Better Estimation of the Kullback–Leibler Divergence Between Language Models

Afra Amini **Tim Vieira** **Ryan Cotterell**
ETH Zürich
{afra.amini, ryan.cotterell}@inf.ethz.ch
tim.f.vieira@gmail.com

Abstract

Estimating the Kullback–Leibler (KL) divergence between language models has many applications, e.g., reinforcement learning from human feedback (RLHF), interpretability, and knowledge distillation. However, computing the exact KL divergence between two arbitrary language models is intractable. Thus, practitioners often resort to sampling-based estimators. While it is easy to fashion a simple Monte Carlo (MC) estimator that provides an unbiased estimate of the KL divergence between language models, this estimator notoriously suffers from high variance and can even result in a negative estimate of the KL divergence, a non-negative quantity. In this paper, we introduce a Rao–Blackwellized estimator that is unbiased and provably has variance less than or equal to that of the standard Monte Carlo estimator. In an empirical study on sentiment-controlled fine-tuning, we show that our estimator provides more stable KL estimates and reduces variance substantially. Additionally, we derive an analogous Rao–Blackwellized estimator of the gradient of the KL divergence, which leads to more stable training and produces models that more frequently appear on the Pareto frontier of reward vs. KL compared to the ones trained with the MC estimator of the gradient.

https://github.com/rycolab/kl-rb

1 Introduction

The Kullback–Leibler [KL; 19] divergence is a statistical divergence that quantifies how one probability distribution differs from another. Measuring the KL divergence between probability distributions is a well-established problem that has been studied extensively in the statistics literature [7, 12, *inter alia*]. In some special cases, e.g., in the case that we wish to measure the KL divergence between two Gaussian measures, the KL divergence has an analytical solution. However, in the general case, exact computation of the KL divergence is not analytically tractable or approximable with an efficient algorithm [14]. This paper treats the case of computing the KL divergence between two language models (LMs), a fundamental task in natural language processing with numerous practical applications.

The KL divergence plays a central role across multiple applications. In reinforcement learning from human feedback [RLHF; 6, 26, 35], it is used as a regularization term to constrain the fine-tuned model from drifting too far from a reference model, preserving fluency and preventing reward over-optimization. In interpretability research, KL divergence quantifies how a specific prompt shifts the model distribution by comparing the model's distributions before and after controlled interventions [9, 27, 36]. As an evaluation metric, KL divergence is used to assess how well language models approximate target distributions [4, 37]. In knowledge distillation, KL divergence is minimized to align a student model with a teacher model [1].

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

The above applications demonstrate that measuring the KL divergence between two language models is useful and widespread. However, in the case of neural language models, it is far from straightforward. It is easy to see why: Given an alphabet of symbols Σ and two language models p and q, distributions over Σ^*, the **KL divergence** is given by the following expression:[1,2]

$$\mathrm{KL}(p \mid\mid q) \stackrel{\text{def}}{=} \sum_{\boldsymbol{y} \in \Sigma^*} p(\boldsymbol{y}) \log \frac{p(\boldsymbol{y})}{q(\boldsymbol{y})}. \tag{1}$$

Recalling that Σ^* is a countably infinite set, we cannot expect, in general, to compute Eq. (1) exactly in finite time without additional assumptions.[3] While in some very special cases, e.g., where p and q are deterministic finite-state automata, there exist efficient algorithms, [4, 20, 22], we should not expect such an algorithm to exist in the case where p and q are neural language models, e.g., those based on the transformer [25, 28, 39]. Thus, most researchers turn to approximation, with Monte Carlo estimation being the most widely used method.

The Monte Carlo (MC) estimator for KL divergence (Eq. (1)) involves sampling M strings $\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(M)} \overset{\text{i.i.d.}}{\sim} p$ and then averaging $\log \frac{p(\boldsymbol{Y}^{(m)})}{q(\boldsymbol{Y}^{(m)})}$. Even though this estimator is unbiased, it often exhibits high variance, which means the approximation can be noisy and unreliable.[4] More pathologically, the naive MC estimator can result in negative estimates of KL, which may be undesirable in practice.[5] To address these issues, practitioners adopt alternative techniques to ensure non-negativity. For example, Schulman [33] proposes an unbiased, non-negative KL estimator that is widely used in practice [13, 40]. However, the proposed method, in its original form, does not theoretically yield an estimator with lower variance, and, as we show empirically, can exhibit enormous variance.

In this paper, we derive an improved estimator of KL using Rao–Blackwellization [RB; 3, 5, 11, 30], a well-established variance-reduction technique from statistics.[6] This results in an estimator that is provably unbiased *and* has a variance that is always less than or equal to that of the standard Monte Carlo estimator, while requiring no additional computational overhead. As a point of comparison to our RB estimator, we also provide a comprehensive formal analysis of various existing methods for estimating KL divergence, examining their bias and variance.

We empirically validate our theoretical findings using the sentiment-controlled generation task [29] as a testbed. Specifically, we measure the KL divergence between a GPT-2 model [28] before and after fine-tuning, where the fine-tuning objective is to steer the model toward generating positive movie reviews. Our experimental results confirm that our proposed estimator significantly reduces the variance of the Monte Carlo estimator, yielding the most stable and reliable estimates among all methods studied. In contrast, alternative estimators from the literature fail to achieve meaningful variance reduction, and in some cases, lead to unbounded variance. We further examine how using our derived estimator in the fine-tuning loop of RLHF impacts the downstream performance. Our results suggest that using our Rao–Blackwellized estimator reduces the instability across different RLHF runs. We further look

[1] Throughout this paper $\log$ denotes the natural logarithm function; thus, KL divergence is measure in *nats* rather than *bits*. We also note that terms of the form $p(\boldsymbol{y}) \log \frac{p(\boldsymbol{y})}{q(\boldsymbol{y})}$ in Eq. (1) where $p(\boldsymbol{y}) = 0$ can *correctly* be taken to equal zero because $\lim_{p \to 0^+} p \log p/q = 0$.

[2] In conditional tasks like dialogue generation, language models are prompted with an input $\boldsymbol{x} \in \Sigma^*$, inducing a conditional distribution $p(\cdot \mid \boldsymbol{x})$. KL divergences are typically averaged over a set of prompts. For simplicity, we omit $\boldsymbol{x}$ in notation and write $p(\boldsymbol{y})$; all of our results extend straightforwardly to the conditional case.

[3] In general, computing the KL divergence between two arbitrary LMs *exactly* is undecidable. To see why, assume that each of the two language models is a probabilistic context-free grammar. In this case, deciding whether their KL divergence is zero is undecidable, as it follows directly from the undecidability of testing equivalence between two unweighted context-free grammars [15]. In the more restrictive case of probabilistic finite-state language models, it is PSPACE-hard. Importantly, however, the intractablity of *exact* computation does not imply that *approximation* is intractable. In practice, one can often obtain good Monte Carlo estimates of the KL divergence, provided that its variance is well-controlled. We study very practical methods for improving variance in this paper.

[4] Monte Carlo estimation formally requires that the underlying random variable have finite variance; if the variance is unbounded, the estimator no longer converges in the limit.

[5] Since the KL divergence is non-negative by definition, a negative estimate can be problematic when KL is used as part of a loss function, as it may destabilize the learning dynamics.

[6] Despite its simplicity, our proposed estimator is absent from existing literature and open-source RLHF libraries [13, 17, 34, 40], highlighting a gap we believe is worth addressing.

at the Pareto frontier of average rewards achieved by the model vs. its KL divergence with the reference model. We observe that models fine-tuned using the RB estimator appear significantly more often on the Pareto frontier of reward vs. KL compared to the models fine-tuned with the MC estimator.

2 Preliminaries

2.1 Language Models

Let Σ be an **alphabet**, a finite, non-empty set of symbols. A **string** is a finite sequence of symbols from Σ. Let Σ^* denote the set of all such strings. A **language model** p is a distribution over Σ^*. The **prefix probability** function $\vec{p}$ of a prefix $\boldsymbol{x} \in \Sigma^*$ is

$$\vec{p}(\boldsymbol{x}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{y} \in \Sigma^*} p(\boldsymbol{x}\boldsymbol{y}), \tag{2}$$

which is the cumulative probability of all strings in the language that have $\boldsymbol{x}$ as their prefix. We denote the **conditional prefix probability** as $\vec{p}(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\vec{p}(\boldsymbol{x}\boldsymbol{y})}{\vec{p}(\boldsymbol{x})}$, where we additionally define $\vec{p}(\text{EOS} \mid \boldsymbol{y}) \stackrel{\text{def}}{=} \frac{p(\boldsymbol{y})}{\vec{p}(\boldsymbol{y})}$. Language models can be factored into the product of distributions using the chain rule of probability, i.e., for any string $\boldsymbol{y} = y_1 \cdots y_N \in \Sigma^*$ we can write

$$p(\boldsymbol{y}) = \vec{p}(\text{EOS} \mid \boldsymbol{y}) \prod_{n=1}^{N} \vec{p}(y_n \mid \boldsymbol{y}_{<n}), \tag{3}$$

where $\boldsymbol{y}_{<n} \stackrel{\text{def}}{=} y_1 \cdots y_{n-1}$ and $\text{EOS} \notin \Sigma$ is a distinguished end-of-string symbol. Let $\overline{\Sigma} \stackrel{\text{def}}{=} \Sigma \cup \{\text{EOS}\}$. In Eq. (3), each $\vec{p}(\cdot \mid \boldsymbol{y}_{<n})$ can fruitfully be viewed as a distribution over $\overline{\Sigma}$. Despite the overloading of the notation, whether $\vec{p}(\cdot \mid \boldsymbol{y}_{<n})$ refers to a prefix probability, a function which takes an argument from Σ^*, or a distribution over $\overline{\Sigma}$ will always be clear from context. Throughout this paper, we use $\boldsymbol{Y}$ to represent the string-valued random variable sampled from p. When taking M i.i.d. samples from p, we use $\boldsymbol{Y}^{(m)}$ to denote the m^{th} sample.

2.2 Monte Carlo KL Estimation

A simple way to estimate the KL divergence is with the **Monte Carlo estimator (MC)** defined as

$$\mu_{\text{MC}} = \frac{1}{M} \sum_{m=1}^{M} \log \frac{p(\boldsymbol{Y}^{(m)})}{q(\boldsymbol{Y}^{(m)})} = \frac{1}{M} \sum_{m=1}^{M} f(\boldsymbol{Y}^{(m)}), \tag{4}$$

where $\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(M)} \overset{\text{i.i.d.}}{\sim} p$ and $f(\boldsymbol{Y}) \stackrel{\text{def}}{=} \log \frac{p(\boldsymbol{Y})}{q(\boldsymbol{Y})}$. Throughout the paper, we assume that the KL divergence is finite, i.e., $\text{KL}(p \mid\mid q) < \infty$. It is straightforward to show μ_{MC} is unbiased, i.e., $\mathbb{E}[\mu_{\text{MC}}] = \text{KL}(p \mid\mid q)$ and the variance of this estimator is $\text{Var}[\mu_{\text{MC}}] = \frac{1}{M} \text{Var}[f(\boldsymbol{Y})]$. In App. A, we discuss the Horvitz–Thompson estimator, another unbiased KL estimator.

Note that while the exact KL value is always non-negative, $f(\boldsymbol{Y})$ may be positive or negative. Consequently, the Monte Carlo estimate μ_{MC} may also be negative. This happens because the estimate is based on a limited number of samples, and some sample draws can lead to negative values. This can be problematic during RLHF, which depends on the KL divergence being non-negative.

2.3 Control Variate Monte Carlo Estimation

A general approach to reduce estimator variance is through *control variates* [32, §8.2]. For KL divergence between language models, this technique was popularized by Schulman [33] and is widely used in RLHF libraries [13, 17, 34]. Formally, a **control variate** is any function $g\colon \Sigma^* \to \mathbb{R}$ for which $G \stackrel{\text{def}}{=} \mathbb{E}[g(\boldsymbol{Y})]$ can be efficiently computed.[7] We define the **control variate Monte Carlo estimator** as

$$\mu_{\text{CV}} = \frac{1}{M} \sum_{m=1}^{M} f(\boldsymbol{Y}^{(m)}) + \alpha \cdot (g(\boldsymbol{Y}^{(m)}) - G). \tag{5}$$

[7] One could also consider control variates of the form $g\colon \Sigma^* \to \mathbb{R}^d$ for $d > 1$ [10].

where $\alpha \in \mathbb{R}$ is a calibration parameter that must be chosen *a priori*. The proposition below characterizes the variance of μ_{CV} as a function of α, which will tell use how to choose α optimally.

