacking behavior has been found to manifest in unwanted or surprising behavior (Denison et al., 2024; Chen et al., 2024) across a variety of tasks (Pan et al., 2024; Gao et al., 2023; Huang et al., 2025). Prior works have proposed various formulations of reward hacking based on true performance behavior (Skalse et al., 2022), correlation between proxy and true reward (Laidlaw et al., 2025), or distribution shift (Fluri et al., 2024). Huang et al. (2025) prove that BoN alignment provably suffers from reward hacking when the number of samples n is large.

To address inference time hacking, a variety of methods have been explored to varying success, such as ensembling (Coste et al., 2024; Eisenstein et al., 2024; Ahmed et al., 2024; Rame et al., 2024), regularization (Ichihara et al., 2025) or rejection sampling (Huang et al., 2025). Puri et al. (2025) simulate n particles resampled using a softmax reward to improve performance of reasoning models, similar to SBoN over reasoning steps. However, all methods suffer from some combination of additional generation cost beyond generating n samples and estimation of additional side-quantities such as KL-divergence or χ^2-divergence. Additionally, a variety of approaches have been proposed to mitigate reward hacking during RLHF finetuning such as regularization (Rashidinejad & Tian, 2024; Yang et al., 2024b; Liu et al.; Miao et al., 2025), χ^2-divergence (Huang et al.; Laidlaw et al., 2025), uncertainty estimation (Zhang et al., 2024b), and reward pessimism (Zhu et al., 2024) although (Kwa et al., 2024) demonstrated that RLHF can still result in reward hacking under heavy-tailed reward mismatch. Finally, prior works have also focused on improving reward models to prevent mismatch and reduce hacking (Chen et al., 2025; Shen et al., 2023; Fu et al., 2025; Liu et al., 2024; Miao et al., 2024; Wang et al., 2025).

Best-of-n. Best-of-n sampling is a simple inference-time approach for alignment (Stiennon et al., 2020; Nakano et al., 2021; Hilton et al., 2022). Prior results have characterized the expected reward gap and KL divergence between BoN sampling and the reference model and have demonstrated that BoN is asymptotically equivalent to KL-constrained reinforcement learning (Beirami et al., 2024; Yang et al., 2024a; Mroueh, 2024). There have been various methodological improvements on BoN sampling. One such improvement is to reduce the cost of sampling n sequences via tree-based or speculative search (Qiu et al., 2024; Zhang et al., 2024a) . Additionally, (Gui et al., 2024; Sessa et al., 2024; Touvron et al., 2023; Amini et al., 2024; Yang et al.) distill the BoN sampling distribution into a model via fine-tuning. Finally, (Balashankar et al., 2025; Chow et al., 2024) propose inference-aware methods to improve BoN. Other works focus on improving the reward model through self-training (Pace et al., 2024). In this work, we focus on a variant of BoN, Soft-Best-of-n (Mayrink Verdun et al., 2025), which allows for finer control between sampling from the base model and the reward-maximizing generation.

7. Conclusion

Our work tackles the fundamental challenge that all proxy rewards are imperfect, yet they remain essential for guiding AI systems. We establish a theoretical framework proving the inevitability of reward hacking in inference-time alignment and introduce practical hedging strategies to mitigate its harmful effects. By developing *Best-of-Poisson* sampling which achieves near-optimal reward-distortion tradeoffs with a single parameter and the `HedgeTune` algorithm for precisely calibrating inference methods, we enable practitioners to extract valuable signals from proxy rewards without falling prey to Goodhart's law. Ultimately, this work demonstrates that principled hedging is a promising direction for building safer, more reliable AI systems.

Acknowledgments

This work is supported by the National Science Foundation under grants CIF 2312667, FAI 2040880, and CIF 2231707. AO is supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2140743.

References

Ahmed, A. M., Rafailov, R., Sharkov, S., Li, X., and Koyejo, S. Scalable Ensembling For Mitigating Reward Overoptimisation, June 2024. URL http://arxiv.org/abs/2406.01013. arXiv:2406.01013 [cs].

Amini, A., Vieira, T., and Cotterell, R. Variational best-of-n alignment. *arXiv preprint arXiv:2407.06057*, 2024.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Anthropic. Claude 3.7 sonnet system card, July 2025. URL https://assets.anthropic.com/m/785e231869ea8b3b/original/claude-3-7-sonnet-system-card.pdf. Technical Report.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Balashankar, A., Sun, Z., Berant, J., Eisenstein, J., Collins, M., Hutter, A., Lee, J., Nagpal, C., Prost, F., Sinha, A., Suresh, A. T., and Beirami, A. InfAlign: Inference-aware language model alignment, February 2025. URL http://arxiv.org/abs/2412.19792. arXiv:2412.19792 [cs].

Bazerman, M. H. and Samuelson, W. F. I won the auction but don't want the prize. *Journal of conflict resolution*, 27(4):618–634, 1983.

Beirami, A., Agarwal, A., Berant, J., D'Amour, A., Eisenstein, J., Nagpal, C., and Suresh, A. T. Theoretical guarantees on the best-of-n alignment policy. *arXiv preprint arXiv:2401.01879*, 2024.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Bondarenko, A., Volk, D., Volkov, D., and Ladish, J. Demonstrating specification gaming in reasoning models. *arXiv preprint arXiv:2502.13295*, 2025.

Buyl, M., Khalaf, H., Mayrink Verdun, C., Monteiro Paes, L., Machado, C. C. V., and Calmon, F. d. P. AI Alignment at Your Discretion. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. ACM, 2025.

Capen, E. C., Clapp, R. V., and Campbell, W. M. Competitive bidding in high-risk situations. *Journal of petroleum technology*, 23(06):641–653, 1971.

Chen, L., Zhu, C., Soselia, D., Chen, J., Zhou, T., Goldstein, T., Huang, H., Shoeybi, M., and Catanzaro, B. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319*, 2024.

Chen, Y., Liu, Y., Wang, X., Yu, Q., Huzhang, G., Zeng, A., Yu, H., and Zhou, Z. Establishing reliability metrics for reward models in large language models. *arXiv preprint arXiv:2504.14838*, 2025.

Chow, Y., Tennenholtz, G., Gur, I., Zhuang, V., Dai, B., Thiagarajan, S., Boutilier, C., Agarwal, R., Kumar, A., and Faust, A. Inference-Aware Fine-Tuning for Best-of-N Sampling in Large Language Models, 2024. URL http://arxiv.org/abs/2412.15287.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Coste, T., Anwar, U., Kirk, R., and Krueger, D. Reward Model Ensembles Help Mitigate Overoptimization, March 2024. URL http://arxiv.org/abs/2310.02743. arXiv:2310.02743 [cs].

Cox, J. C. and Isaac, R. M. In search of the winner's curse. *Economic Inquiry*, 22(4):579–592, 1984.

Csiszár, I., Shields, P. C., et al. Information theory and statistics: A tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4):417–528, 2004.

Denison, C., MacDiarmid, M., Barez, F., Duvenaud, D., Kravec, S., Marks, S., Schiefer, N., Soklaski, R., Tamkin, A., Kaplan, J., Shlegeris, B., Bowman, S. R., Perez, E., and Hubinger, E. Sycophancy to Subterfuge: Investigating Reward-Tampering in Large Language Models, 2024. URL http://arxiv.org/abs/2406.10162.

Dubois, Y., Li, C. X., Taori, R., Zhang, T., Gulrajani, I., Ba, J., Guestrin, C., Liang, P. S., and Hashimoto, T. B. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

Eisenstein, J., Nagpal, C., Agarwal, A., Beirami, A., D'Amour, A. N., Dvijotham, K. D., Fisch, A., Heller, K. A., Pfohl, S. R., Ramachandran, D., et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. In *First Conference on Language Modeling*, 2024.

El-Mhamdi, E.-M. and Hoang, L.-N. On goodhart's law, with an application to value alignment. *arXiv preprint arXiv:2410.09638*, 2024.

Fluri, L., Lang, L., Abate, A., Forré, P., Krueger, D., and Skalse, J. The perils of optimizing learned reward functions: Low training error does not guarantee low regret. *arXiv preprint arXiv:2406.15753*, 2024.

Fu, J., Zhao, X., Yao, C., Wang, H., Han, Q., and Xiao, Y. Reward Shaping to Mitigate Reward Hacking in RLHF, 2025. URL http://arxiv.org/abs/2502.18770.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Goodhart, C. A. and Goodhart, C. *Problems of monetary management: the UK experience*. Springer, 1984.

Gui, L., Garbacea, C., and Veitch, V. 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.

Hadfield-Menell, D., Milli, S., Abbeel, P., Russell, S. J., and Dragan, A. Inverse Reward Design. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/32fdab6559cdfa4f167f8c31b9199643-Abstract.html.

Hilton, J., Clark, P., et al. Measuring goodhart's law: Towards an evaluation framework for open-ended generative models. OpenAI Blog, 2022. URL https://openai.com/index/measuring-goodharts-law. Accessed: 2025-01-30.

Huang, A., Zhan, W., Xie, T., Lee, J. D., Sun, W., Krishnamurthy, A., and Foster, D. J. Correcting the mythos of kl-regularization: Direct alignment without overoptimization via chi-squared preference optimization. In *The Thirteenth International Conference on Learning Representations*.

Huang, A., Block, A., Liu, Q., Jiang, N., Foster, D. J., and Krishnamurthy, A. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment, 2025.