Proposition 1. *Consider the control variate MC estimator μ_{CV} defined in Eq. (5), and assume that $\mathbb{E}[g(\boldsymbol{Y})] < \infty$. Then μ_{CV} is an unbiased estimator, and its variance is given by*

$$\text{Var}[\mu_{\text{CV}}] = \frac{\text{Var}[f] + \alpha^2 \,\text{Var}[g] + 2\alpha \,\text{Cov}[f, g]}{M}. \tag{6}$$

Proof. See App. B. ■

Assume $0 < \text{Var}[g] < \infty$. It is straightforward to show that $\alpha^* \stackrel{\text{def}}{=} -\,\text{Cov}[f, g]/\,\text{Var}[g]$ is the value that minimizes the variance. If we plug α^* in Eq. (6), and simplify, we see that

$$\text{Var}[\mu_{\text{CV}}] = \frac{1}{M}\,\text{Var}[f]\Big(1 - \text{Corr}[f, g]^2\Big), \tag{7}$$

which directly translates to reducing the variance of the MC estimator. The magnitude of the correlation between f and g determines the degree of variance reduction. Note that the value of α^* may be estimated from a pilot sample when it cannot be computed analytically.[8]

KL Estimation with a Control Variate. A specific control variate for KL estimation was proposed by Schulman [33], who defined $g(\boldsymbol{Y}) = \frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})}$. Substituting this into Eq. (5), the MC estimator of the KL divergence with this control variate is

$$\mu_{\text{CV}} = \frac{1}{M}\sum_{m=1}^{M} \log \frac{p(\boldsymbol{Y}^{(m)})}{q(\boldsymbol{Y}^{(m)})} + \alpha \cdot \left(\frac{q(\boldsymbol{Y}^{(m)})}{p(\boldsymbol{Y}^{(m)})} - 1\right). \tag{8}$$

Remarks. This proposal has some notable properties. First, $G = \mathbb{E}\left[\frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})}\right] = 1$, meaning G is known in advance.[9] Second, $\text{Cov}[f, g] = 0$ is zero if and only if p is equal to q (Prop. 8), meaning that when the two distributions are not equal, and α is chosen suitably, we are guaranteed to *strictly* reduce variance. Schulman [33] proposes setting $\alpha = 1$; the benefit of this suboptimal choice is that the resulting estimator is always non-negative (Prop. 7).[10] However, setting $\alpha = 1$ will *increase* variance when $\alpha^* < \frac{1}{2}$ (Remark 10). Indeed, our experiments (§5.1) confirm that $\alpha = 1$ is a poor choice in practice—it is better to estimate α^*[11] to ensure that the control variate is correctly calibrated.

3 Rao–Blackwellized Monte Carlo

In this article, we propose the application of another classical technique to reduce the variance of the Monte Carlo estimation of the KL divergence—**Rao–Blackwellization** [RB; 5, 11]. Despite its standing in the statistics literature, a Rao–Blackwellized Monte Carlo estimator has yet to gain traction in the context of RLHF [13, 17, 34, 40].

We define the **Rao–Blackwellized Monte Carlo estimator** μ_{RB} as follows:

$$\mu_{\text{RB}} \stackrel{\text{def}}{=} \frac{1}{M}\sum_{m=1}^{M} \sum_{n=1}^{|\boldsymbol{Y}^{(m)}|} \text{KL}(\vec{p}(\cdot \mid \boldsymbol{Y}^{(m)}_{<n}) \| \vec{q}(\cdot \mid \boldsymbol{Y}^{(m)}_{<n})). \tag{9}$$

The key benefit of this estimator is that we *analytically* compute the expectation over the n^{th} symbol in each string rather than relying on the single sampled value at that position.

Rao–Blackwellization Background. The starting point of Rao–Blackwellization is the following inequality involving the conditional variance: $\text{Var}[\mathbb{E}[\mu \mid T]] \leq \text{Var}[\mu]$, where μ is an unbiased estimator of $\mathbb{E}[f]$ and T is a statistic[12] for which we can explicitly compute $\mathbb{E}[\mu \mid T]$. This

[8] Note that the control variate method can be straightforwardly extended to support multiple control variates.

[9] Note that for this to hold, we need have $q(\boldsymbol{y}) = 0$ whenever $p(\boldsymbol{y}) = 0$, which is different from the support condition we assumed for $\text{KL}(p \,||\, q) < \infty$.

[10] Note that $\alpha = 1$ is the only value of α that guarantees nonnegativity (see proof of Prop. 7).

[11] Prop. 9 provides the exact conditions on α required for a variance reduction.

[12] Note that T is a function of $\{\boldsymbol{Y}^{(m)}\}_{m=1}^{M}$. We have suppressed this function's arguments in our notation for improved readability.

technique is often referred to as *Rao-–Blackwellization* because the inequality is associated with the Rao–Blackwell theorem [21].[13]

Notation. Before moving forward, we introduce some convenient notation. Let $\overline{\boldsymbol{Y}}^{(m)}$ denotes the EOS-padded version of $\boldsymbol{Y}^{(m)}$. Additionally, we extend the definition of the prefix probabilities $\vec{p}$ and $\vec{q}$ to strings containing padding symbols, i.e., $\boldsymbol{y} \notin \Sigma^*$, with the following additional case: $\vec{p}(y \mid \boldsymbol{y}) \stackrel{\text{def}}{=} \mathbb{1}\{y = \text{EOS}\}$, and $\vec{q}(y \mid \boldsymbol{y}) \stackrel{\text{def}}{=} \mathbb{1}\{y = \text{EOS}\}$.

Understanding Our Rao–Blackwellized Estimator. We now present a proof that our Rao–Blackwellized estimator μ_{RB} is unbiased and indeed does result in a variance reduction. One might wonder why this requires more than a straightforward application of the Rao–Blackwell theorem. The reason is that μ_{RB} does not arise directly from the standard formulation of Rao–Blackwellization. Instead, we apply Rao–Blackwellization separately to each summand, where each summand corresponds to an estimator over strings of a particular length. In general, Rao–Blackwellizing the summands pointwise does not guarantee a reduction in the variance of their sum, since the summands may be correlated. However, in the specific case of μ_{RB}, we *can* prove that the overall variance is reduced, despite Rao–Blackwellizing the summands independently, as we state in the following theorem.

Theorem 2. *Suppose the MC estimator μ_{MC} has finite variance, i.e., $\text{Var}[\mu_{\text{MC}}] < \infty$. Then the following properties regarding μ_{RB} hold:*

(i) $\mathbb{E}[\mu_{\text{RB}}] = \text{KL}(p \,||\, q)$ *(unbiasedness)* $\quad$ (ii) $\text{Var}[\mu_{\text{RB}}] \leq \text{Var}[\mu_{\text{MC}}]$ *(variance reduction)*

Proof. See App. C for a detailed proof. However, we provide the proof sketch below for the reader to quickly understand the structure of the argument, which is broken down into three steps.

(1) **Step-wise Estimation.** We begin by Rao–Blackwellizing the **step-wise Monte Carlo estimator** for any $n > 0$,

$$\mu_{\text{MC}}^n \stackrel{\text{def}}{=} \frac{1}{M} \sum_{m=1}^{M} \log \frac{\vec{p}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})}{\vec{q}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})}. \tag{10}$$

Intuitively, μ_{MC}^n measures the average KL of the n^{th} symbol. The next step is to define $T_n(\overline{\boldsymbol{Y}}) = \overline{\boldsymbol{Y}}_{<n}$ and apply Rao–Blackwellization to each μ_{MC}^n as follows:

$$\mu_{\text{RB}}^n \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^{(1)},\ldots,\overline{\boldsymbol{Y}}^{(M)}} [\mu_{\text{MC}}^n \mid T_n] \tag{11a}$$

$$= \frac{1}{M} \sum_{m=1}^{M} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^{(m)}} \left[\log \frac{\vec{p}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})}{\vec{q}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})} \middle| T_n(\overline{\boldsymbol{Y}}^{(m)}) \right] \tag{11b}$$

$$= \frac{1}{M} \sum_{m=1}^{M} \text{KL}(\vec{p}(\cdot \mid \overline{\boldsymbol{Y}}_{<n}^{(m)}) \| \vec{q}(\cdot \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})). \tag{11c}$$

Now, it is clear from the Rao–Blackwellization theorem that μ_{RB}^n is unbiased and it provides a variance reduction (i.e., $\text{Var}[\mu_{\text{RB}}^n] \leq \text{Var}[\mu_{\text{MC}}^n]$) for all $n > 0$. Inuitively, the source of the variance reduction in the μ_{RB}^n estimator is that we compute the expectation over the n^{th} symbol exactly rather relying on the sampled symbol at that position.

(2) **Truncated Estimation.** Next, we define $\mu_{\text{MC}}^{(N)} \stackrel{\text{def}}{=} \sum_{n=1}^{N} \mu_{\text{MC}}^n$ and $\mu_{\text{RB}}^{(N)} \stackrel{\text{def}}{=} \sum_{n=1}^{N} \mu_{\text{RB}}^n$. Intuitively, these estimators target the KL divergence for symbols up to a maximum length of N. In Lemma 11, we prove that $\mathbb{E}[\mu_{\text{RB}}^{(N)}] = \mathbb{E}[\mu_{\text{MC}}^{(N)}]$, and $\text{Var}\left[\mu_{\text{RB}}^{(N)}\right] \leq \text{Var}\left[\mu_{\text{MC}}^{(N)}\right]$ for all $N > 0$ using the law of total expectation and Jensen's inequality, where the latter is used in a manner analougsly to have it is used in the original Rao–Blackwellization theorem.

[13] Note that in the Rao–Blackwell theorem, we get the stronger result that when T is a sufficient statistic, $\text{Var}[\mathbb{E}[\mu \mid T]]$ is optimal, i.e., it is the minimal-variance, unbiased estimator. However, we can perform Rao–Blackwellization even when T is not sufficient and are still guaranteed that the variance is no worse [31].

(3) **Complete Estimation.** Now, we consider the complete estimation. First, we observe that the limit of the truncated estimators converge to $\mu_{\text{MC}} = \lim_{N\to\infty} \mu_{\text{MC}}^{(N)}$ (Lemma 12), and analogously, $\mu_{\text{RB}} = \lim_{N\to\infty} \mu_{\text{RB}}^{(N)}$. Thereby, we are able to show that μ_{RB} is an unbiased estimator of the KL divergence with variance less than or equal to that of μ_{MC}.

■

Remarks. Notably, μ_{RB} is guaranteed to be non-negative, a property desired by some when designing estimators for the KL divergence between two language models, as the KL divergence itself is always non-negative (cf. remarks in §2.3). In the case of our Rao–Blackwellized estimator, non-nonegativity follows from the fact that each step-wise estimator computes the *exact* KL divergence between the two next-symbol distributions, conditioned on the sampled context $\boldsymbol{y}_{<n}^{(m)}$. Since all terms in Eq. (11c) are non-negative, μ_{RB} remains non-negative as well.

Complexity Analysis. Although computing μ_{RB} might seem more expensive than μ_{MC}, the overall runtime is dominated by the cost of forward passes. Because a forward pass already involves producing the full distribution over $\overline{\Sigma}$ at each position n, the additional $\mathcal{O}\big(MN|\overline{\Sigma}|\big)$ work required by μ_{RB} is negligible compared to the M forward passes needed for both μ_{CV} and μ_{MC}.

4 Estimating the Gradient

KL estimation is essential in many applications, especially in fine-tuning large language models. In reinforcement learning from human feedback (RLHF), for example, the objective includes a KL regularization term to balance reward maximization with staying close to a reference model. Since the language model is a differentiable function of parameters $\boldsymbol{\theta}$ optimized via gradient descent, this setup requires computing the gradient of the KL divergence with respect to $\boldsymbol{\theta}$:

$$\boldsymbol{G} \stackrel{\text{def}}{=} \nabla_{\boldsymbol{\theta}} \text{KL}(p_{\boldsymbol{\theta}} \mid\mid q) = \mathbb{E}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y})}{q(\boldsymbol{Y})} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{Y})\right]. \tag{12}$$

Therefore, the Monte Carlo estimator of this gradient is

$$\boldsymbol{\delta}_{\text{MC}} = \frac{1}{M} \sum_{m=1}^{M} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y}^{(m)})}{q(\boldsymbol{Y}^{(m)})} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{Y}^{(m)}). \tag{13}$$

Now, we derive the Rao–Blackwellized Monte Carlo estimator of the gradient. We start with restating Theorem 2.2 in Malagutti et al. [24], which will prove useful.

Theorem 3 (Malagutti et al. [24]; Theorem 2.2)**.** *Let p and q be language models over Σ. The following equality holds*

$$\text{KL}(p \mid\mid q) = \sum_{\boldsymbol{y} \in \Sigma^*} \vec{p}(\boldsymbol{y}) \text{KL}(\vec{p}(\cdot \mid \boldsymbol{y}) \mid\mid \vec{q}(\cdot \mid \boldsymbol{y})), \tag{14}$$

where we treat $\vec{p}(\cdot \mid \boldsymbol{y})$ and $\vec{q}(\cdot \mid \boldsymbol{y})$ as probability distributions over $\overline{\Sigma}^*$.

We refer the reader to Malagutti et al. [24] for the proof. Next, to derive the Rao–Blackwellized estimator of the gradient, we take the gradient of the local KL as stated in the following theorem.

Theorem 4. *Let $p_{\boldsymbol{\theta}}$ and q be two language models over Σ and $\vec{p}_{\boldsymbol{\theta}}$ the prefix probability function of $p_{\boldsymbol{\theta}}$. Then, the following equality holds*

$$\nabla_{\boldsymbol{\theta}} \text{KL}(p_{\boldsymbol{\theta}} \mid\mid q) = \sum_{\boldsymbol{y} \in \Sigma^*} \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \mathop{\mathbb{E}}_{Y} \Big[\log \frac{\vec{p}_{\boldsymbol{\theta}}(Y \mid \boldsymbol{y})}{\vec{q}(Y \mid \boldsymbol{y})} \cdot (\nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(Y \mid \boldsymbol{y}) + \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y})) \Big]. \tag{15}$$

Proof. See App. D. ■

We then construct the Monte Carlo estimator of the gradient using the above theorem, which naturally results in the following unbiased, Rao–Blackwellized Monte Carlo estimator of the gradient:

$$\boldsymbol{\delta}_{\text{RB}} = \frac{1}{M} \sum_{m=1}^{M} \sum_{n=1}^{|\boldsymbol{Y}^{(m)}|} \mathop{\mathbb{E}}_{Y} \Big[\log \frac{\vec{p}_{\boldsymbol{\theta}}(Y \mid \boldsymbol{Y}_{<n}^{(m)})}{\vec{q}(Y \mid \boldsymbol{Y}_{<n}^{(m)})} \cdot \Big(\nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{Y}_{<n}^{(m)}) + \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(Y \mid \boldsymbol{Y}_{<n}^{(m)}) \Big) \Big]. \tag{16}$$

The above estimator is unbiased, as it is the MC estimator of Eq. (15).

Theorem 5. *Assuming* $\mathrm{Var}[\boldsymbol{\delta}_{\mathrm{RB}}] < \infty, \mathrm{Var}[\boldsymbol{\delta}_{\mathrm{MC}}] < \infty$*, we have*

$$\mathbb{E}\left[\|\boldsymbol{\delta}_{\mathrm{RB}} - \boldsymbol{G}\|^2\right] \leq \mathbb{E}\left[\|\boldsymbol{\delta}_{\mathrm{MC}} - \boldsymbol{G}\|^2\right]. \tag{17}$$

Proof. See App. D.1. ■

Off-policy Gradient. So far, we have discussed how to estimate the gradient of $\mathrm{KL}(p_{\boldsymbol{\theta}} \,||\, q)$ using samples drawn from the current policy $p_{\boldsymbol{\theta}}$. Crucially, we derive the gradient manually rather than relying on automatic differentiation because the samples depend on $\boldsymbol{\theta}$ through $p_{\boldsymbol{\theta}}$. However, in practice and for efficiency reasons, we often collect large batches of samples in parallel with the optimization loop. As a result, these samples are generated from a slightly outdated version of the policy, denoted $p_{\boldsymbol{\theta}_{\mathrm{old}}}$. To compute the KL divergence using samples from $p_{\boldsymbol{\theta}_{\mathrm{old}}}$, we first write the KL as the expectation under $p_{\boldsymbol{\theta}_{\mathrm{old}}}$ as

$$\mathrm{KL}(p_{\boldsymbol{\theta}} \,||\, q) = \mathop{\mathbb{E}}_{\boldsymbol{Y}\sim p_{\boldsymbol{\theta}_{\mathrm{old}}}}\left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y})}{p_{\boldsymbol{\theta}_{\mathrm{old}}}(\boldsymbol{Y})}\log\frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y})}{q(\boldsymbol{Y})}\right]. \tag{18}$$

Therefore, the MC estimator using samples $\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(M)} \overset{\text{i.i.d.}}{\sim} p_{\boldsymbol{\theta}_{\mathrm{old}}}$ is

$$\mu_{\mathrm{MC}}^{\mathrm{old}} = \frac{1}{M}\sum_{m=1}^{M}\frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y}^{(m)})}{p_{\boldsymbol{\theta}_{\mathrm{old}}}(\boldsymbol{Y}^{(m)})}\log\frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y}^{(m)})}{q(\boldsymbol{Y}^{(m)})}. \tag{19}$$

Given the unbiasedness proof of the Rao–Blackwellization in Thm. 2, we can similarly write

$$\mathrm{KL}(p_{\boldsymbol{\theta}} \,||\, q) = \mathop{\mathbb{E}}_{\boldsymbol{Y}\sim p_{\boldsymbol{\theta}_{\mathrm{old}}}}\left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y})}{p_{\boldsymbol{\theta}_{\mathrm{old}}}(\boldsymbol{Y})}\sum_{n=1}^{|\boldsymbol{Y}^{(m)}|}\mathop{\mathbb{E}}_{Y}\left[\log\frac{\vec{p}_{\boldsymbol{\theta}}(Y \mid \boldsymbol{Y}_{<n})}{\vec{q}(Y \mid \boldsymbol{Y}_{<n})}\right]\right]. \tag{20}$$

Therefore, the Rao–Blackwellized MC estimator using samples $\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(M)} \overset{\text{i.i.d.}}{\sim} p_{\boldsymbol{\theta}_{\mathrm{old}}}$ is

$$\mu_{\mathrm{RB}}^{\mathrm{old}} = \frac{1}{M}\sum_{m=1}^{M}\frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y}^{(m)})}{p_{\boldsymbol{\theta}_{\mathrm{old}}}(\boldsymbol{Y}^{(m)})}\sum_{n=1}^{|\boldsymbol{Y}^{(m)}|}\mathop{\mathbb{E}}_{Y}\left[\log\frac{\vec{p}_{\boldsymbol{\theta}}(Y \mid \boldsymbol{Y}^{(m)}_{<n})}{\vec{q}(Y \mid \boldsymbol{Y}^{(m)}_{<n})}\right]. \tag{21}$$

Since $\mu_{\mathrm{MC}}^{\mathrm{old}}$ and $\mu_{\mathrm{RB}}^{\mathrm{old}}$ use samples from the old policy that does not depend on $\boldsymbol{\theta}$, we can apply automatic differentiation to compute the estimate of the KL gradient by computing the gradient of $\mu_{\mathrm{MC}}^{\mathrm{old}}$ and $\mu_{\mathrm{RB}}^{\mathrm{old}}$.

5 Experiments

We use the sentiment control task as the testbed to empirically evaluate our theoretical findings on the KL estimators. Concretely, the reference model, denoted as q, is the `GPT-IMDB`[14] model, i.e., a `GPT-2` [28] model fine-tuned on IMDB corpus [23]. The goal of the task is to fine-tune this language model such that the samples from it are movie reviews with a positive sentiment. The fine-tuned language model is denoted with $p_{\boldsymbol{\theta}}$. In the following experiments, we estimate the KL divergence between $p_{\boldsymbol{\theta}}$ and q. We provide a code snippet for implementing the RB estimator in App. F.1.

5.1 Analyzing the KL Estimators

In this experiment, we empirically evaluate the bias, variance, and consistency of various KL estimators. To obtain $p_{\boldsymbol{\theta}}$, we fine-tune q with direct preference optimization [DPO; 29] on a sample of 5,000 data points from the IMDB training set. To create the preference data required for DPO training, following Rafailov et al. [29], we sample 4 responses for each prompt and create 6 pairs per prompt.

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

To determine the preferred response in each pair, we employ a binary sentiment classifier,[15] selecting the response with the higher probability of positive sentiment. Upon successful fine-tuning, $p_{\boldsymbol{\theta}}$ should assign a higher probability mass to movie reviews with positive sentiment while maintaining a low KL divergence with q. We then evaluate this KL divergence using our estimators to assess their reliability in measuring distributional shifts induced by fine-tuning.

Table 1: Estimated value $\pm$ empirical standard deviation of different estimators. When aggregating over prompts, μ_{HT} and μ_{CV} fail to significantly reduce the variance of μ_{MC}. RB estimator, however, achieves the lowest standard deviation.

	$M=1$	$M=5$	$M=10$
μ_{MC}	6.76 ± 0.16	6.76 ± 0.07	6.76 ± 0.05
μ_{HT}	6.76 ± 0.16	6.76 ± 0.07	6.76 ± 0.05
μ_{CV1}	6.28 ± 2.54	6.28 ± 1.13	6.28 ± 0.79
μ_{CV}	6.76 ± 0.16	6.76 ± 0.07	6.76 ± 0.05
μ_{RB}	$6.76 \pm \mathbf{0.11}$	$6.76 \pm \mathbf{0.05}$	$6.76 \pm \mathbf{0.03}$

We evaluate on 512 examples from the IMDB dataset. For each review, we randomly select a prefix length between 2 and 8 tokens and use it as the prompt. we then generate 4000 samples from $p_{\boldsymbol{\theta}}$ for each prompt. Using these samples, we compute the MC, control variate (CV), and Rao–Blackwellized (RB) estimators and estimate their standard We also implement the Horvitz–Thompson (HT) estimator; see App. A for details. The CV estimator, μ_{CV}, is computed twice: once using the optimal α estimated from 1000 samples, and once with $\alpha = 1$ to match the setup in Schulman [33].

In Tab. 1, we report the expected KL estimate along with the empirical standard deviation of different estimators evaluated at sample sizes $M = 1, 5, 10$. To obtain these estimates, we compute each estimator using M samples, repeating the process $4000/M$ times to estimate both the expected value and the standard deviation of the estimates. Our findings confirm that all estimators except one ($\mu_{\text{CV}}, \alpha = 1$), are unbiased and report an expected KL divergence of 6.76. We also observe that the CV estimators fail to significantly reduce the variance of the standard MC estimator. Importantly, the RB estimator achieves the lowest standard deviation and offers a more robust estimate compared to the MC estimator. Interestingly, we observe that the μ_{CV} estimator exhibits a noticeable bias and high standard deviation when $\alpha = 1$, i.e., when it is not set to its optimal value. The bias arises from numerical instability during the computation of $g(\boldsymbol{Y}) = \frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})} - 1$. The high variance is due to large values of $\text{Var}[g(\boldsymbol{Y})]$. Specifically, for certain prompts, $\text{Var}[g(\boldsymbol{Y})]$ can be unbounded. We visualize the estimates for 3 example prompts in App. E.

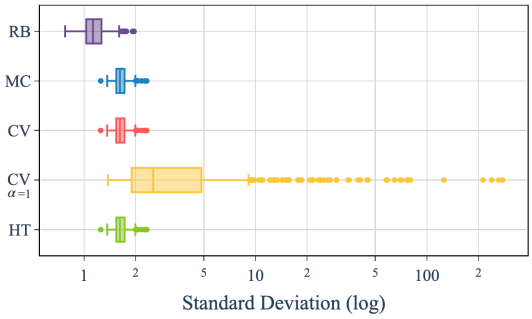

Figure 1: Standard deviation of KL estimators across various prompts in the IMDB datasest.

Since the robustness of the estimators depends on the choice of the prompt, we further analyze their estimated standard deviations across all prompts. Fig. 1 presents a box plot of standard deviations (in log scale) for each estimator. The μ_{CV} estimator with $\alpha = 1$ shows significant instability for certain prompts, with numerous outliers indicating high variance. In contrast, the μ_{MC} and the standard μ_{CV} estimators exhibit comparable standard deviations. In particular, the Rao-Blackwellized estimator consistently achieves the lowest standard deviation, suggesting that it provides the most stable estimates.