Ichihara, Y., Jinnai, Y., Morimura, T., Abe, K., Ariu, K., Sakamoto, M., and Uchibe, E. Evaluation of best-of-n sampling strategies for language model alignment. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Karwowski, J., Hayman, O., Bai, X., Kiendlhofer, K., Griffin, C., and Skalse, J. M. V. Goodhart's law in reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Krakovna, V., Uesato, J., Mikulik, V., Rahtz, M., Everitt, T., Kumar, R., Kenton, Z., Leike, J., and Legg, S. Specification gaming: the flip side of AI ingenuity. https://deepmind.google/discover/blog/specification-gaming-the-flip-side-of-ai-ingenuity/, April 2020.

Kwa, T., Thomas, D., and Garriga-Alonso, A. Catastrophic Goodhart: regularizing RLHF with KL divergence does not mitigate heavy-tailed reward misspecification. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Laidlaw, C., Singhal, S., and Dragan, A. Correlated proxies: A new definition and improved mitigation for reward hacking. In *The Thirteenth International Conference on Learning Representations*, 2025.

Lambert, N., Pyatkin, V., Morrison, J., Miranda, L., Lin, B. Y., Chandu, K., Dziri, N., Kumar, S., Zick, T., Choi, Y., et al. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024.

Liu, T., Xiong, W., Ren, J., Chen, L., Wu, J., Joshi, R., Gao, Y., Shen, J., Qin, Z., Yu, T., et al. RRM: Robust reward model training mitigates reward hacking. *arXiv preprint arXiv:2409.13156*, 2024.

Liu, Z., Lu, M., Zhang, S., Liu, B., Guo, H., Yang, Y., Blanchet, J., and Wang, Z. Provably mitigating overoptimization in rlhf: Your sft loss is implicitly an adversarial regularizer. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Mayrink Verdun, C., Oesterling, A., Lakkaraju, H., and Calmon, F. P. Soft best-of-n sampling for model alignment. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, 2025.

Miao, Y., Zhang, S., Ding, L., Bao, R., Zhang, L., and Tao, D. Inform: Mitigating reward hacking in rlhf via information-theoretic reward modeling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Miao, Y., Zhang, S., Ding, L., Zhang, Y., Zhang, L., and Tao, D. The energy loss phenomenon in rlhf: A new perspective on mitigating reward hacking. *arXiv preprint arXiv:2501.19358*, 2025.

Mroueh, Y. Information theoretic guarantees for policy alignment in large language models. *arXiv preprint arXiv:2406.05883*, 2024.

Mudgal, S., Lee, J., Ganapathy, H., Li, Y., Wang, T., Huang, Y., Chen, Z., Cheng, H.-T., Collins, M., Strohman, T., et al. Controlled decoding from language models. In *Forty-first International Conference on Machine Learning*, 2024.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

OpenAI. Detecting misbehavior in frontier reasoning models, March 2025. URL https://openai.com/index/chain-of-thought-monitoring/.

Pace, A., Mallinson, J., Malmi, E., Krause, S., and Severyn, A. West-of-n: Synthetic preference generation for improved reward modeling. *arXiv preprint arXiv:2401.12086*, 2024.

Pan, A., Bhatia, K., and Steinhardt, J. The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models, February 2022. URL http://arxiv.org/abs/2201.03544. arXiv:2201.03544 [cs].

Pan, J., He, H., Bowman, S. R., and Feng, S. Spontaneous reward hacking in iterative self-refinement. *arXiv preprint arXiv:2407.04549*, 2024.

Puri, I., Sudalairaj, S., Xu, G., Xu, K., and Srivastava, A. A probabilistic inference approach to inference-time scaling of llms using particle-based monte carlo methods. *arXiv preprint arXiv:2502.01618*, 2025.

Qiu, J., Lu, Y., Zeng, Y., Guo, J., Geng, J., Wang, H., Huang, K., Wu, Y., and Wang, M. Treebon: Enhancing inference-time alignment with speculative tree-search and best-of-n sampling. *arXiv preprint arXiv:2410.16033*, 2024.

Quarteroni, A., Sacco, R., and Saleri, F. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2006.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Rame, A., Vieillard, N., Hussenot, L., Dadashi-Tazehozi, R., Cideron, G., Bachem, O., and Ferret, J. WARM: On the Benefits of Weight Averaged Reward Models. In *International Conference on Machine Learning*, pp. 42048–42073. PMLR, 2024.

Rashidinejad, P. and Tian, Y. Sail into the Headwind: Alignment via Robust Rewards and Dynamic Labels against Reward Hacking. *arXiv preprint arXiv:2412.09544*, 2024.

Sessa, P. G., Dadashi, R., Hussenot, L., Ferret, J., Vieillard, N., Ramé, A., Shariari, B., Perrin, S., Friesen, A., Cideron, G., et al. Bond: Aligning llms with best-of-n distillation. *arXiv preprint arXiv:2407.14622*, 2024.

Shah, R., Varma, V., Kumar, R., Phuong, M., Krakovna, V., Uesato, J., and Kenton, Z. Goal misgeneralization: Why correct specifications aren't enough for correct goals. *arXiv preprint arXiv:2210.01790*, 2022.

Shen, L., Chen, S., Song, L., Jin, L., Peng, B., Mi, H., Khashabi, D., and Yu, D. The Trickle-down Impact of Reward (In-)consistency on RLHF, September 2023. URL http://arxiv.org/abs/2309.16155. arXiv:2309.16155 [cs].

Skalse, J., Howe, N., Krasheninnikov, D., and Krueger, D. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021, 2020.

Thaler, R. H. Anomalies: The winner's curse. *Journal of economic perspectives*, 2(1):191–202, 1988.

Thaler, R. H. *The winner's curse: Paradoxes and anomalies of economic life*. Simon and Schuster, 2012.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Wang, C., Zhao, Z., Jiang, Y., Chen, Z., Zhu, C., Chen, Y., Liu, J., Zhang, L., Fan, X., Ma, H., et al. Beyond reward hacking: Causal rewards for large language model alignment. *arXiv preprint arXiv:2501.09620*, 2025.

Welleck, S., Bertsch, A., Finlayson, M., Schoelkopf, H., Xie, A., Neubig, G., Kulikov, I., and Harchaoui, Z. From decoding to meta-generation: Inference-time algorithms for large language models. *Transactions on Machine Learning Research*, 2024.

Weng, L. Reward hacking in reinforcement learning. https://lilianweng.github.io/posts/2024-11-28-reward-hacking/, November 2024. Accessed: 2025-05-03.

Xu, Z., Vemuri, S., Panaganti, K., Kalathil, D., Jain, R., and Ramachandran, D. Distributionally Robust Direct Preference Optimization, February 2025. URL http://arxiv.org/abs/2502.01930. arXiv:2502.01930 [cs].

Yang, J. Q., Salamatian, S., Sun, Z., Suresh, A. T., and Beirami, A. Asymptotics of language model alignment. *arXiv preprint arXiv:2404.01730*, 2024a.

Yang, R., Ding, R., Lin, Y., Zhang, H., and Zhang, T. Regularizing Hidden States Enables Learning Generalizable Reward Model for LLMs, October

2024b. URL http://arxiv.org/abs/2406.10216. arXiv:2406.10216 [cs].

Yang, T., Mei, J., Dai, H., Wen, Z., Cen, S., Schuurmans, D., Chi, Y., and Dai, B. Faster WIND: Accelerating Iterative Best-of-N Distillation for LLM Alignment. In *The 28th International Conference on Artificial Intelligence and Statistics*.

Zeng, Z., Yu, J., Gao, T., Meng, Y., Goyal, T., and Chen, D. Evaluating large language models at evaluating instruction following. *arXiv preprint arXiv:2310.07641*, 2023.

Zhang, R., Haider, M., Yin, M., Qiu, J., Wang, M., Bartlett, P., and Zanette, A. Accelerating best-of-n via speculative rejection. In *2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ ICML 2024)*, 2024a.

Zhang, X., Ton, J.-F., Shen, W., Wang, H., and Liu, Y. Overcoming reward overoptimization via adversarial policy optimization with lightweight uncertainty estimation. *arXiv preprint arXiv:2403.05171*, 2024b.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623, 2023.

Zhu, B., Jordan, M. I., and Jiao, J. Iterative data smoothing: Mitigating reward overfitting and overoptimization in rlhf. *arXiv preprint arXiv:2401.16335*, 2024.

This appendix is organized as follows:

- **Section A** expands on the inevitability of reward hacking, stating Theorem 1, highlighting the key assumptions, and summarizing its main consequences.
- **Section B** contains the proof of Theorem 2, including derivations of the KL–reward tradeoff and discussion of the Best-of-Poisson example.
- **Section C** discusses additional details on Theorem 3, characterizing the hacking threshold and outlining computational approaches.
- **Section D** presents additional experimental details and results.

A. Inference-Time Reward Hacking

One can observe that initially intensifying optimization to a proxy objective may improve intended performance. However, beyond a certain point – which we call the *hacking threshold* – further proxy-optimization actually degrades the true reward. Theorem 1 in Section 2 formalizes this phenomenon: under very general conditions on a one-parameter family of proxy distributions π_θ, the map

$$\theta \;\mapsto\; \mathbb{E}_{X\sim\pi_\theta}[r_{\mathrm{t}}(X)] := f(\theta) \tag{7}$$

can have at most one interior extremum. Thus, there is either a monotonic benefit (or disadvantage) to strengthening the proxy or exactly one "sweet spot" before reward hacking sets in.