5.2 KL Estimation and RLHF Training Dynamics

A key application of KL estimation is in the RLHF training loop. From the previous experiment, we observed that the RB estimator significantly reduces the standard deviation of the MC estimator. Therefore, it is natural to ask how this affects RLHF performance when this estimator is used in the training loop. The RLHF objective consists of two terms: (i) the expected rewards for samples generated by the language model $p_{\boldsymbol{\theta}}$, which in this case is the samples' score under a sentiment classifier[16], and (ii) the negative KL divergence between the fine-tuned model $p_{\boldsymbol{\theta}}$ and the reference model q, which represents the language model before fine-tuning.

[15] Specifically, we use `https://huggingface.co/lvwerra/distilbert-imdb`.
[16] Specifically, we look at the logits of the positive class.

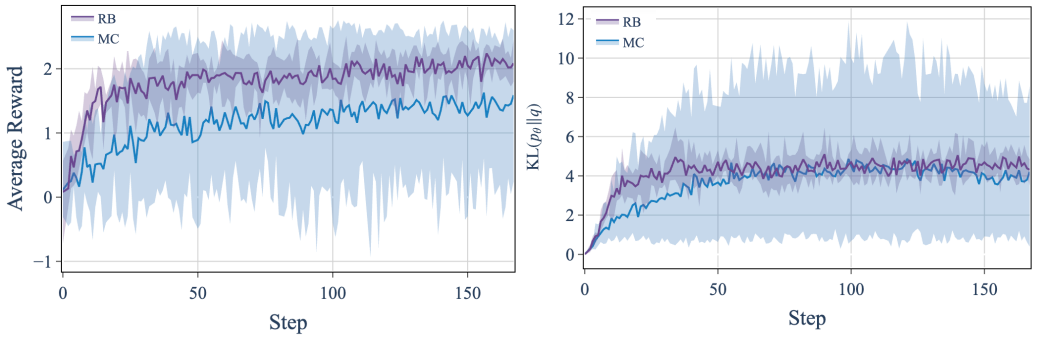

Figure 2: Comparison of the Monte Carlo (MC) and Rao–Blackwellized (RB) estimators in the RLHF fine-tuning loop. We perform RLHF with each estimator 5 times and plot the mean and standard deviation (in shades) of the average reward values and the KL at each fine-tuning step. We observe that the MC estimator is not as stable as the RB estimator and its performance varies significantly across different runs. However, RB estimator reliably offers a good balance between achieving low KL and high reward values in all runs.

We compare the MC and RB estimators for computing the gradient of the KL divergence term in the RLHF training loop. We use the RL algorithm[17] proposed by Ahmadian et al. [2],[18] which is an improved verfaiion of the REINFORCE algorithm [41].[19]

First, we empirically test Thm. 5 by measuring the variance of the gradient norm. We sample 40 prompts and compute the gradient of the KL divergence with respect to the model parameters using both the MC and RB estimators. To estimate variance empirically, we repeat this process 5 times. Both estimators are evaluated on the same prompts and model initializations to ensure a fair comparison. We find that the variance of the gradient norm estimated with the MC estimator is 59.90, whereas with the RB estimator it is 45.44. This corresponds to a 24.6% reduction in variance, providing direct empirical evidence consistent with our theoretical motivation.

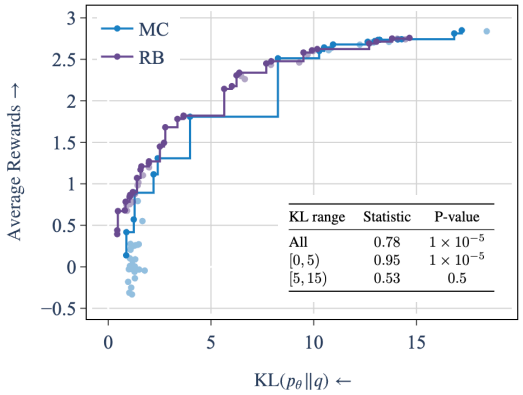

Figure 3: Compared to models trained with MC esimator, models trained with RB appear on the Pareto front 78% of the time.

We then proceed with using both estimators in RL fine-tuning. We track two metrics throughout fine-tuning: (i) the average reward associated with samples drawn from $p_{\boldsymbol{\theta}}$, and (ii) the KL divergence between $p_{\boldsymbol{\theta}}$ and q. The results are visualized in Fig. 2. The purple trace represents the training run where the μ_{RB} is used in the optimization loop to estimate the gradient of the KL divergence, while the blue trace represents the run using μ_{MC}. The x-axis denotes the fine-tuning step, with the left plot showing the evolution of the average reward and the right plot displaying the KL divergence between $p_{\boldsymbol{\theta}}$ and q over the course of fine-tuning. We repeat each experiment 5 times and report the mean and standard deviation of each metric. Notably, the KL values in the right plot are estimated using the RB estimator. However, we observe the same overall trend when using the MC estimator for evaluation.

As illustrated in Fig. 2, the performance of the models trained using the MC estimator varies significantly across the 5 experiments, resulting in a large standard deviation in both average rewards and the KL divergence. However, RB estimator consistantly achieves high rewards and reasonable KL values across all runs. This observation suggests that the RB estimator makes RLHF runs more stable.

[17] App. D.2 discusses a common mistake when Rao–Blackwellizing the KL estimator in trust-region algorithms.
[18] Specifically, we use the available implementation of this algorithm in the `trl` library [40].
[19] App. F reports the hyperparameters used for the algorithm.

Finally, we vary the KL coefficient, β, in $[0.01, 0.1]$ range and fine-tune 18 models with each estimator. For each estimator, we plot the Pareto frontier of average rewards versus KL divergence in Fig. 3, displaying models that do not appear on the Pareto front with reduced opacity. Overall, we find that fine-tuning with the RB estimator is more effective at achieving high rewards while maintaining a low KL divergence from the reference model. To quantify this effect, we compute the fraction of RB fine-tuned models that appear on the overall Pareto front—i.e., the frontier obtained when considering all models fine-tuned with either estimator. We then conduct a permutation test and report the results in Fig. 3. We find that 78% of the points on the overall Pareto front come from RB fine-tuned models. Restricting to models with KL values below 5, this fraction rises to 95%, with both results being statistically significant.

6 Conclusion

In this paper, we study the problem of estimating the KL divergence between language models. We provide a comprehensive formal analysis of various KL estimators, with a focus on their bias and variance. We introduce the RB estimator, which is provably unbiased and has variance at most equal to that of the standard NC estimator. This estimator applies the well-known Rao–Blackwellization technique to reduce the variance of the standard MC method. Our empirical results show that the RB estimator significantly reduces the variance compared to the MC estimator, while other estimators fail to achieve meaningful variance reduction or, in some cases, suffer from unbounded variance. Additionally, we find that using our proposed RB estimator makes RLHF more stable and produces models that more frequently lie on the Pareto frontier of reward versus KL, compared to models fine-tuned with the MC estimator.

Impact Statement

In this paper, we investigate the fundamental problem of estimating KL divergence between language models. One key application of KL estimation is in RLHF, which aims to enhance fluency while aligning language models with user preferences. However, RLHF can also be misused by bad actors to optimize models for generating misleading, biased, or harmful content. While our work provides a deeper understanding of KL estimation techniques, it is purely foundational research and does not introduce new risks or directly contribute to harmful applications.

Limitations

In our RLHF experiments, evaluating the variance of our estimator and comparing it to existing methods requires training a large number of models. For instance, the significance test in §5.2 involves training 36 models. Due to limited computational resources, we used the controlled-generation task as a testbed. Given the strength of both our theoretical and empirical results, we hope future work will adopt the Rao–Blackwellized estimator and apply it to larger language models and a wider variety of RL-based approaches to LM alignment.

Acknowledgements

We thank Ahmad Beirami and Cristina Pinneri for the insightful discussions throughout the course of this project. We also thank Alexander K. Lew for the valuable feedback on a draft of this paper. Afra Amini is supported by the ETH AI Center doctoral fellowship.

References

[1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The International Conference on Learning Representations*.

[2] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in llms.

[3] David Blackwell. 1947. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, 18(1):105–110.

[4] Nadav Borenstein, Anej Svete, Robin Chan, Josef Valvoda, Franz Nowak, Isabelle Augenstein, Eleanor Chodroff, and Ryan Cotterell. 2024. What languages are easy to language-model? a perspective from learning probabilistic regular languages. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[5] George Casella and Christian P. Robert. 1996. Rao–Blackwellisation of sampling schemes. *Biometrika*.

[6] Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*.

[7] Imre Csiszár. 1967. On information-type measure of difference of probability distributions and indirect observations. *Studia Scientiarum Mathematicarum Hungarica*.

[8] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. UltraFeedback: Boosting language models with scaled AI feedback. In *Proceedings of the International Conference on Machine Learning*.

[9] Kevin Du, Vésteinn Snæbjarnarson, Niklas Stoehr, Jennifer White, Aaron Schein, and Ryan Cotterell. 2024. Context versus prior knowledge in language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[10] Tomas Geffner and Justin Domke. 2018. Using large ensembles of control variates for variational inference. In *Advances in Neural Information Processing Systems*.

[11] Alan E. Gelfand and Adrian F. M. Smith. 1990. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

[12] Alison L. Gibbs and Francis Edward Su. 2002. On choosing and bounding probability metrics. *International Statistical Review / Revue Internationale de Statistique*.

[13] Alexander Havrilla, Maksym Zhuravinskyi, Duy Phung, Aman Tiwari, Jonathan Tow, Stella Biderman, Quentin Anthony, and Louis Castricato. 2023. trlX: A framework for large scale reinforcement learning from human feedback. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

[14] John R. Hershey and Peder A. Olsen. 2007. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

[15] Hendrik Jan Hoogeboom. 2015. Undecidable problems for context-free grammars.

[16] D. G. Horvitz and D. J. Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*.

[17] Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Weixun Wang, Dehao Zhang, and Yu Cao. 2024. OpenRLHF: An easy-to-use, scalable and high-performance RLHF framework. *Computing Research Repository*.

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

[19] S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*.

[20] Daniel J. Lehmann. 1977. Algebraic structures for transitive closure. *Theoretical Computer Science*.

[21] E. L. Lehmann and George Casella. 1998. *Theory of Point Estimation*.

[22] Zhifei Li and Jason Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

[23] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

[24] Luca Malagutti, Andrius Buinovskij, Anej Svete, Clara Meister, Afra Amini, and Ryan Cotterell. 2024. The role of n-gram smoothing in the age of neural networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.

[25] Meta. 2023. Llama 2: Open foundation and fine-tuned chat models. Technical report, Meta.

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

[27] Pouya Pezeshkpour. 2023. Measuring and modifying factual knowledge in large language models. *Computing Research Repository*, arXiv:2306.06264.

[28] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

[29] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*.

[30] C. R. Rao. 1945. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37:81–91.

[31] Christian P. Robert and George Casella. 2004. *Monte Carlo Statistical Methods*.

[32] Sheldon M. Ross. 2002. *Simulation*.

[33] John Schulman. 2020. Approximating KL divergence. `http://joschu.net/blog/kl-approx.html`.

[34] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. HybridFlow: A flexible and efficient RLHF framework. *Computing Research Repository*, arXiv:2409.19256.

[35] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*.

[36] Niklas Stoehr, Kevin Du, Vésteinn Snæbjarnarson, Robert West, Ryan Cotterell, and Aaron Schein. 2024. Activation scaling for steering and interpreting language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

[37] Anej Svete, Nadav Borenstein, Mike Zhou, Isabelle Augenstein, and Ryan Cotterell. 2024. Can transformers learn n-gram language models? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

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

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

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

[41] R. J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*.