Let $\psi(x,\theta)$ denote the score function of distribution $\pi_\theta(x)$ with density $p_\theta(x)$. Standard calculations under mild regularity conditions gives us the derivative of the true reward under π_θ:

$$f'(\theta) = \int r_t(x)\nabla_\theta p_\theta(x)\mathrm{d}x = \int r_t(x)p_\theta(x)\nabla_\theta \log p_\theta(x)\mathrm{d}x = \mathbb{E}_{X\sim\pi_\theta}[r_t(X)\psi(X,\theta)] \tag{8}$$

The rest of this appendix proceeds as follows. We first restate Theorem 1 in more detail (see Theorem 4). We prove the single-crossing property for the derivative of the true reward expectation by invoking variation-diminishing kernels. We then derive Corollary 2 and 3 and Lemma 1. We specialize the discussion to two concrete examples with Best-of-n and Best-of-Poisson.

Theorem 4 (Inevitability of Reward Hacking)**.** Let $\{\pi_\theta\}_{\theta\in\Theta\subset\mathbb{R}}$ be a family of probability measures on a measurable space $(\mathcal{X},\mathcal{A})$ with density $p_\theta(x)$ relative to a common dominating measure. Assume that

1. p_θ is *strictly TP*$_2$ (totally positive with order 2): for every $\theta_1 < \theta_2$ and $x_1 < x_2$,
$$p_{\theta_1}(x_1)p_{\theta_2}(x_2) \;>\; p_{\theta_1}(x_2)p_{\theta_2}(x_1).$$
2. The score function $\psi(x,\theta) := \partial_\theta \log p_\theta(x)$ exists, is continuous in x, and is *strictly increasing in* x for each fixed θ.
3. Denote $f(\theta) \;:=\; \mathbb{E}_{\pi_\theta}[r_t(X)]$ where $r_t : \mathcal{X} \to [0,\infty)$ is non-negative, bounded, and not identically zero.

Then the derivative $f'(\theta) \;=\; \mathbb{E}_{\pi_\theta}[r_t(X)\,\psi(X,\theta)]$ changes sign at most once on Θ. Consequently f is monotone or possesses exactly one interior extremum.

The main idea to establish this result is that when we use parameter θ to control inference-time methods, we create a family of densities $\{p_\theta(x)\}$ that act as positive kernels. Crucially, these kernels satisfy the strict total positivity conditions required for variation-diminishing theorems to apply. The key insight is that the expression $f'(\theta) = \mathbb{E}_\theta[r_t(X)\psi(X,\theta)]$ captures how the **rate of change** of expected reward depends on the interaction between:

- $r_t(X)$: The true reward function (which may have complex variation patterns). Our only assumption on the true reward is that it is bounded. We then translate the reward function so that it is non-negative. The boundedness assumption is a natural one in the alignment setting because real-world rewards originate from human judgments given on finite scales (e.g. star ratings, Likert scores, or normalized preference probabilities). Moreover, clipping or normalizing the reward prevents unbounded returns, improving the stability of policy updates and inference-time mechanisms.

- $\psi(X, \theta) = \partial_\theta \log p_\theta(X)$: The score function (which encodes how the distribution changes with the parameter)

The variation-diminishing theorem tells us that this interaction can have at most one sign change, which directly implies that $f'(\theta)$ can cross zero at most once, i.e., $f(\theta)$ is either monotonic or has exactly one extremum and this, in turn implies that *reward hacking follows predictable patterns*. This theoretical finding then explains the characteristic behavior of reward hacking as shown in empirical findings.

We now present the proof for Theorem 4.

Proof. Fix θ and set $h_\theta(x) := r_t(x)\psi(x, \theta)$. Because $\psi(\cdot, \theta)$ is strictly increasing, it has at most one zero; since $r_t \geq 0$, h_θ has the same single $(- \to +)$ sign change in x. Strict TP_2 of p_θ and Karlin's variation–diminishing theorem imply that $\theta \mapsto F(\theta) := \int h_\theta(x)p_\theta(x)\,dx = f'(\theta)$ inherits *at most* the same number of sign changes, namely one. □

Having established the inevitability result, we next explore how it specializes in classical families via a simple corollary.

Corollary 2 (Strict MLR densities)**.** Let $p_\theta(x)$ be *strictly monotone–likelihood–ratio* in x (i.e. $p_{\theta_2}(x)/p_{\theta_1}(x)$ strictly increases in x whenever $\theta_2 > \theta_1$). If $\psi(x, \theta) = \partial_\theta \log p_\theta(x)$ is strictly increasing in x, then all conclusions of Theorem 4 apply.

Example 1 (One-parameter exponential families)**.** Any regular canonical exponential family

$$p_\theta(x) = \exp\{\eta(\theta)T(x) - A(\theta) + B(x)\},$$

with strictly monotone statistic T and strictly monotone natural parameter η is strict MLR and satisfies $\psi(x, \theta) = \eta'(\theta)T(x) - A'(\theta)$, which is strictly increasing in x. Hence the single–crossing property holds for every bounded non-negative $r_t \geq 0$.

The conditions are satisfied by two inference-time methods we study:

Best-of-n: The distribution corresponds to the maximum of n i.i.d. samples from the reference distribution. When proxy rewards are uniformly distributed, this yields $p_n(u) = nu^{n-1}$ for $u \in [0, 1]$. The likelihood ratio $\frac{p_{n_2}(u)}{p_{n_1}(u)} = \frac{n_2}{n_1}u^{n_2-n_1}$ is strictly increasing in u when $n_2 > n_1$, establishing strict MLR (which implies strict TP2). The score function $\psi(u, n) = \frac{1}{n} + \log u$ is strictly increasing in u.

Best-of-Poisson: The distribution $p_\mu(u) = (\mu u + 1)e^{\mu(u-1)}$ for $u \in [0, 1]$ can be verified to satisfy strict MLR by direct computation of likelihood ratios. The score function $\psi(u, \mu) = u - 1 + \frac{u}{\mu u+1}$ is strictly increasing in u.

Under the conditions presented in Theorem 4, we know that at most one interior extrema exists. The following corollary gives precise conditions on when such an interior extrema exists.

Lemma 1. Under the assumptions of Theorem 1 and with $r_t \in C^1$ on a neighborhood of its boundaries (0 and 1) and let $\Theta = [\theta_l, \theta_r]$,

$$\lim_{\theta \downarrow \theta_\ell} f'(\theta) = r_t'(0+)\,\mathbb{E}_{\theta_\ell}[X\,\psi(X, \theta_\ell)], \qquad \lim_{\theta \uparrow \theta_r} f'(\theta) = -\,r_t'(1-)\,\mathbb{E}_{\theta_r}[(1 - X)\psi(X, \theta_r)].$$

Then, a stationary point exists *iff*

$$\lim_{\theta \downarrow \theta_\ell} f'(\theta) \;\text{ and }\; \lim_{\theta \uparrow \theta_r} f'(\theta)$$

are of opposite sign (or one limit is 0 while the other is non-zero). By continuity, the Intermediate-Value Theorem then forces one root of f' and single–crossing rules out a second. For example. for BoN, $\psi(x, n) = 1/n + \log x$ with $\mathbb{E}_n[X\,\psi] = 1/(n+1)^2$, $\mathbb{E}_n[(1-X)\psi] = -1/(n+1)^2$. Hence a stationary point exists iff $r_t'(0+)$ and $r_t'(1-)$ have opposite signs, and its location $n_\star$ solves $\mathbb{E}_n[r_t(X)\psi(X, n)] = 0$.

Proof. We prove the left boundary results (the right one is identical with $x \mapsto 1 - x$.) Write the first-order expansion $r_t(x) = r_t(0+) + r_t'(0+)x + R(x)$ with $R(x) = o(x)$ as $x \to 0$. Because $\mathbb{E}_\theta[\psi] = 0$,

$$f'(\theta) = r_t'(0+)\mathbb{E}_\theta[X\psi] + \mathbb{E}_\theta[R(X)\psi].$$

Strict increase of ψ implies $|X\psi| \leq C(1+X)$ on $[0,1] \times [\theta_\ell, \theta_\ell + \rho]$, so $\mathbb{E}_\theta[X\psi] \to \mathbb{E}_{\theta_\ell}[X\psi]$ by dominated convergence. Next, fix $\delta \in (0,1)$ and split the expectation:

$$\mathbb{E}_\theta[r_t(X)\psi] = \mathbb{E}_\theta[r_t(X)\psi \mathbf{1}_{\{X \leq \delta\}}] + \mathbb{E}_\theta[r_t(X)\psi \mathbf{1}_{\{X > \delta\}}].$$

Near 0, we have $|R(x)| \leq c_\delta x$, hence the term is bounded by $c_\delta \mathbb{E}_\theta[X|\psi|]$; choose δ so small that $c_\delta \mathbb{E}_{\theta_\ell}[X|\psi|] < \varepsilon$ and continuity keeps it $< 2\varepsilon$ for θ close enough to θ_ℓ. Away from 0, we use boundedness of r_t and local boundedness of ψ to obtain a factor $P_\theta\{X > \delta\} \to 0$.