A Horvitz–Thompson Estimation

When estimating the KL divergence between language models p and q, we have access not only to samples from p but also to the probability assigned by p to any string $\boldsymbol{y}$. This enables the use of a more informed estimator, which leverages these probabilities during its construction. Notably, this estimator is a specific instance of the **Horvitz–Thompson (HT)** estimator [16], defined as

$$\mu_{\text{HT}} = \sum_{\boldsymbol{y} \in S} \frac{p(\boldsymbol{y})}{\pi_S(\boldsymbol{y})} \log \frac{p(\boldsymbol{y})}{q(\boldsymbol{y})} = \sum_{\boldsymbol{y} \in S} \frac{p(\boldsymbol{y})}{\pi_S(\boldsymbol{y})} f(\boldsymbol{y}), \tag{22}$$

where S is the random variable representing the *set* of all sampled strings. Any sampling design can be specified to generate the elements of S. The **inclusion probability**, denoted by $\pi_S(\boldsymbol{y})$, is the probability that a particular string $\boldsymbol{y}$ is included in S, or equivalently, $\mathbb{E}\left[\mathbb{1}\{\boldsymbol{y} \in S\}\right]$.

Proposition 6. *μ_{HT} is an unbiased estimator of the KL divergence, i.e., $\mathbb{E}\left[\mu_{\text{HT}}\right] = \text{KL}(p \mid\mid q)$.*

Proof. The bias of the estimator is as follows:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{S}\left[\mu_{\text{HT}}\right] - \text{KL}(p \mid\mid q) \\
&= \mathop{\mathbb{E}}_{S}\Big[\sum_{\boldsymbol{y} \in S} \frac{p(\boldsymbol{y})}{\pi_S(\boldsymbol{y})} \log \frac{p(\boldsymbol{y})}{q(\boldsymbol{y})}\Big] - \text{KL}(p \mid\mid q) && \text{(definition of } \mu_{\text{HT}}\text{)} && \text{(23a)} \\
&= \mathop{\mathbb{E}}_{S}\Big[\sum_{\boldsymbol{y} \in \Sigma^*} \frac{p(\boldsymbol{y})}{\pi_S(\boldsymbol{y})} \log \frac{p(\boldsymbol{y})}{q(\boldsymbol{y})} \cdot \mathbb{1}\{\boldsymbol{y} \in S\}\Big] - \text{KL}(p \mid\mid q) && && \text{(23b)} \\
&= \sum_{\boldsymbol{y} \in \Sigma^*} \frac{p(\boldsymbol{y})}{\pi_S(\boldsymbol{y})} \log \frac{p(\boldsymbol{y})}{q(\boldsymbol{y})} \cdot \mathop{\mathbb{E}}_{S}\Big[\mathbb{1}\{\boldsymbol{y} \in S\}\Big] - \text{KL}(p \mid\mid q) && \text{(linearity of expectation)} && \text{(23c)} \\
&= \sum_{\boldsymbol{y} \in \Sigma^*} \frac{p(\boldsymbol{y})}{\cancel{\pi_S(\boldsymbol{y})}} \log \frac{p(\boldsymbol{y})}{q(\boldsymbol{y})} \cdot \cancel{\pi_S(\boldsymbol{y})} - \text{KL}(p \mid\mid q) && \text{(definition of } \pi_S(\boldsymbol{y})\text{)} && \text{(23d)} \\
&= 0. && && \text{(23e)}
\end{aligned}$$

■

Similar to the MC estimator, the HT estimator does not necessarily return a non-negative estimate of the KL. In principle, however, we should prefer Eq. (22) to Eq. (4) because it exploits more information—namely, the knowledge of p. Whether the HT estimator yields lower variance than the MC estimator depends on the sampling design used to construct S. In our experiments in App. E, we used the sampling-with-replacement design, where $\pi_S(\boldsymbol{y}) = 1 - (1 - p(\boldsymbol{y}))^M$. Compared to the MC estimator, we observed no significant reduction in variance in our experiments.

B Control Variate Monte Carlo Estimation

Proposition 1. *Consider the control variate MC estimator μ_{CV} defined in Eq. (5), and assume that $\mathbb{E}[g(\boldsymbol{Y})] < \infty$. Then μ_{CV} is an unbiased estimator, and its variance is given by*

$$\text{Var}[\mu_{\text{CV}}] = \frac{\text{Var}[f] + \alpha^2 \,\text{Var}[g] + 2\alpha \,\text{Cov}[f, g]}{M}. \tag{6}$$

Proof. Recall that the control variate Monte Carlo estimator is defined as

$$\mu_{\text{CV}} = \frac{1}{M} \sum_{m=1}^{M} f(\boldsymbol{Y}^{(m)}) + \alpha \cdot (g(\boldsymbol{Y}^{(m)}) - G). \tag{5}$$

Note that $\{\boldsymbol{Y}^{(m)}\}_{m=1}^{M} \overset{\text{i.i.d.}}{\sim} p$. First, we look at the expected value of the estimator:

$$\mathbb{E}[\mu_{\text{CV}}] = \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}\Big[f(\boldsymbol{Y}^{(m)})\Big] + \alpha \cdot \underbrace{(\mathbb{E}\Big[g(\boldsymbol{Y}^{(m)})\Big] - G)}_{=0} = \mathbb{E}[f] \tag{24a}$$

Therefore, it is unbiased. Next, we manipulate the variance

$$\text{Var}[\mu_{\text{CV}}] = \frac{1}{M} \text{Var}[f + \alpha \cdot (g - G)] = \frac{1}{M} \text{Var}[f] + \frac{\alpha^2}{M} \text{Var}[g] + \frac{2\alpha}{M} \text{Cov}[f, g], \tag{24b}$$

where the second equality above stems from well-known variance identities.[20] ■

Proposition 7. *Provided that the target KL divergence is finite, the estimator μ_{CV} with $\alpha = 1$ is nonnegative with probability one. Furthermore, $\alpha = 1$ is the only value for which we can guarantee nonnegativity of μ_{CV}.*

Proof. To prove that μ_{CV} is nonnegative with probability one when $\alpha = 1$, we will prove that each term $\log \frac{p(\boldsymbol{Y}^{(m)})}{q(\boldsymbol{Y}^{(m)})} + \alpha \cdot \left(\frac{q(\boldsymbol{Y}^{(m)})}{p(\boldsymbol{Y}^{(m)})} - 1\right)$ in the summation that defines μ_{CV} is nonnegative.

Let $r = \frac{q(\boldsymbol{Y}^{(m)})}{p(\boldsymbol{Y}^{(m)})}$. We identify α for which $\log \frac{1}{r} + \alpha \cdot (r - 1) \geq 0$ holds for all $r > 0$.[21] Equivalently, we want to find α values for which:

$$\inf_{r>0} -\log r + \alpha \cdot (r - 1) \geq 0. \tag{25a}$$

Next, we prove that $\alpha = 1$ is the only value satisfies the inequality Eq. (25a). To see why, consider the first-order optimality conditions for the minimization over r. These conditions are necessary and sufficient as $-\log r + \alpha \cdot (r - 1)$ is convex for all α over the region $r > 0$. Solving for r such that the derivative is zero gives us

$$0 = \frac{\partial}{\partial r} [-\log r + \alpha \cdot (r - 1)] \tag{25b}$$

$$\iff 0 = -1/r + \alpha \tag{25c}$$

$$\iff r = 1/\alpha. \tag{25d}$$

Plugging that value allows us to simplify away the minimization:

$$0 \leq \inf_{r>0} -\log r + \alpha \cdot (r - 1) \tag{25e}$$

$$= -\log(1/\alpha) + \alpha \cdot (1/\alpha - 1) \tag{25f}$$

$$= \log(\alpha) + 1 - \alpha. \tag{25g}$$

For each term in μ_{CV} to be nonnegative, we need $\log(\alpha) + 1 - \alpha \geq 0$. However, the function $\log(\alpha) + 1 - \alpha$ is strictly concave and nonpositive for all $\alpha > 0$, attaining zero only at $\alpha = 1$. Thus, when $\alpha = 1$ every term in μ_{CV} is nonnegative. When $\alpha \neq 1$, some terms may be nonnegative, leading to the possibility that μ_{CV} may be negative.

■

[20] Let X and Y be real-valued random variables, and let a be a real scalar. Then, the following hold: $\text{Var}[X + Y] = \text{Var}[X] + \text{Var}[Y] + 2\,\text{Cov}[X, Y]$, $\text{Var}[aX] = a^2 \,\text{Var}[X]$, and $\text{Var}[X + a] = \text{Var}[X]$.

[21] Note that $r > 0$ whenever this term is finite.

Proposition 8. *Given the random variable $\boldsymbol{Y} \sim p$, let $f(\boldsymbol{Y}) = \log \frac{p(\boldsymbol{Y})}{q(\boldsymbol{Y})}$ and $g(\boldsymbol{Y}) = \frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})}$. We prove $\mathrm{Cov}[f, g]$ is zero if and only if $p = q$.*

Proof.

$$\mathrm{Cov}[f,g] = \mathbb{E}\left[\left(\log\frac{p(\boldsymbol{Y})}{q(\boldsymbol{Y})} - \mathbb{E}\left[\log\frac{p(\boldsymbol{Y})}{q(\boldsymbol{Y})}\right]\right)\left(\frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})} - \underbrace{\mathbb{E}\left[\frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})}\right]}_{=1}\right)\right] \tag{26a}$$

$$= \mathbb{E}\left[\frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})}\log\frac{p(\boldsymbol{Y})}{q(\boldsymbol{Y})} - \cancel{\log\frac{p(\boldsymbol{Y})}{q(\boldsymbol{Y})}} - \mathbb{E}\left[\log\frac{p(\boldsymbol{Y})}{q(\boldsymbol{Y})}\right]\frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})} + \cancel{\mathbb{E}\left[\log\frac{p(\boldsymbol{Y})}{q(\boldsymbol{Y})}\right]}\right] \tag{26b}$$

$$= \mathbb{E}\left[\frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})}\log\frac{p(\boldsymbol{Y})}{q(\boldsymbol{Y})}\right] - \mathbb{E}\left[\log\frac{p(\boldsymbol{Y})}{q(\boldsymbol{Y})}\right]\underbrace{\mathbb{E}\left[\log\frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})}\right]}_{=1} \tag{26c}$$

$$= \mathbb{E}\left[-\frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})}\log\frac{q(\boldsymbol{Y})}{p(\boldsymbol{Y})}\right] - \mathbb{E}\left[\log\frac{p(\boldsymbol{Y})}{q(\boldsymbol{Y})}\right] \tag{26d}$$

$$= -\mathrm{KL}(q \,||\, p) - \mathrm{KL}(p \,||\, q). \tag{26e}$$

Since each KL divergence is non-negative and equals zero if and only if the two distributions coincide almost everywhere, we have,

$$\mathrm{Cov}[f,g] = 0 \iff -\mathrm{KL}(q \,||\, p) - \mathrm{KL}(p \,||\, q) = 0 \iff p = q. \tag{26f}$$

■

Proposition 9.

$$\mathrm{Var}\left[\mu_{\mathrm{CV}}^{(\alpha)}\right] \leq \mathrm{Var}\left[\mu_{\mathrm{MC}}\right] \iff \alpha \in [\min(0, 2\alpha^*), \max(0, 2\alpha^*)] \tag{27}$$

Proof. We first establish conditions of α for which the variance of $\mathrm{Var}[\mu_{\mathrm{CV}}]$ does not increase that of $\mathrm{Var}[\mu_{\mathrm{MC}}]$. In other words, we seek conditions on α such that the following inequality holds:

$$\mathrm{Var}\left[\mu_{\mathrm{CV}}^{(\alpha)}\right] \leq \mathrm{Var}\left[\mu_{\mathrm{MC}}\right] \tag{28}$$

$$\frac{\mathrm{Var}[f] + \alpha^2\,\mathrm{Var}[g] + 2\alpha\,\mathrm{Cov}[f,g]}{M} \leq \frac{\mathrm{Var}[f]}{M} \tag{29}$$

$$\mathrm{Var}[f] + \alpha^2\,\mathrm{Var}[g] + 2\alpha\,\mathrm{Cov}[f,g] \leq \mathrm{Var}[f] \tag{30}$$

$$\alpha^2\,\mathrm{Var}[g] + 2\alpha\,\mathrm{Cov}[f,g] \leq 0. \tag{31}$$

Observe that the function on the left-hand side is quadratic in α, and, moreover, it is *convex* because $\mathrm{Var}[g] \geq 0$. The minimum of this quadratic is $\alpha^* = -\frac{\mathrm{Cov}[f,g]}{\mathrm{Var}[g]}$. We can use the quadratic formula to identify the two values of α where it equals 0, i.e., $\alpha \in \{0, 2\alpha^*\}$. Now, because the quadratic is convex, we have that it is ≤ 0 for values of $\alpha \in [\min(0, 2\alpha^*), \max(0, 2\alpha^*)]$. Note that α^* can be positive or negative; hence the $\min$ and $\max$. ■

Remark 10. *When does Schulman's (2020) suboptimal choice of $\alpha = 1$ not hurt estimator variance? To answer this question, we substitute $\alpha = 1$ into the variance of μ_{CV} given in Eq. (6), we have*

$$\mathrm{Var}[\mu_{\mathrm{CV}}] = \frac{\mathrm{Var}[f] + \mathrm{Var}[g] + 2\,\mathrm{Cov}[f,g]}{M}. \tag{32}$$

Therefore, for $\mathrm{Var}[\mu_{\mathrm{CV}}] \leq \mathrm{Var}[\mu_{\mathrm{MC}}]$, we must have

$$\frac{\mathrm{Var}[f] + \mathrm{Var}[g] + 2\,\mathrm{Cov}[f,g]}{M} \leq \frac{\mathrm{Var}[f]}{M} \tag{33a}$$

$$\mathrm{Var}[f] + \mathrm{Var}[g] + 2\,\mathrm{Cov}[f,g] \leq \mathrm{Var}[f] \tag{33b}$$

$$\mathrm{Var}[g] + 2\,\mathrm{Cov}[f, g] \leq 0 \tag{33c}$$

$$-\frac{\mathrm{Cov}[f, g]}{\mathrm{Var}[g]} \geq \frac{1}{2} \tag{33d}$$

$$\alpha^* \geq \frac{1}{2}. \tag{33e}$$

Therefore, choosing $\alpha = 1$ does not hurt the variance if $\alpha^* \geq \frac{1}{2}$, but does otherwise.

C Rao–Blackwellized Estimator

Lemma 11. *With regard to the estimator $\mu_{\text{RB}}^{(N)}$, the following two properties hold for all $N > 0$*

1. $\mathbb{E}[\mu_{\text{RB}}^{(N)}] = \mathbb{E}[\mu_{\text{MC}}^{(N)}]$ *(unbiasedness)*
2. $\text{Var}\left[\mu_{\text{RB}}^{(N)}\right] \leq \text{Var}\left[\mu_{\text{MC}}^{(N)}\right]$ *(variance reduction)*

Proof. In the proof below, we consider the case where $M = 1$. The proof easily generalizes to $M > 1$ using the i.i.d. assumption. Additionally, we write $\mu_{\text{RB}}^{(N)}(\overline{\boldsymbol{Y}})$, rather than suppressing the argument to the estimator as we do in the main text. We begin by proving the unbiasedness property of the estimator.

$$\mathbb{E}[\mu_{\text{RB}}^{(N)}] = \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\sum_{n=1}^{N} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\mu_{\text{MC}}^{n}(\overline{\boldsymbol{Y}}) \mid \overline{\boldsymbol{Y}}_{<n} = \overline{\boldsymbol{Y}}'_{<n}\right]\right] \qquad \text{(definition of } \mu_{\text{RB}}^{(N)}\text{)} \tag{34a}$$

$$= \sum_{n=1}^{N} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\mu_{\text{MC}}^{n}(\overline{\boldsymbol{Y}}) \mid \overline{\boldsymbol{Y}}_{<n} = \overline{\boldsymbol{Y}}'_{<n}\right]\right] \qquad \text{(linearity of expectation)} \tag{34b}$$

$$= \sum_{n=1}^{N} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\mu_{\text{MC}}^{n}(\overline{\boldsymbol{Y}})\right] \qquad \text{(law of total expectation)} \tag{34c}$$

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\sum_{n=1}^{N} \mu_{\text{MC}}^{n}(\overline{\boldsymbol{Y}})\right] \qquad \text{(linearity of expectation)} \tag{34d}$$

$$= \mathbb{E}[\mu_{\text{MC}}^{(N)}] \qquad \text{(definition of } \mu_{\text{MC}}^{(N)}\text{)} \tag{34e}$$

Next, we prove the variance-reduction property:

$$\text{Var}\left[\mu_{\text{RB}}^{(N)}\right]$$

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\left(\sum_{n=1}^{N} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\mu_{\text{MC}}^{n}(\overline{\boldsymbol{Y}}) \,\middle|\, \overline{\boldsymbol{Y}}_{<n} = \overline{\boldsymbol{Y}}'_{<n}\right]\right)^2\right] - \mathbb{E}[\mu_{\text{MC}}^{(N)}]^2 \tag{35a}$$

(defintion of $\mu_{\text{RB}}^{(N)}$ and unbiasedness)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\left(\sum_{n=1}^{N} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^n}\left[\mu_{\text{MC}}^{n}(\overline{\boldsymbol{Y}}^n) \,\middle|\, \overline{\boldsymbol{Y}}^n_{<n} = \overline{\boldsymbol{Y}}'_{<n}\right]\right)^2\right] - \mathbb{E}[\mu_{\text{RB}}^{(N)}]^2 \tag{35b}$$

(each $\overline{\boldsymbol{Y}}^n$ is distributed independently and identically to $\overline{\boldsymbol{Y}}$)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\left(\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^1}\left[\cdots \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^N}\left[\sum_{n=1}^{N} \mu_{\text{MC}}^{n}(\overline{\boldsymbol{Y}}^n) \,\middle|\, \overline{\boldsymbol{Y}}^N_{<N} = \overline{\boldsymbol{Y}}'_{<N}\right] \cdots \,\middle|\, \overline{\boldsymbol{Y}}^1_{<1} = \overline{\boldsymbol{Y}}'_{<1}\right]\right)^2\right] + \mathbb{E}[\mu_{\text{MC}}^{(N)}]^2 \tag{35c}$$

(linearity of expectation)

$$\leq \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^1}\left[\cdots \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^N}\left[\left(\sum_{n=1}^{N} \mu_{\text{MC}}^{n}(\overline{\boldsymbol{Y}}^n)\right)^2 \,\middle|\, \overline{\boldsymbol{Y}}^N_{<N} = \overline{\boldsymbol{Y}}'_{<N}\right] \cdots \,\middle|\, \overline{\boldsymbol{Y}}^1_{<1} = \overline{\boldsymbol{Y}}'_{<1}\right]\right] - \mathbb{E}[\mu_{\text{MC}}^{(N)}]^2 \tag{35d}$$

(Jensen's inequality)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^1}\left[\cdots \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^N}\left[\left(\sum_{n=1}^{N} \mu_{\text{MC}}^{n}(\overline{\boldsymbol{Y}}^n)\right)^2\right]\right] - \mathbb{E}[\mu_{\text{MC}}^{(N)}]^2 \tag{35e}$$

(law of total expectation)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\left(\sum_{n=1}^{N} \mu_{\text{MC}}^{n}(\overline{\boldsymbol{Y}})\right)^2\right] - \mathbb{E}[\mu_{\text{MC}}^{(N)}]^2 \tag{35f}$$

($\{\overline{\boldsymbol{Y}}^n\}_{n=1}^N$ are i.i.d.)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\left(\mu_{\text{MC}}^{(N)}(\overline{\boldsymbol{Y}})\right)^2\right] - \mathbb{E}[\mu_{\text{MC}}^{(N)}]^2 \tag{35g}$$

(definition of $\mu_{\text{MC}}^{(N)}$)

$$= \text{Var}\left[\mu_{\text{MC}}^{(N)}\right] \tag{35h}$$

(definition of variance)

■

Lemma 12.

$$\sum_{n=1}^{\infty} \mu_{\text{MC}}^n = \mu_{\text{MC}} \tag{36}$$

Proof.

$$\mu_{\text{MC}} = \frac{1}{M}\sum_{m=1}^{M} \log \frac{p(\boldsymbol{Y}^{(m)})}{q(\boldsymbol{Y}^{(m)})} \tag{37a}$$

$$= \frac{1}{M}\sum_{m=1}^{M} \log \frac{\vec{p}(\text{EOS} \mid \boldsymbol{Y}^{(m)})}{\vec{q}(\text{EOS} \mid \boldsymbol{Y}^{(m)})} \prod_{n=1}^{|\boldsymbol{Y}^{(m)}|} \log \frac{\vec{p}(Y_n^{(m)} \mid \boldsymbol{Y}_{<n}^{(m)})}{\vec{q}(Y_n^{(m)} \mid \boldsymbol{Y}_{<n}^{(m)})} \tag{37b}$$

$$= \frac{1}{M}\sum_{m=1}^{M} \log \prod_{n=1}^{\infty} \frac{\vec{p}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})}{\vec{q}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})} \tag{37c}$$

$$= \frac{1}{M}\sum_{m=1}^{M} \log \lim_{N\to\infty} \prod_{n=1}^{N} \frac{\vec{p}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})}{\vec{q}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})} \tag{37d}$$

$$= \lim_{N\to\infty} \frac{1}{M}\sum_{m=1}^{M} \log \prod_{n=1}^{N} \frac{\vec{p}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})}{\vec{q}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})} \tag{37e}$$

$$= \lim_{N\to\infty} \frac{1}{M}\sum_{m=1}^{M} \log \frac{\vec{p}(\overline{\boldsymbol{Y}}_{\leq N}^{(m)})}{\vec{q}(\overline{\boldsymbol{Y}}_{\leq N}^{(m)})} \tag{37f}$$

$$= \lim_{N\to\infty} \sum_{n=1}^{N} \mu_{\text{MC}}^n \tag{37g}$$

Note that going from Eq. (37b) to Eq. (37c), we use the padding construction given in §3. ■

Theorem 2. *Suppose the MC estimator μ_{MC} has finite variance, i.e., $\text{Var}[\mu_{\text{MC}}] < \infty$. Then the following properties regarding μ_{RB} hold:*

(i) $\mathbb{E}[\mu_{\text{RB}}] = \text{KL}(p \,||\, q)$ *(unbiasedness)* $\quad$ (ii) $\text{Var}[\mu_{\text{RB}}] \leq \text{Var}[\mu_{\text{MC}}]$ *(variance reduction)*

Proof. In this proof, we consider the special case of $M = 1$. The proof easily generalizes to $M > 1$ using the i.i.d. assumption. Additionally, we write $\mu_{\text{RB}}(\overline{\boldsymbol{Y}})$, rather than suppressing the argument to the estimator as we do in the main text. We begin with proving the unbiasedness of the estimator, using Lemma 12 and Lemma 11.

$$\mathbb{E}\left[\mu_{\text{RB}}(\overline{\boldsymbol{Y}})\right] = \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\lim_{N\to\infty}\sum_{n=1}^{N}\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}) \mid \overline{\boldsymbol{Y}}_{<n} = \overline{\boldsymbol{Y}}'_{<n}\right]\right] \tag{38a}$$

$$= \lim_{N\to\infty}\sum_{n=1}^{N}\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}) \mid \overline{\boldsymbol{Y}}_{<n} = \overline{\boldsymbol{Y}}'_{<n}\right]\right] \tag{38b}$$

(Tonelli's Theorem)

$$= \lim_{N\to\infty} \sum_{n=1}^{N} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}} \left[\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}})\right] \tag{38c}$$

(Lemma 11)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}} \left[\lim_{N\to\infty} \sum_{n=1}^{N} \mu_{\text{MC}}^n(\overline{\boldsymbol{Y}})\right] \tag{38d}$$

(Fubini's Theorem, $\mathbb{E}[\sum_{n=1}^{\infty} |\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}})|] < \infty$)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}} \left[\mu_{\text{MC}}(\overline{\boldsymbol{Y}})\right] \tag{38e}$$

(Lemma 12)

$$= \text{KL}(p \,||\, q). \tag{38f}$$

(unbiasedness of μ_{MC})

Finally, we prove the variance-reduction property.

$$\text{Var}[\mu_{\text{RB}}] = \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'} \left[\left(\lim_{N\to\infty} \sum_{n=1}^{N} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}} \left[\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}) \mid \overline{\boldsymbol{Y}}_{<n} = \overline{\boldsymbol{Y}}'_{<n}\right] - \text{KL}(p \,||\, q)\right)^2\right] \tag{39a}$$

(Definition of $\mu_{\text{RB}}(\boldsymbol{Y})$)

$$= \lim_{N\to\infty} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'} \left[\left(\sum_{n=1}^{N} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}} \left[\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}) \mid \overline{\boldsymbol{Y}}_{<n} = \overline{\boldsymbol{Y}}'_{<n}\right] - \text{KL}(p \,||\, q)\right)^2\right] \tag{39b}$$

(dominated convergence theorem, $\text{Var}[\mu_{\text{RB}}] < \infty$)

$$\leq \lim_{N\to\infty} \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}} \left[\left(\sum_{n=1}^{N} \mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}) - \text{KL}(p, q)\right)^2\right] \tag{39c}$$

(Lemma 11, variance reduction)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}} \left[\lim_{N\to\infty} \left(\sum_{n=1}^{N} \mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}) - \text{KL}(p, q)\right)^2\right] \tag{39d}$$

(dominated convergence theorem, $\text{Var}[\mu_{\text{MC}}] < \infty$)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}} \left[\left(\mu_{\text{MC}}(\overline{\boldsymbol{Y}}) - \text{KL}(p \,||\, q)\right)^2\right] \tag{39e}$$

(Lemma 12)

$$= \text{Var}[\mu_{\text{MC}}]. \tag{39f}$$

■

D Rao–Blackwellized Estimator of the Gradient

Theorem 4. *Let $p_{\boldsymbol{\theta}}$ and q be two language models over Σ and $\vec{p}_{\boldsymbol{\theta}}$ the prefix probability function of $p_{\boldsymbol{\theta}}$. Then, the following equality holds*

$$\nabla_{\boldsymbol{\theta}} \mathrm{KL}(p_{\boldsymbol{\theta}} \mid\mid q) = \sum_{\boldsymbol{y} \in \Sigma^*} \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \mathop{\mathbb{E}}_{Y} \Big[\log \frac{\vec{p}_{\boldsymbol{\theta}}(Y \mid \boldsymbol{y})}{\vec{q}(Y \mid \boldsymbol{y})} \cdot (\nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(Y \mid \boldsymbol{y}) + \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y})) \Big]. \tag{15}$$

Proof.