Combining the two parts gives $\mathbb{E}_\theta[R(X)\psi] \to 0$, yielding the claimed limit. □

Corollary 3 (Reward behavior for MLR samplers)**.** Assume Lemma 1 holds and that the family $\{p_\theta\}_{\theta \in \Theta}$ is *monotone–likelihood–ratio* in $x \in (0,1)$. Then for every $\theta \in \Theta$

$$L_\theta := \mathbb{E}_\theta[X\,\psi(X,\theta)] > 0, \qquad R_\theta := \mathbb{E}_\theta[(1-X)\psi(X,\theta)] < 0,$$

so

$$\operatorname{sign} f'(\theta_\ell^+) = \operatorname{sign} r_t'(0+), \qquad \operatorname{sign} f'(\theta_r^-) = \operatorname{sign} r_t'(1-)$$

Single-crossing of f' implies that $f(\theta)$ can assume *exactly* one of the four shapes:

regime	$r_t'(0+)$	$r_t'(1-)$
monotonic improvement	≥ 0	≥ 0
reward hacking	> 0	< 0
reward grokking	< 0	> 0
immediate decline	≤ 0	≤ 0

Proof. Let p_θ be differentiable in θ with score $\psi(x,\theta) = \partial_\theta \log p_\theta(x)$. For any integrable g we have shown that:

$$\frac{d}{d\theta}\mathbb{E}_\theta[g(X)] = \mathbb{E}_\theta[g(X)\psi(X,\theta)]$$

We first use that the MLR property implies first-order stochastic dominance. For every *increasing* function g, we have that $\theta \mapsto \mathbb{E}_\theta[g(X)]$ is non-decreasing and its derivative is ≥ 0. Choosing $g(x) = x$ gives

$$L_\theta = \mathbb{E}_\theta[X\,\psi(X,\theta)] = \frac{d}{d\theta}\mathbb{E}_\theta[X] \;>\; 0$$

On the other hand, $g(x) = 1 - x$ (strictly decreasing) yields:

$$R_\theta = \mathbb{E}_\theta[(1-X)\psi(X,\theta)] = \frac{d}{d\theta}\mathbb{E}_\theta[1-X] \;<\; 0$$

Thus $L_\theta > 0$ and $R_\theta < 0$ for every $\theta \in \Theta$. □

We have shown the conditions under which the expected value of true reward has a critical point with respect to θ. We now show the conditions under which our results extend identically if we are studying the expected value of true reward as a function of the KL divergence with respect to the reference distribution π_{θ_0}.

Lemma 2 (Score–covariance formula for $D'(\theta\|\theta_0)$)**.** Let $\{\pi_\theta\}_{\theta \in \Theta}$ be a regular parametric family with density $p_\theta(x)$ such that

- p_θ and $\partial_\theta p_\theta$ are jointly measurable and $\partial_\theta p_\theta(x)$ is locally integrable in θ;
- the *score* $\psi(x,\theta) := \partial_\theta \log p_\theta(x)$ is square–integrable: $\mathbb{E}_\theta[\psi^2] < \infty$.

For a fixed reference point $\theta_0 \in \Theta$ define the Kullback–Leibler divergence

$$D(\theta\|\theta_0) \;:=\; \int p_\theta(x)\,\log\frac{p_\theta(x)}{p_{\theta_0}(x)}\,d\mu(x).$$

Then $D(\theta\|\theta_0)$ is differentiable and

$$\frac{d}{d\theta}\,D(\theta\|\theta_0) \;=\; \mathbb{E}_\theta\Big[(\log p_\theta(x) - \log p_{\theta_0}(x))\,\psi(X,\theta)\Big].$$

In particular, for canonical exponential families with strictly increasing natural parameter, this simplifies to

$$\frac{d}{d\theta}\,D(\theta\|\theta_0) = (\eta(\theta) - \eta(\theta_0))\,A''(\theta),$$

which is strictly positive when $\theta > \theta_0$ (and negative for $\theta < \theta_0$).

Proof. Write $g_\theta(x) := \log p_\theta(x) - \log p_{\theta_0}(x)$. Then $D(\theta\|\theta_0) = \mathbb{E}_\theta[g_\theta(X)]$. For a *parameter-dependent* integrand the classical Fisher–Leibniz rule gives

$$\frac{d}{d\theta}\mathbb{E}_\theta[g_\theta(X)] = \mathbb{E}_\theta[\partial_\theta g_\theta(X)] + \mathbb{E}_\theta[g_\theta(X)\,\psi(X,\theta)], \tag{†}$$

whenever $\partial_\theta g_\theta$ exists and an L^1 dominated–convergence bound holds (true here by the square–integrable score assumption). Since $\partial_\theta g_\theta(x) = \psi(x,\theta)$ and $\mathbb{E}_\theta[\psi] = 0$, the first term vanishes, leaving exactly

$$\frac{d}{d\theta}D(\theta\|\theta_0) = \mathbb{E}_\theta[g_\theta(X)\,\psi(X,\theta)]$$

□

B. Best-of-Poisson

In this section, we prove Theorem 2 and establish, as a consequence, that Best-of-Poisson is numerically near-optimal as compared to tilted distribution π^*_λ in terms of KL divergence. As a consequence, hedging in the optimal tilted distribution is almost equivalent to hedging with BoP, making BoP an effective inference-time stand-in for the optimal tilted distribution which can be difficult to numerically estimate. We start by establishing the BoP distribution in the uniform case.

Theorem 5 (BoP Distribution)**.** Let $\mu > 0$ be the parameter of the Best-of-Poisson sampling method and let X_μ be the random variable representing the response selected by BoP. The probability density function $q_\mu(x)$ of X_μ is given by:

$$q_\mu(x) = (1 + \mu x)e^{\mu(x-1)}, \tag{9}$$

for $x \in [0,1]$, where $n = n' + 1$ with $n' \sim \text{Poisson}(\mu)$.

Proof. Write $X_\mu = \max\{U_0, U_1, \ldots, U_{n'}\}$ where $n' \sim \text{Poisson}(\mu)$ and $U_i \overset{\text{iid}}{\sim} \text{Unif}[0,1]$. Consider $U_0 \sim \text{Unif}[0,1]$ to be the mandatory draw to achieve a sample size of at least one.

For $x \in [0,1]$,

$$F_\mu(x) := \Pr(X_\mu \le x) = \Pr(U_0 \le x)\,\Pr(U_i \le x \text{ for } 1 \le i \le n') = x\,\mathbb{E}[x^{n'}] = x\,e^{-\mu(1-x)},$$

because $\mathbb{E}[x^{n'}] = \exp\{-\mu(1-x)\}$ is the moment-generating function of a Poisson variable evaluated at $\log x$.

Now, differentiating F_μ on $(0,1)$ gives

$$q_\mu(x) = e^{-\mu(1-x)} + \mu x\,e^{-\mu(1-x)} = (1+\mu x)\,e^{\mu(x-1)},$$

which extends continuously to the endpoints. A direct computation verifies $\int_0^1 q_\mu(x)\,dx = 1$, so q_μ is a valid density. □

Now, we can prove Theorem 2.

Theorem 6 (KL Divergence of BoP)**.** Let X_μ be the random variable representing the response selected by BoP with parameter μ. Then:

$$D_{\mathsf{KL}}(\pi_{\text{Poisson}} \| \pi_{\text{ref}}) = \frac{e^{-\mu-1}(\text{Ei}(\mu+1) - \text{Ei}(1))}{\mu} + \log(\mu+1) - 1, \tag{10}$$

$$\mathbb{E}[X_\mu] = 1 - \frac{1}{\mu} + \frac{1-e^{-\mu}}{\mu^2}, \tag{11}$$

where $\text{Ei}(z) = -\int_{-z}^{\infty} \frac{e^{-t}}{t} dt$ is the exponential integral function.

Proof. **Mean.**

$$\mathbb{E}[X_\mu] = \int_0^1 x(1+\mu x)e^{\mu(x-1)}dx = \int_0^1 (x+\mu x^2)e^{\mu(x-1)}dx$$

With the substitution $u = \mu(x-1)$, we get that

$$\int_0^1 xe^{\mu(x-1)}dx = \frac{1}{\mu^2}\int_{-\mu}^0 (u+\mu)e^u du = \frac{\mu - 1 + e^{-\mu}}{\mu^2}$$

$$\int_0^1 x^2 e^{\mu(x-1)}dx = \frac{1}{\mu^3}\int_{-\mu}^0 (u+\mu)^2 e^u du = \frac{\mu^2 - 2\mu + 2 - 2e^{-\mu}}{\mu^3}$$

Hence

$$\mathbb{E}[X_\mu] = \frac{\mu - 1 + e^{-\mu}}{\mu^2} + \mu\frac{\mu^2 - 2\mu + 2 - 2e^{-\mu}}{\mu^3} = 1 - \frac{1}{\mu} + \frac{1-e^{-\mu}}{\mu^2}$$

KL divergence. Because $\log q_\mu(x) = \log(1+\mu x) + \mu(x-1)$,

$$D_{\mathsf{KL}}(q_\mu \| U) = \int_0^1 q_\mu(x)\log(1+\mu x)\,dx + \mu[\mathbb{E}[X_\mu] - 1]$$

To compute the integral, set $t = 1 + \mu x$:

$$\int_0^1 q_\mu(x)\log(1+\mu x)\,dx = \frac{e^{-\mu-1}}{\mu}\int_1^{\mu+1} t\,e^t \log t\,dt$$

Integration by parts ($f = \log t,\ dg = te^t dt$) yields

$$\int te^t \log t\,dt = \frac{1}{2}t^2 e^t\Big(\log t - \frac{1}{2}\Big) - \frac{1}{2}\text{Ei}(t) + C$$

hence

$$\int_0^1 q_\mu(x)\log(1+\mu x)\,dx = \log(\mu+1) - \frac{1-e^{-\mu}}{\mu} + \frac{e^{-\mu-1}}{\mu}[\text{Ei}(\mu+1) - \text{Ei}(1)]$$

The resulting term becomes:

$$D_{\mathsf{KL}}(q_\mu \| \text{Unif}) = \frac{e^{-\mu-1}}{\mu}[\text{Ei}(\mu+1) - \text{Ei}(1)] + \log(\mu+1) - 1$$ □

Now that we have derived the exact formulas for BoP's KL divergence and expected reward, we can establish that BoP provides a practical approximation to the optimal tilted distribution with negligible performance loss.