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}} \mathrm{KL}(p \mid\mid q) \\
&= \sum_{\boldsymbol{y} \in \Sigma^*} \mathrm{KL}(\vec{p}_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{y}) \mid\mid \vec{q}(\cdot \mid \boldsymbol{y})) \nabla_{\boldsymbol{\theta}} \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) + \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \nabla_{\boldsymbol{\theta}} \mathrm{KL}(\vec{p}_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{y}) \mid\mid \vec{q}(\cdot \mid \boldsymbol{y})) && \text{(40a)} \\
&= \sum_{\boldsymbol{y} \in \Sigma^*} \mathrm{KL}(\vec{p}_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{y}) \mid\mid \vec{q}(\cdot \mid \boldsymbol{y})) \nabla_{\boldsymbol{\theta}} \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) + \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \sum_{\overline{y} \in \overline{\Sigma}} \nabla_{\boldsymbol{\theta}} \vec{p}_{\boldsymbol{\theta}}(\overline{y} \mid \boldsymbol{y}) \log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{y} \mid \boldsymbol{y})}{\vec{q}(\overline{y} \mid \boldsymbol{y})} && \text{(40b)} \\
&= \sum_{\boldsymbol{y} \in \Sigma^*} \mathrm{KL}(\vec{p}_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{y}) \mid\mid \vec{q}(\cdot \mid \boldsymbol{y})) \nabla_{\boldsymbol{\theta}} \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) + \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \sum_{\overline{y} \in \overline{\Sigma}} \log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{y} \mid \boldsymbol{y})}{\vec{q}(\overline{y} \mid \boldsymbol{y})} \nabla_{\boldsymbol{\theta}} \vec{p}_{\boldsymbol{\theta}}(\overline{y} \mid \boldsymbol{y}) + \nabla_{\boldsymbol{\theta}} \vec{p}_{\boldsymbol{\theta}}(\overline{y} \mid \boldsymbol{y}) && \text{(40c)} \\
&= \sum_{\boldsymbol{y} \in \Sigma^*} \mathrm{KL}(\vec{p}_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{y}) \mid\mid \vec{q}(\cdot \mid \boldsymbol{y})) \nabla_{\boldsymbol{\theta}} \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) + \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \mathop{\mathbb{E}}_{\overline{Y}} \left[\left(\log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y})}{\vec{q}(\overline{Y} \mid \boldsymbol{y})} + 1 \right) \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y}) \right] && \text{(40d)} \\
&= \sum_{\boldsymbol{y} \in \Sigma^*} \mathrm{KL}(\vec{p}_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{y}) \mid\mid \vec{q}(\cdot \mid \boldsymbol{y})) \nabla_{\boldsymbol{\theta}} \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) + \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \mathop{\mathbb{E}}_{\overline{Y}} \left[\log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y})}{\vec{q}(\overline{Y} \mid \boldsymbol{y})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y}) \right] && \text{(40e)} \\
&= \sum_{\boldsymbol{y} \in \Sigma^*} \mathop{\mathbb{E}}_{\overline{Y}} \left[\log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y})}{\vec{q}(\overline{Y} \mid \boldsymbol{y})} \right] \nabla_{\boldsymbol{\theta}} \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) + \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \mathop{\mathbb{E}}_{\overline{Y}} \left[\log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y})}{\vec{q}(\overline{Y} \mid \boldsymbol{y})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y}) \right] && \text{(40f)} \\
&= \sum_{\boldsymbol{y} \in \Sigma^*} \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \mathop{\mathbb{E}}_{\overline{Y}} \left[\log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y})}{\vec{q}(\overline{Y} \mid \boldsymbol{y})} \right] \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) + \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \mathop{\mathbb{E}}_{\overline{Y}} \left[\log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y})}{\vec{q}(\overline{Y} \mid \boldsymbol{y})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y}) \right] && \text{(40g)} \\
&= \sum_{\boldsymbol{y} \in \Sigma^*} \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \mathop{\mathbb{E}}_{\overline{Y}} \left[\log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y})}{\vec{q}(\overline{Y} \mid \boldsymbol{y})} \left(\nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{Y} \mid \boldsymbol{y}) + \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{y}) \right) \right]. && \text{(40h)}
\end{aligned}$$

■

D.1 Variance-Reduction Proof

The proof structure is as follows: We first prove that the inequality holds when we constrain $\overline{\boldsymbol{Y}}$ to have length less than or equal to N. We then generalize to the infinite-length sequences by analyzing as $N \to \infty$. We begin with defining the truncated MC and RB estimators. Let $\boldsymbol{\delta}_{\text{MC}}^{(N)}$ be the truncated MC estimator of the gradient:

$$\boldsymbol{\delta}_{\text{MC}}^{(N)} = \sum_{n=1}^{N} \frac{1}{M} \sum_{m=1}^{M} \log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})}{\vec{q}(\overline{Y}_n^{(m)} \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq N}^{(m)}). \tag{41}$$

Let $\boldsymbol{\delta}_{\text{RB}}^{(N)}$ be the truncated RB estimator:

$$\boldsymbol{\delta}_{\text{RB}}^{(N)} = \frac{1}{M} \sum_{m=1}^{M} \sum_{n=1}^{N} \mathop{\mathbb{E}}_{\overline{Y}_n} \left[\log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})}{\vec{q}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})} \left(\nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n}^{(m)}) + \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{<n}^{(m)}) \right) \right]. \tag{42}$$

Lemma 13. *The truncated MC estimator of the gradient $\boldsymbol{\delta}_{\text{MC}}^{(N)}$ converges to $\boldsymbol{\delta}_{\text{MC}}$ as N goes to ∞, i.e., $\lim_{N\to\infty} \boldsymbol{\delta}_{\text{MC}}^{(N)} = \boldsymbol{\delta}_{\text{MC}}$.*

Proof.

$$\boldsymbol{\delta}_{\text{MC}} = \frac{1}{M}\sum_{m=1}^{M} \log \frac{\vec{p}_{\boldsymbol{\theta}}(\boldsymbol{Y}^{(m)})}{\vec{q}(\boldsymbol{Y}^{(m)})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{Y}^{(m)}) \tag{43}$$

$$= \frac{1}{M}\sum_{m=1}^{M} \log \frac{\vec{p}_{\boldsymbol{\theta}}(\text{EOS} \mid \boldsymbol{Y}^{(m)})}{\vec{q}(\text{EOS} \mid \boldsymbol{Y}^{(m)})} \prod_{n=1}^{|\boldsymbol{Y}^{(m)}|} \frac{\vec{p}_{\boldsymbol{\theta}}(Y_n \mid \boldsymbol{Y}_{<n}^{(m)})}{\vec{q}(Y_n \mid \boldsymbol{Y}_{<n}^{(m)})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\boldsymbol{Y}^{(m)}) \tag{44}$$

$$= \frac{1}{M}\sum_{m=1}^{M} \log \prod_{n=1}^{\infty} \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})}{\vec{q}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}^{(m)}) \tag{45}$$

$$= \frac{1}{M}\sum_{m=1}^{M} \log \lim_{N\to\infty} \prod_{n=1}^{N} \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})}{\vec{q}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq N}^{(m)}) \tag{46}$$

$$= \frac{1}{M}\sum_{m=1}^{M} \lim_{N\to\infty} \log \prod_{n=1}^{N} \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})}{\vec{q}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n}^{(m)})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq N}^{(m)}) \tag{47}$$

$$= \lim_{N\to\infty} \frac{1}{M}\sum_{m=1}^{M} \log \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq N}^{(m)})}{\vec{q}(\overline{\boldsymbol{Y}}_{\leq N}^{(m)})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq N}^{(m)}) \tag{48}$$

$$= \lim_{N\to\infty} \boldsymbol{\delta}_{\text{MC}}^{(N)}. \tag{49}$$

■

Lemma 14. *The following identity holds:*

$$\nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}) = \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'} \left[\nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}'_{\leq N}) \,\middle|\, \overline{\boldsymbol{Y}}'_{\leq n} = \overline{\boldsymbol{Y}}_{\leq n} \right], \tag{50}$$

for any $N > n$.

Proof. Note that $\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}) = \sum_{\overline{\boldsymbol{y}}' \in \overline{\Sigma}^{N-n}} \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}\overline{\boldsymbol{y}}')$ for any $N > n$. Therefore, we have

$$\nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}) = \frac{\nabla_{\boldsymbol{\theta}} \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n})}{\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n})} \tag{51a}$$

$$= \sum_{\overline{\boldsymbol{y}}' \in \overline{\Sigma}^{N-n}} \frac{\nabla_{\boldsymbol{\theta}} \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}\overline{\boldsymbol{y}}')}{\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n})} \tag{51b}$$

$$= \sum_{\overline{\boldsymbol{y}}' \in \overline{\Sigma}^{N-n}} \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}\overline{\boldsymbol{y}}')}{\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}\overline{\boldsymbol{y}}') \tag{51c}$$

$$= \sum_{\overline{\boldsymbol{y}}' \in \overline{\Sigma}^{N-n}} \frac{\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{y}}' \mid \overline{\boldsymbol{Y}}_{\leq n}) \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n})}{\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n})} \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}\overline{\boldsymbol{y}}') \tag{51d}$$

$$= \sum_{\overline{\boldsymbol{y}}' \in \overline{\Sigma}^{N-n}} \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{y}}' \mid \overline{\boldsymbol{Y}}_{\leq n}) \nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}\overline{\boldsymbol{y}}') \tag{51e}$$

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'} \left[\nabla_{\boldsymbol{\theta}} \log \vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}'_{\leq N}) \,\middle|\, \overline{\boldsymbol{Y}}'_{\leq n} = \overline{\boldsymbol{Y}}_{\leq n} \right] \tag{51f}$$

■

Lemma 15. *Let $\boldsymbol{\delta}_{\text{MC}}^{(N)}$ be the truncated MC estimator of the gradient defined in Eq. (41) and $\boldsymbol{\delta}_{\text{RB}}^{(N)}$ the truncated RB estimator of the gradient defined in Eq. (42). Define $\boldsymbol{G}^N \overset{\text{def}}{=} \mathbb{E}[\boldsymbol{\delta}_{\text{MC}}^{(N)}] = \mathbb{E}[\boldsymbol{\delta}_{\text{RB}}^{(N)}]$. We have*

$$\mathbb{E}\left[\left\| \boldsymbol{\delta}_{\text{RB}}^{(N)} - \boldsymbol{G}^N \right\|^2 \right] \leq \mathbb{E}\left[\left\| \boldsymbol{\delta}_{\text{MC}}^{(N)} - \boldsymbol{G}^N \right\|^2 \right]. \tag{52}$$

Proof. Without loss of generality, we assume $M = 1$. The proof generalizes to $M > 1$ with the i.i.d. assumption.