Theorem 7 (Near-Optimality of BoP)**.** Let π_{Poisson} be the distribution induced by Best-of-Poisson with parameter $\mu > 0$, and let π_λ^* be the optimal KL-constrained tilted distribution with parameter $\lambda > 0$, defined as:

$$\pi_\lambda^*(x) = \frac{\pi_{\text{ref}}(x)e^{\lambda r_p(x)}}{Z(\lambda)}, \tag{12}$$

where $Z(\lambda)$ is the normalization constant. For any given expected reward level, there exists a μ for BoP and a λ for the tilted distribution such that $\mathbb{E}_{X\sim\pi_{\text{Poisson}}}[r_p(X)] = \mathbb{E}_{X\sim\pi_\lambda^*}[r_p(X)]$, and the KL divergence gap between these distributions is given by:

$$D_{\mathsf{KL}}(\pi_{\text{Poisson}}\|\pi_{\text{ref}}) - D_{\mathsf{KL}}(\pi_\lambda^*\|\pi_{\text{ref}}) = \frac{e^{-(\mu+1)}(\text{Ei}(\mu+1) - \text{Ei}(1))}{\mu} + \ln\left(\frac{(\mu+1)(e^\lambda - 1)}{\lambda}\right) - \frac{\lambda e^\lambda}{e^\lambda - 1} \tag{13}$$

In particular, numerical evaluation shows that for all μ, if $\lambda > 0$ is chosen so that $\mathbb{E}_{X\sim\pi_{\text{Poisson}}}[r_p(X)] = \mathbb{E}_{X\sim\pi_\lambda^*}[r_p(X)]$, then the KL-gap satisfies

$$0 \le D_{\mathsf{KL}}(\pi_{\text{Poisson}}\|\pi_{\text{ref}}) - D_{\mathsf{KL}}(\pi_\lambda^*\|\pi_{\text{ref}}) \;\le\; 7 \times 10^{-3}.$$

That is, Best-of-Poisson achieves nearly the optimal trade-off between expected reward and KL divergence from the reference distribution.

Proof. Theorem 2 above established that

$$D_{\mathsf{KL}}(\pi_{\text{Poisson}}\|\pi_{\text{ref}}) = \frac{e^{-\mu-1}(\text{Ei}(\mu+1) - \text{Ei}(1))}{\mu} + \log(\mu+1) - 1 \tag{14}$$

Now, we need to derive the KL divergence between the optimal tilted distribution π_λ^* and the reference distribution π_{ref}. Recall that the optimal tilted distribution with parameter λ is given by:

$$\pi_\lambda^*(x) = \frac{\pi_{\text{ref}}(x)e^{\lambda r_p(x)}}{Z(\lambda)} \tag{15}$$

where $Z(\lambda) = \int_{\mathcal{X}} \pi_{\text{ref}}(x)e^{\lambda r_p(x)}dx$ is the normalization constant.

The KL divergence is defined as:

$$D_{D_{\mathsf{KL}}}(\pi_\lambda^*\|\pi_{\text{ref}}) = \int \pi_\lambda^*(x) \log \frac{\pi_\lambda^*(x)}{\pi_{\text{ref}}(x)} dx \tag{16}$$

Substituting $\pi_\lambda^*(x) = \frac{\pi_{\text{ref}}(x)e^{\lambda r_p(x)}}{Z(\lambda)}$ in the KL divergence gives:

$$\begin{aligned} D_{\mathsf{KL}}(\pi_\lambda^*\|\pi_{\text{ref}}) &= \int \frac{\pi_{\text{ref}}(x)e^{\lambda r_p(x)}}{Z(\lambda)} \log \frac{e^{\lambda r_p(x)}}{Z(\lambda)} dx && (17)\\ &= \int \frac{\pi_{\text{ref}}(x)e^{\lambda r_p(x)}}{Z(\lambda)} \left[\lambda r_p(x) - \log Z(\lambda)\right] dx && (18)\\ &= \lambda \int \frac{\pi_{\text{ref}}(x)e^{\lambda r_p(x)}}{Z(\lambda)} r_p(x)dx - \log Z(\lambda) \int \frac{\pi_{\text{ref}}(x)e^{\lambda r_p(x)}}{Z(\lambda)} dx && (19) \end{aligned}$$

The second integral equals 1 since π_λ^* is a probability distribution. For the first integral, we recognize it as the expected value of $r_p(X)$ under the distribution π_λ^*:

$$\mathbb{E}_{\pi_\lambda^*}[r_p(X)] = \int r_p(x)\pi_\lambda^*(x)dx = \int r_p(x) \frac{\pi_{\text{ref}}(x)e^{\lambda r_p(x)}}{Z(\lambda)} dx \tag{20}$$

Therefore:

$$D_{\mathsf{KL}}(\pi_\lambda^*\|\pi_{\text{ref}}) = \lambda \mathbb{E}_{\pi_\lambda^*}[r_p(X)] - \log Z(\lambda) \tag{21}$$

This shows that the KL divergence is given by the expected reward minus the log partition function. We can further simplify by noting that the derivative of $\log Z(\lambda)$ with respect to λ gives us the expected reward:

$$\frac{d}{d\lambda}\log Z(\lambda) = \frac{1}{Z(\lambda)}\frac{d}{d\lambda}\int \pi_{\text{ref}}(x)e^{\lambda r_p(x)}dx = \frac{1}{Z(\lambda)}\int \pi_{\text{ref}}(x)e^{\lambda r_p(x)}r_p(x)dx = \mathbb{E}_{\pi^*_\lambda}[r_p(X)] \tag{22}$$

Therefore:

$$D_{\mathsf{KL}}(\pi^*_\lambda \| \pi_{\text{ref}}) = \lambda\frac{d}{d\lambda}\log Z(\lambda) - \log Z(\lambda) \tag{23}$$

Which is a standard result for exponential families namely, that the KL divergence equals the Bregman divergence of the log partition function. For our specific case where π_{ref} is the uniform distribution on $[0, 1]$ and $r_p(x) = x$ (after the probability integral transform), we have:

$$Z(\lambda) = \int_0^1 e^{\lambda x}dx = \frac{e^\lambda - 1}{\lambda} \log Z(\lambda) = \log\left(\frac{e^\lambda - 1}{\lambda}\right) \tag{24}$$

The derivative is:

$$\frac{d}{d\lambda}\log Z(\lambda) = \frac{e^\lambda}{e^\lambda - 1} - \frac{1}{\lambda} \tag{25}$$

So:

$$D_{\mathsf{KL}}(\pi^*_\lambda \| \pi_{\text{ref}}) = \lambda\left(\frac{e^\lambda}{e^\lambda - 1} - \frac{1}{\lambda}\right) - \log\left(\frac{e^\lambda - 1}{\lambda}\right) \tag{26}$$

$$= \lambda\frac{e^\lambda}{e^\lambda - 1} - 1 - \log\left(\frac{e^\lambda - 1}{\lambda}\right) \tag{27}$$

After further algebraic manipulation:

$$D_{\mathsf{KL}}(\pi^*_\lambda \| \pi_{\text{ref}}) = \lambda - 1 + \frac{\lambda}{e^\lambda - 1} - \log\left(\frac{e^\lambda - 1}{\lambda}\right) \tag{28}$$

This shows that the difference in KL is given by

$$D_{\mathsf{KL}}(\pi_{\text{Poisson}} \| \pi_{\text{ref}}) - D_{\mathsf{KL}}(\pi^*_\lambda \| \pi_{\text{ref}}) \tag{29}$$

$$= \frac{e^{-\mu-1}(\text{Ei}(\mu + 1) - \text{Ei}(1))}{\mu} + \log(\mu + 1) - 1 - \left(\lambda - 1 + \log\left(\frac{\lambda}{e^\lambda - 1}\right) + \frac{\lambda}{e^\lambda - 1}\right) \tag{30}$$

To numerically verify the near-optimality claim, we solve the equation $\mathbb{E}_{X\sim\pi_{\text{BoP}}}[r_p(X)] = \mathbb{E}_{X\sim\pi^*_\lambda}[r_p(X)]$ using Newton's method to find λ as a function of μ. For each value of μ, we then evaluate the KL divergence gap:

$$D_{\mathsf{KL}}(\pi_{\text{BoP}} \| \pi_{\text{ref}}) - D_{\mathsf{KL}}(\pi^*_\lambda \| \pi_{\text{ref}}) = \frac{e^{-(\mu+1)}(\text{Ei}(\mu + 1) - \text{Ei}(1))}{\mu} + \ln\left(\frac{(\mu + 1)(e^\lambda - 1)}{\lambda}\right) - \frac{\lambda e^\lambda}{e^\lambda - 1} \tag{31}$$

□

Numerical evaluation confirms that this difference is bounded by approximately 7×10^{-3} across the entire range of μ as in Figure 2. This validates that Best-of-Poisson indeed provides a practically equivalent approximation to the optimal tilted distribution with negligible computational overhead.

C. Proof of Theorem 3

In Section 4, we present an efficient way to find the optimal hacking threshold. We restate Theorem 3 and prove it.

Theorem 8 (Hacking Threshold Characterization)**.** Let r_t be a true reward oracle and $\theta^\dagger$ be the hacking threshold from Definition 1. For each inference-time method, $\theta^\dagger$ is characterized by the following conditions:

$$\text{For BoN, } n^\dagger \text{ satisfies:} \quad \nabla_\alpha \mathbb{E}_{u\sim\text{Beta}(\alpha,1)}[r_t(u)] = \int_0^1 r_t(u)\,(1+n^\dagger \log u)\,u^{n^\dagger-1}\,du = 0, \tag{32}$$

$$\text{For SBoN, } \lambda^\dagger \text{ satisfies:} \quad \nabla_\lambda \mathbb{E}_{u\sim f_\lambda}[r_t(u)] = \mathbb{E}_{U_1,\ldots,U_n}[\text{Cov}(r_t(V), V|U_1,\ldots,U_n)]|_{\lambda=\lambda^\dagger} = 0, \tag{33}$$

$$\text{For BoP, } \mu^\dagger \text{ satisfies:} \quad \nabla_\mu \mathbb{E}_{u\sim f_\mu}[r_t(u)] = \mathbb{E}_{u\sim f_{\mu^\dagger}}\left[r_t(u)\left(u-1+\frac{u}{\mu^\dagger u+1}\right)\right] = 0. \tag{34}$$

Proof. We now consider each mechanism separately:

Hedging in Best-of-n. In BoN, we approximate the integer n via a continuous parameter α by placing a Beta$(\alpha,1)$ prior on u. Its density is $f_\alpha(u)=\alpha u^{\alpha-1}$, so $\psi(u,\alpha)=\partial_\alpha[\ln\alpha+(\alpha-1)\ln u]=\dfrac{1}{\alpha}+\ln u$. Thus, the optimality condition becomes

$$\mathbb{E}_{u\sim\text{Beta}(\alpha,1)}\left[r_t(u)(\frac{1}{\alpha}+\ln u)\right]=0 \iff \int_0^1 r_t(u)\,(1+n\log u)\,u^{n-1}\,du=0, \tag{35}$$

which one solves for α to pick an effective sample size. In practice, this equation must be solved numerically. We do this by discretizing $[0,1]$ into M points and forming the Riemann-sum residual

$$R(\alpha)=\sum_{i=1}^{M} r_t(u_i)\left(\frac{1}{\alpha}+\ln u_i\right)u_i^{\alpha-1}\Delta u,$$

The root $R(\alpha)=0$ is equivalent to the hedging condition. Then, one applies any root-finding method, see, e.g., (Quarteroni et al., 2006), to locate the unique solution $\alpha^\dagger$. Finally, the discrete sample size is chosen as $N=[\alpha]$, the integer nearest $\alpha^\dagger$.

Hedging in Soft Best-of-n. Unlike BoN and BoP, SBoN does not admit a simple closed-form density due to its sampling mechanism. In SBoN, we first sample n responses $X_1,\ldots,X_n$ from the reference distribution, then select response X_i with probability $\frac{e^{\lambda r(X_i)}}{\sum_{j=1}^n e^{\lambda r(X_j)}}$. The resulting distribution is:

$$\pi_{n,\lambda}(x)=\mathbb{E}_{X_1,\ldots,X_{n-1}\sim\pi_{\text{ref}}}\left[\pi_{\text{ref}}(x)\cdot\frac{e^{\lambda r(x)}}{\frac{1}{n}\left(e^{\lambda r(x)}+\sum_{i=1}^{n-1}e^{\lambda r(X_i)}\right)}\right]$$

We can now derive the hedging condition $\frac{\partial}{\partial\lambda}\mathbb{E}_{u\sim f_\lambda}[r_t(u)]=0$. For the case of proxy reward percentiles, remember that the SBoN sampling mechanism works as follows

1. Sample $U_1,\ldots,U_n\sim$ Uniform$[0,1]$ independently
2. Select index Z with probability $P(Z=i|U_1,\ldots,U_n)=\frac{e^{\lambda U_i}}{\sum_{j=1}^n e^{\lambda U_j}}$
3. Return $V=U_Z$

Then, to start, we compute $\frac{\partial p_i}{\partial\lambda}$:

$$\frac{\partial p_i}{\partial\lambda}=\frac{\partial}{\partial\lambda}\left[\frac{e^{\lambda U_i}}{S}\right]=\frac{U_i e^{\lambda U_i}\cdot S-e^{\lambda U_i}\cdot\frac{\partial S}{\partial\lambda}}{S^2} \tag{36}$$

Since $\frac{\partial S}{\partial\lambda}=\sum_{j=1}^n U_j e^{\lambda U_j}$:

$$\frac{\partial p_i}{\partial\lambda}=\frac{e^{\lambda U_i}}{S}\left[U_i-\frac{\sum_{j=1}^n U_j e^{\lambda U_j}}{S}\right]=p_i\left[U_i-\sum_{j=1}^n U_j p_j\right] \tag{37}$$

Now, we compute the derivative of the expected reward:

$$\frac{\partial}{\partial\lambda}\mathbb{E}_{u\sim f_\lambda}[r_t(u)] = \mathbb{E}_{U_1,\ldots,U_n}\left[\frac{\partial}{\partial\lambda}\sum_{i=1}^{n} r_t(U_i)p_i\right] \tag{38}$$

$$= \mathbb{E}_{U_1,\ldots,U_n}\left[\sum_{i=1}^{n} r_t(U_i)\frac{\partial p_i}{\partial\lambda}\right] \tag{39}$$

$$= \mathbb{E}_{U_1,\ldots,U_n}\left[\sum_{i=1}^{n} r_t(U_i)p_i\left(U_i - \sum_{j=1}^{n} U_j p_j\right)\right] \tag{40}$$

Expanding this expression:

$$= \mathbb{E}_{U_1,\ldots,U_n}\left[\sum_{i=1}^{n} r_t(U_i)p_i U_i - \sum_{i=1}^{n} r_t(U_i)p_i \sum_{j=1}^{n} U_j p_j\right] \tag{41}$$

$$= \mathbb{E}_{U_1,\ldots,U_n}\left[\sum_{i=1}^{n} r_t(U_i)p_i U_i - \left(\sum_{i=1}^{n} r_t(U_i)p_i\right)\left(\sum_{j=1}^{n} U_j p_j\right)\right] \tag{42}$$

The expression inside the expectation is exactly the conditional covariance

$$\text{Cov}(r_t(V), V|U_1,\ldots,U_n) = \mathbb{E}[r_t(V)\cdot V|U_1,\ldots,U_n] - \mathbb{E}[r_t(V)|U_1,\ldots,U_n]\cdot\mathbb{E}[V|U_1,\ldots,U_n], \tag{43}$$

where:

$$\mathbb{E}[r_t(V)|U_1,\ldots,U_n] = \sum_{i=1}^{n} r_t(U_i)p_i \tag{44}$$

$$\mathbb{E}[V|U_1,\ldots,U_n] = \sum_{i=1}^{n} U_i p_i \tag{45}$$

$$\mathbb{E}[r_t(V)\cdot V|U_1,\ldots,U_n] = \sum_{i=1}^{n} r_t(U_i)U_i p_i \tag{46}$$

Therefore:

$$\frac{\partial}{\partial\lambda}\mathbb{E}_{u\sim f_\lambda}[r_t(u)] = \mathbb{E}_{U_1,\ldots,U_n}[\text{Cov}(r_t(V), V|U_1,\ldots,U_n)] \tag{47}$$

This condition must be evaluated numerically using the following procedure:

1. For a given λ, generate M independent realizations of $(U_1^{(m)},\ldots,U_n^{(m)}) \sim \text{Uniform}[0,1]^n$ for $m = 1,\ldots,M$.
2. For each realization m, compute the conditional covariance:

$$C^{(m)} = \text{Cov}(r_t(V), V|U_1^{(m)},\ldots,U_n^{(m)}) = \sum_{i=1}^{n} r_t(U_i^{(m)})U_i^{(m)}p_i^{(m)} - \left(\sum_{i=1}^{n} r_t(U_i^{(m)})p_i^{(m)}\right)\left(\sum_{j=1}^{n} U_j^{(m)}p_j^{(m)}\right) \tag{48}$$

   where $p_i^{(m)} = \frac{e^{\lambda U_i^{(m)}}}{\sum_{j=1}^{n} e^{\lambda U_j^{(m)}}}$.

3. Estimate the derivative as $R(\lambda) = \frac{1}{M}\sum_{m=1}^{M} C^{(m)}$.

4. Use root-finding methods (e.g., bisection or Newton's method) to locate $\lambda^\dagger$ where $R(\lambda^\dagger) = 0$.

Hedging in Best-of-Poisson. Here, one draws a Poisson(μ) number of samples (plus one) and selects the proxy-maximal u. For uniform u, the density is $p_\mu(u) = (\mu u + 1)e^{-\mu(1-u)}$, giving $\psi(u, \mu) = \partial_\mu \ln p_\lambda(u) = u - 1 + \dfrac{u}{\mu u + 1}$. The hedging equation

$$\nabla_\mu \mathbb{E}_{\pi_\mu}[r_y(X)] = \mathbb{E}_{u \sim f_\mu}\left[r_t(u) \left(u - 1 + \frac{u}{\mu u + 1} \right) \right] = 0 \tag{49}$$

In an analogous way to the previous hedging equations, we solve the residual

$$R(\mu) = \sum_{i=1}^{M} r_t(u_i)\psi(u_i, \mu)p_\mu(u_i)\Delta u \tag{50}$$

to locate $\mu^\dagger$ such that $R(\mu^\dagger) = 0$. This $\mu^\dagger$ is the Poisson optimal hedge. □

D. Experimental Details

In this section, we provide additional details on our experimental setup. We provide our code here.

D.1. Toy Example

In Figure 1, we present a toy example with one-dimensional rewards defined on $[0, 1]$. We set the proxy reward to be $r_p(x) = x$ and the gold reward

$$r_t(x) \;=\; \frac{x^p\,(1-x)}{C}, \quad C \;=\; \Big(\frac{p}{p+1}\Big)^p \,\frac{1}{p+1}$$

where C is a normalization constant so that the gold reward is bounded between 0 and 1 for convenience. We choose this gold reward as the reward under the Best-of-n distribution has a simple closed-form solution. We set $p = 12$ and we find the expected value of true reward and the KL divergence with respect to the reference distribution under four mechanisms:

1. **Tilted Distribution:** Exponential tilting of the gold reward, $Q_\lambda(x) \propto \exp(\lambda\, r_t(x))$ and of the proxy reward, $Q_\lambda(x) \propto \exp(\lambda\, r_p(x))$.
2. **Best-of-n (BoN):** Selection of the maximum of n i.i.d. uniform draws.
3. **Soft Best-of-n (SBoN):** Softmax–based sampling of n i.i.d. uniform draws with inverse temperature λ.
4. **Best-of-Poisson (BoP):** Selection of the maximum of n i.i.d uniform draws where n is drawn from a Poisson distribution with rate μ.

Below we detail each one in turn.

Tilted Distribution We first consider tilting with respect to the gold reward, defined as:

$$Q_\lambda(x) = \frac{e^{\lambda\, r_t(x)}}{Z(\lambda)}, \quad Z(\lambda) = \int_0^1 e^{\lambda\, r_t(x)}\, dx,$$

We then compute

$$\mathbb{E}_{Q_\lambda}[r_t] = \frac{1}{Z(\lambda)} \int_0^1 r_t(x)\, e^{\lambda\, r_t(x)}\, dx, \qquad D_{\mathsf{KL}}(Q_\lambda \,\|\, U[0,1]) = \lambda\, \mathbb{E}_{Q_\lambda}[r_t] \;-\; \ln Z(\lambda).$$

All integrals are evaluated via numerical quadrature on $[0, 1]$. Analogously, letting

$$Z_{\text{proxy}}(\lambda) = \int_0^1 e^{\lambda\, r_p(x)}\, dx = \begin{cases} 1, & \lambda = 0, \\ \frac{e^\lambda - 1}{\lambda}, & \lambda \neq 0, \end{cases}$$

we define

$$Q_\lambda^{\text{proxy}}(x) = \frac{e^{\lambda x}}{Z_{\text{proxy}}(\lambda)},$$

and compute $\mathbb{E}_{Q_\lambda^{\text{proxy}}}[r_t]$ and its KL with respect to the reference distribution (the uniform distribution).

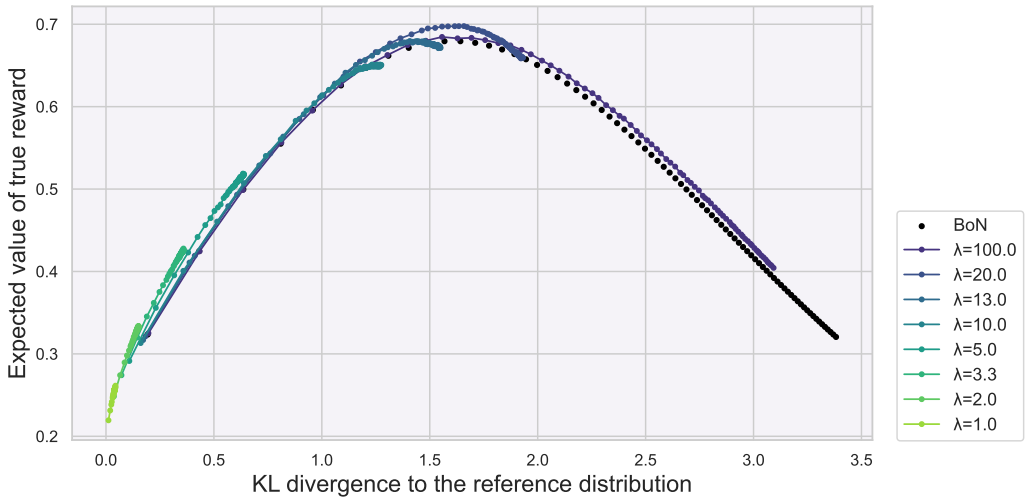


Figure 6. Expected value of true reward versus KL divergence to the reference distribution using both Best-of-n and Soft Best-of-n in the setup presented in Sec. D.1. For the same range of $n = \{1, \cdots, 80\}$, Soft BoN *softens* the winner's curse with an appropriate λ while attaining a competitive reward vs KL tradeoff.

Best-of-n For n i.i.d. draws $X_{1:n} \sim U[0,1]$, the max has density $n\,x^{n-1}$. One shows

$$\mathbb{E}[r_t(\max X_{1:n})] = \frac{n}{C\,(n+p)\,(n+p+1)}, \qquad D_{\mathsf{KL}} = \ln n - 1 + \frac{1}{n}.$$

Soft-BoN We draw $m = 1.2 \times 10^5$ samples of $(X_{1:n})$, apply a softmax with temperature λ, and numerically estimate $\mathbb{E}[r_t]$ and D_{KL}. We sweep λ over 600 log-spaced values in $[10^{-4}, 10^4]$.

Best-of-Poisson (BoP) We treat the number of draws as distributed from a Poisson distribution $n \sim \text{Poisson}$ with sampling density $q_\lambda(x) = (\lambda x + 1)\, e^{\lambda(x-1)}$, from which $\mathbb{E}[r_t]$ and D_{KL} are again computed by numerical quadrature. We sweep μ over 800 uniformly-spaced values in [0, 300].

We apply `HedgeTune` as presented in Alg. 4 for BoN and BoP. In both cases, we attain an operating point which corresponds exactly to the true hacking threshold as shown in Fig. 1. The success of this algorithm hinges on Theorem 1 which guarantees an existence of (at least) one hacking threshold. However, this guarantee does not hold for Soft BoN. Therefore, the optimization problem becomes much more challenging, and using vanilla estimators for the density causes numerical instabilities when applied for Soft BoN.

D.2. Synthetic setup

We randomly select a single Reddit post from the validation split of the `trl-internal-testing/tldr-preference-trl-style` dataset on Hugging Face. Using this prompt (see Figure 7), we generate $60,000$ candidate summaries from a supervised fine-tuned Pythia-1.4B model (`cleanrl/EleutherAI_pythia-1b-deduped__sft__tldr` on HuggingFace) via ancestral sampling with temperature $T = 1$ and a top-k filter of $k = 50$ (see examples of candidate summaries in Figure 8). After generating candidate summaries, we annotate them by scoring with a reward model trained on summarization task. Concretely, we use `cleanrl/EleutherAI_pythia-1b-deduped__reward__tldr` found on Hugging Face.

Using this annotated data, we would like to empirically observe the *winner's curse*. Methods like Best-of-n that aggressively optimize for the proxy are particularly vulnerable to this phenomenon. We artificially miscalibrate the proxy reward model at the extreme high-end of its scores. Concretely, let P_α denote the top α%-percentile of proxy scores. For any response whose proxy score lies in P_α, we replace its proxy score by a *negatively* linear function of its true reward (so that higher

Below is a reddit POST and the corresponding SUBREDDIT and TITLE.
Write a both precise and concise summary of the contents of the post.
—
SUBREDDIT: r/relationships
TITLE: I [23F] have just come out of 8 year relationship. Feel like I don't know how to date/flirt. Scared will grow old with many cats. Any advice?
POST: This is my first post so please be kind :)

I know that lots of people often feel confused when they come out of a long-term relationship. They think they have forgotten how to be single, or how to flirt/date.

I am one of these people.

The problem is, my relationship started when I had just turned 16. I have never been single—as an adult. That might sound silly. But the only time I have ever flirted or dated was as an over-confident, hormone-riddled teenager.

Now I have a pretty demanding job, responsibilities blah blah... And I just don't know how to this!

I'm in no way in a rush to get into a new relationship, but that doesn't mean I want to be completely alone in the meantime.

If anyone has experienced anything similar, or just generally has some advice, it would be greatly appreciated!
—
Summary:

Figure 7. Prompt template used to summarize a selected Reddit post

true-reward responses receive lower proxy scores). All other responses retain their true reward as the proxy. Formally,

$$r_p(x) = \begin{cases} a - b\, r_t(x), & \text{if } r_t(x) \geq P_\alpha, \\ r_t(x), & \text{otherwise,} \end{cases}$$

where $a, b > 0$ are chosen so that this mapping spans the same range as the original scores. We then map both distributions to uniform percentiles via their sorted ranks, yielding two arrays in $[0, 1]$ amenable to direct comparison. We consider α in the set $\{80, 85, 90, 95\}$, simulating increasingly egregious misalignment at the top end of the proxy. This synthetic setup let us measure how each selection strategy (BoN, SBoN, and BoP) degrades in true-reward performance as the proxy becomes more misleading among its highest-ranked candidates.

We define our sampling grids as $n \in \{1, \ldots, 1000\}$ (BoN), $\lambda \in [0, 20]$ (SBoN), $\mu \in [0, 50]$ (BoP) each discretized into 300, 150, and 250 points respectively. For every parameter setting and for each bootstrap replicate ($B = 300$), we apply the selection mechanism, record the chosen item's true reward, and check whether this true reward exceeds the true reward of a randomly selected response by at least 10%. We aggregate these outcomes to estimate, for each combination, the probability of at least 10% true-reward improvement, and compute 95% confidence intervals via a normal approximation. We consider this probability as a calibrated reward. In Figure 3, we present the proxy and true (calibrated) rewards using BoN and BoP where we vary the average number of samples n. Moreover, we find the value of n corresponding to the peak true reward for each case using `HedgeTune`. Meanwhile, in Figure 4, we compare the performance of Soft Best-of-n mechanism with an optimized temperature $\lambda^\dagger$ to Best-of-n. We do so by plotting the gap between the (calibrated) reward attained by SBoN and that attained with BoN. With larger n, Best-of-n is more likely to fall victim to the winner's curse, making hedging an effective solution to improve performance.

Summary 1: " I am a 23/F looking for advice on how to find potential lovers. I think I find no one, and it frustrates me. Any advice?"

Summary 2: "I just came out of a relationship, a really long one. I know I'm young but I feel like I must be naive and lack experience. Please give me anything you can!"

Summary 3: "Teenage dating syndrome. Don't know-how-to- avoid it. Care to chat and/or share stories about your dating/relationship experiences?"

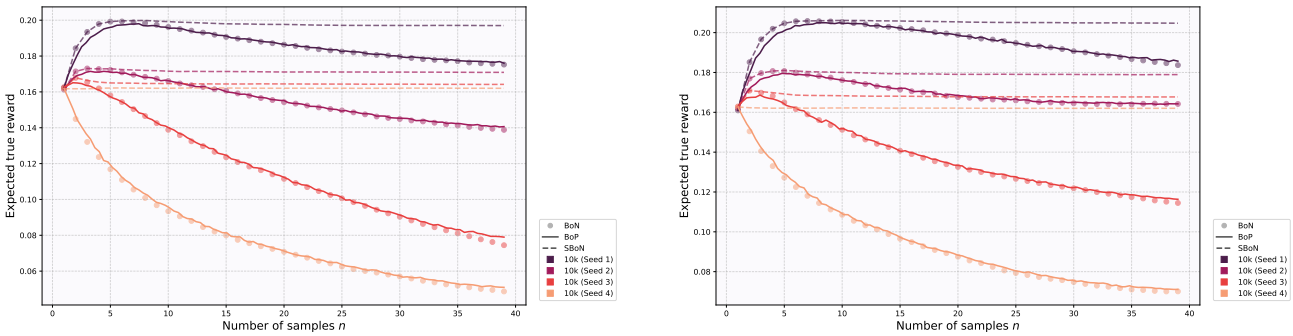
Figure 8. Example of generated summaries of the Reddit post shown in Fig. 7

D.3. Reward hacking in the wild

To observe reward hacking *in the wild*, we follow the setup of Coste et al. (Coste et al., 2024). We first use an annotated dataset provided by (Coste et al., 2024) which contains 1,000 prompts from the validation split of AlpacaFarm dataset, along with 12,600 response generations per prompt from a 1.4b fine-tuned Pythia model. Each prompt-response pair is labeled with the AlpacaFarm `reward-model-human` to give 'gold' scores. Next, we would like to train proxy reward models on the preferences of this true reward. We randomly sample a prompt with two responses from the annotated dataset and curate a dataset of the form (prompt, chosen, rejected) where the chosen response is the response with the higher gold reward score. We follow this procedure to curate four datasets with varying sizes (10k, 20k, 46k, 80k). For each dataset, we consider two variants: one with no label noise and one with random 25% label noise. Next, we use the code kindly provided by the authors of (Coste et al., 2024) in their Github repository to train proxy reward models with the different datasets over four random seeds (1, 2, 3 and 4) using their default hyperparameters (e.g., 10^{-5} learning rate and five epochs). Lastly, we score the annotated dataset using the trained proxy reward models. The end result is a set of 800 prompts, 12 600 responses per prompt, along with gold and proxy scores for each prompt-response pair.

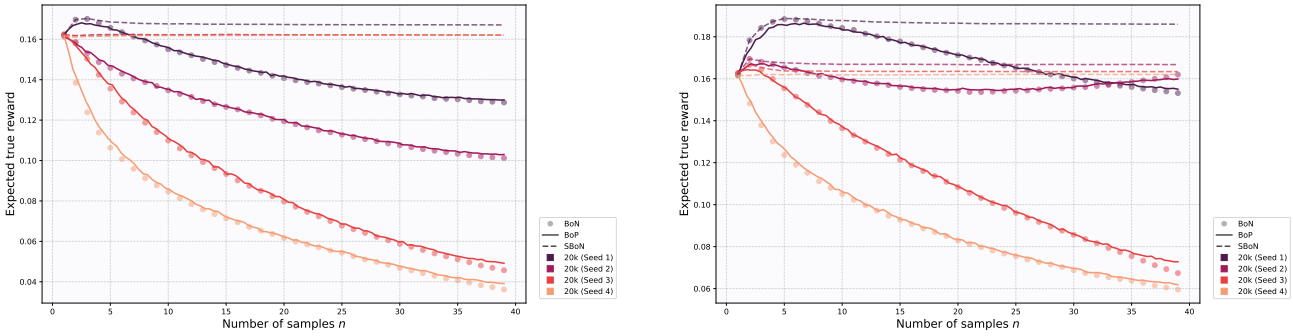
While reward hacking can appear without label noise (see left panels of Figures 9 and 10), reward hacking is more pronounced with label noise as expected. Moreover, reward hacking is more apparent when the proxy reward is trained on **less** data. One potential explanation is that, with fewer training examples, the proxy is less well-calibrated and its estimation errors vary more sharply across inputs. In that case, a small n might produce a deceiving reward gain. In contrast, errors may surface early on with a large training dataset, so true reward declines immediately as sampling increases. In cases of reward hacking, we see that SBoN with an appropriately chosen λ can (1) achieve the maximum reward achieved by BoN/BoP and (2) mitigate reward hacking, as shown with the reward almost flatlining after it reaches its peak value. We also witness cases where the proxy reward **always** misaligns with the gold reward, causing a collapse of true reward from the onset of BoN. In that case, the optimal hedging behavior is a uniform selection over responses, which is recovered with $n = 1$ for BoN or $\lambda = 0$ for SBoN.

Interestingly, we observe instances of what we call **reward grokking** as shown in the right panels of Figures 10 and 12, where the true reward decreases or flat-lines across low- to mid-range sample counts, only to undergo a sudden uptick at higher sample regimes, revealing a delayed but apparent realignment of proxy and true objectives. We leave detailed investigation of reward grokking and its implications for hedging strategies to future work.



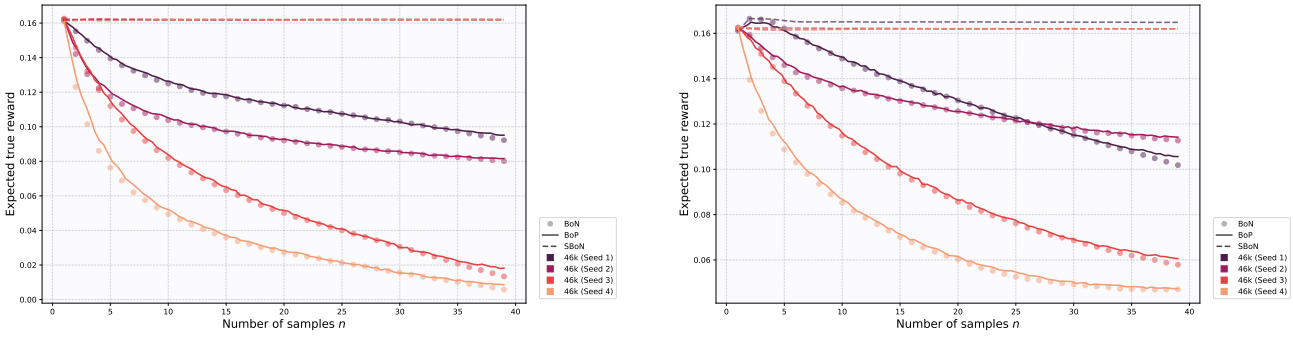
(a) No label noise **(b)** 25% label noise

Figure 9. Expected true-reward vs. average number of samples with proxy trained on 10 000 examples: (a) without label noise; (b) with 25% label noise.



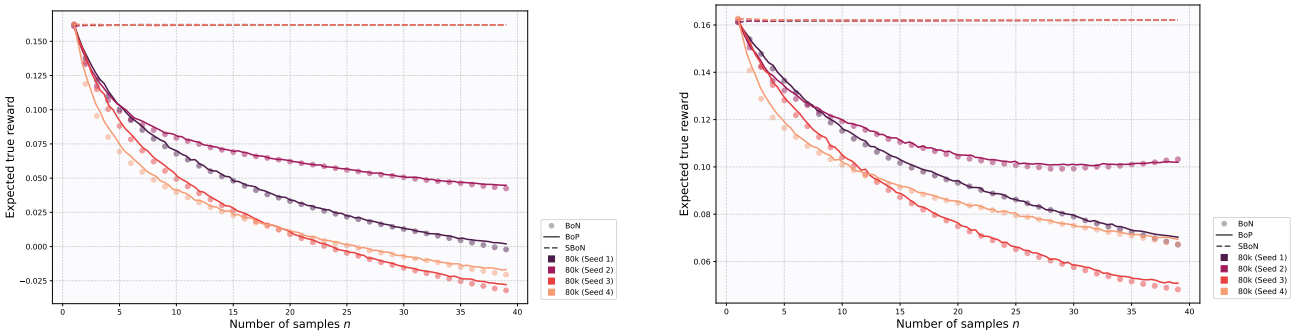
(a) No label noise **(b)** 25% label noise

Figure 10. Expected true-reward vs. average number of samples with proxy trained on 20 000 examples: (a) without label noise; (b) with 25% label noise.



(a) No label noise **(b)** 25% label noise

Figure 11. Expected true-reward vs. average number of samples with proxy trained on 46 000 examples: (a) without label noise; (b) with 25% label noise.



(a) No label noise **(b)** 25% label noise

Figure 12. Expected true-reward vs. average number of samples with proxy trained on 80 000 examples: (a) without label noise; (b) with 25% label noise.