$$\mathbb{E}\left[\left\|\boldsymbol{\delta}_{\text{RB}}^{(N)} - \boldsymbol{G}^N\right\|^2\right] \tag{53a}$$

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\left\|\sum_{n=1}^{N}\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}_{\leq n})\underbrace{\left(\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{<n}) + \nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n})\right)}_{\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n})} - \boldsymbol{G}^N \,\middle|\, \overline{\boldsymbol{Y}}_{<n} = \overline{\boldsymbol{Y}}'_{<n}\right]\right\|^2\right] \tag{53b}$$

(definition of $\boldsymbol{\delta}_{\text{RB}}^{(N)}$)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\left\|\sum_{n=1}^{N}\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^n}\left[\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}_{\leq n}^n)\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}^n) - \boldsymbol{G}^N \,\middle|\, \overline{\boldsymbol{Y}}_{<n}^n = \overline{\boldsymbol{Y}}'^n_{<n}\right]\right\|^2\right] \tag{53c}$$

(each $\overline{\boldsymbol{Y}}^n$ is distributed independently and identically to $\overline{\boldsymbol{Y}}$)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\left\|\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^1}\left[\cdots\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^N}\left[\sum_{n=1}^{N}\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}_{\leq n}^n)\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}^n) - \boldsymbol{G}^N \,\middle|\, \overline{\boldsymbol{Y}}_{<N}^N = \overline{\boldsymbol{Y}}'^N_{<N}\right]\cdots \,\middle|\, \overline{\boldsymbol{Y}}_{<1}^1 = \overline{\boldsymbol{Y}}'^1_{<1}\right]\right\|^2\right] \tag{53d}$$

(linearity of expectation)

$$\leq \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^1}\left[\cdots\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^N}\left[\left\|\sum_{n=1}^{N}\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}_{\leq n}^n)\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}^n) - \boldsymbol{G}^N\right\|^2 \,\middle|\, \overline{\boldsymbol{Y}}_{<N}^N = \overline{\boldsymbol{Y}}'^N_{<N}\right]\cdots \,\middle|\, \overline{\boldsymbol{Y}}_{<1}^1 = \overline{\boldsymbol{Y}}'^1_{<1}\right]\right] \tag{53e}$$

(Jensen's inequality)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^1}\left[\cdots\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}^N}\left[\left\|\sum_{n=1}^{N}\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}_{\leq n}^n)\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}^n) - \boldsymbol{G}^N\right\|^2\right]\right] \tag{53f}$$

(law of total expectation)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\left\|\sum_{n=1}^{N}\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}_{\leq n})\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}_{\leq n}) - \boldsymbol{G}^N\right\|^2\right] \tag{53g}$$

($\overline{\boldsymbol{Y}}^1, \ldots, \overline{\boldsymbol{Y}}^N$ are i.i.d.)

$$\leq \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\left\|\sum_{n=1}^{N}\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}_{\leq n})\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}'_{\leq N}) \mid \overline{\boldsymbol{Y}}'_{\leq n} = \overline{\boldsymbol{Y}}_{\leq n}\right] - \boldsymbol{G}^N\right\|^2\right] \tag{53h}$$

(Lemma 14)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\left\|\sum_{n=1}^{N}\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}'_{\leq n})\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}'_{\leq N}) - \boldsymbol{G}^N \,\middle|\, \overline{\boldsymbol{Y}}'_{\leq n} = \overline{\boldsymbol{Y}}_{\leq n}\right]\right\|^2\right] \tag{53i}$$

(linearity of expectation)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\left\|\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'^1}\left[\cdots\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'^N}\left[\sum_{n=1}^{N}\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}'^n_{\leq n})\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}'^n_{\leq N}) - \boldsymbol{G}^N \,\middle|\, \boldsymbol{Y}'^N_{\leq N} = \boldsymbol{Y}_{\leq N}\right]\cdots \,\middle|\, \boldsymbol{Y}'^1_{\leq 1} = \overline{\boldsymbol{Y}}_{\leq 1}\right]\right\|^2\right] \tag{53j}$$

(linearity of expectation)

$$\leq \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'^1}\left[\cdots \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'^N}\left[\left\|\sum_{n=1}^{N}\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}'^n_{\leq n})\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}'^n_{\leq N})-\boldsymbol{G}^N\right\|^2\middle|\overline{\boldsymbol{Y}}'^N_{\leq N}=\overline{\boldsymbol{Y}}_{\leq N}\right]\cdots\middle|\overline{\boldsymbol{Y}}'^1_{\leq 1}=\overline{\boldsymbol{Y}}_{\leq 1}\right]\right] \tag{53k}$$

(Jensen's inequality)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'^1}\left[\cdots \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'^N}\left[\left\|\sum_{n=1}^{N}\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}'^n_{\leq n})\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}'^n_{\leq N})-\boldsymbol{G}^N\right\|^2\right]\right] \tag{53l}$$

(law of total expectation)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}'}\left[\left\|\sum_{n=1}^{N}\mu_{\text{MC}}^n(\overline{\boldsymbol{Y}}'_{\leq n})\nabla_{\boldsymbol{\theta}}\log\vec{p}_{\boldsymbol{\theta}}(\overline{\boldsymbol{Y}}'_{\leq N})-\boldsymbol{G}^N\right\|^2\right] \tag{53m}$$

($\overline{\boldsymbol{Y}}'^1, \cdots, \overline{\boldsymbol{Y}}'^N$ are i.i.d.)

$$= \mathop{\mathbb{E}}_{\overline{\boldsymbol{Y}}}\left[\left\|\boldsymbol{\delta}_{\text{MC}}^{(N)}(\overline{\boldsymbol{Y}})-\boldsymbol{G}^N\right\|^2\right]. \tag{53n}$$

(definition of $\boldsymbol{\delta}_{\text{MC}}^{(N)}$)

■

Theorem 5. *Assuming* $\text{Var}[\boldsymbol{\delta}_{\text{RB}}]<\infty, \text{Var}[\boldsymbol{\delta}_{\text{MC}}]<\infty$*, we have*

$$\mathbb{E}\left[\|\boldsymbol{\delta}_{\text{RB}}-\boldsymbol{G}\|^2\right]\leq\mathbb{E}\left[\|\boldsymbol{\delta}_{\text{MC}}-\boldsymbol{G}\|^2\right]. \tag{17}$$

Proof. Without loss of generality, we assume $M=1$. The proof generalizes to $M>1$ with the i.i.d. assumption.

$$\mathbb{E}\left[\|\boldsymbol{\delta}_{\text{RB}}-\boldsymbol{G}\|^2\right]=\mathbb{E}\left[\left\|\lim_{N\to\infty}\boldsymbol{\delta}_{\text{RB}}^{(N)}-\boldsymbol{G}^N\right\|^2\right] \tag{54a}$$

(definition of $\boldsymbol{\delta}_{\text{RB}}$)

$$=\lim_{N\to\infty}\mathbb{E}\left[\left\|\boldsymbol{\delta}_{\text{RB}}^{(N)}-\boldsymbol{G}^N\right\|^2\right] \tag{54b}$$

(dominated convergence theorem, $\text{Var}[\boldsymbol{\delta}_{\text{RB}}]<\infty$)

$$\leq\lim_{N\to\infty}\mathbb{E}\left[\left\|\boldsymbol{\delta}_{\text{MC}}^{(N)}-\boldsymbol{G}^N\right\|^2\right] \tag{54c}$$

(Lemma 15)

$$=\mathbb{E}\left[\left\|\lim_{N\to\infty}\boldsymbol{\delta}_{\text{MC}}^{(N)}-\boldsymbol{G}^N\right\|^2\right] \tag{54d}$$

(dominated convergence theorem, $\text{Var}[\boldsymbol{\delta}_{\text{MC}}]<\infty$)

$$=\mathbb{E}\left[\|\boldsymbol{\delta}_{\text{MC}}-\boldsymbol{G}\|^2\right]. \tag{54e}$$

(Lemma 13)

■

D.2 A Note on Rao–Blackwellizing KL in Trust-Region Algorithms

The conventional Monte Carlo estimator of $\text{KL}(p_{\boldsymbol{\theta}}\,||\,q)$ used in the PPO algorithm in open-sourced RLHF libraries, e.g., [13, 40], is as follows:

$$\mu_{\text{MC}}^{\text{PPO}}=\frac{1}{M}\sum_{m=1}^{M}\frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y}^{(m)})}{p_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{Y}^{(m)})}\log\frac{p_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{Y}^{(m)})}{q(\boldsymbol{Y}^{(m)})}, \tag{55}$$

where $\boldsymbol{Y}^{(1)}, \cdots, \boldsymbol{Y}^{(M)} \overset{\text{i.i.d.}}{\sim} p_{\boldsymbol{\theta}_{\text{old}}}$. Notably, the expected value of the above estimator is *not equal to* $\text{KL}(p_{\boldsymbol{\theta}} \mid\mid q)$ and is

$$\mathbb{E}[\mu_{\text{MC}}^{\text{PPO}}] = \mathop{\mathbb{E}}_{\boldsymbol{Y} \sim p_{\boldsymbol{\theta}_{\text{old}}}} \left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y})}{p_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{Y})} \log \frac{p_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{Y})}{q(\boldsymbol{Y})} \right] = \mathop{\mathbb{E}}_{\boldsymbol{Y} \sim p_{\boldsymbol{\theta}}} \left[\log \frac{p_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{Y})}{q(\boldsymbol{Y})} \right]. \tag{56}$$

A natural question at this point is: what is the relationship between $\mu_{\text{MC}}^{\text{PPO}}$ and μ_{MC}, and why is minimizing $\mu_{\text{MC}}^{\text{PPO}}$ a valid proxy for minimizing μ_{MC}? Crucially, the KL divergence between $p_{\boldsymbol{\theta}}$ and q can be decomposed into the sum of $\mathbb{E}[\mu_{\text{MC}}^{\text{PPO}}]$ and the KL divergence between $p_{\boldsymbol{\theta}}$ and $p_{\boldsymbol{\theta}_{\text{old}}}$, as shown in the following equation:

$$\underbrace{\mathop{\mathbb{E}}_{\boldsymbol{Y} \sim p_{\boldsymbol{\theta}}} \left[\log \frac{p_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{Y})}{q(\boldsymbol{Y})} \right]}_{\mathbb{E}[\mu_{\text{MC}}^{\text{PPO}}]} + \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{Y} \sim p_{\boldsymbol{\theta}}} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y})}{p_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{Y})} \right]}_{\text{trust region}, \text{KL}(p_{\boldsymbol{\theta}}, p_{\boldsymbol{\theta}_{\text{old}}})} = \mathop{\mathbb{E}}_{\boldsymbol{Y} \sim p_{\boldsymbol{\theta}}} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y})}{q(\boldsymbol{Y})} \right] = \text{KL}(p_{\boldsymbol{\theta}} \mid\mid q). \tag{57}$$

Therefore, minimizing $\text{KL}(p_{\boldsymbol{\theta}} \mid\mid q)$ is equivalent to minimizing both $\mathbb{E}[\mu_{\text{MC}}^{\text{PPO}}]$ and $\text{KL}(p_{\boldsymbol{\theta}} \mid\mid p_{\boldsymbol{\theta}_{\text{old}}})$. Notably, since the KL divergence between the current policy and the old policy, $\text{KL}(p_{\boldsymbol{\theta}} \mid\mid p_{\boldsymbol{\theta}_{\text{old}}})$, is already constrained by PPO's clipping mechanism, the algorithm effectively focuses on penalizing only the first term, using $\mu_{\text{MC}}^{\text{PPO}}$.

A naïve approach to Rao–Blackwellizing $\mu_{\text{MC}}^{\text{PPO}}$ defined in Eq. (55), is as follows:

$$\mu_{\text{RB}}^{\text{PPO}} = \frac{1}{M} \sum_{m=1}^{M} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{Y}^{(m)})}{p_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{Y}^{(m)})} \lim_{N \to \infty} \sum_{n=1}^{N} \mathop{\mathbb{E}}_{Y_n \sim p_{\boldsymbol{\theta}_{\text{old}}}} \left[\log \frac{\vec{p}_{\boldsymbol{\theta}_{\text{old}}}(Y_n \mid \boldsymbol{Y}_{<n}^{(m)})}{\vec{q}(Y_n \mid \boldsymbol{Y}_{<n}^{(m)})} \right]. \tag{58}$$

Importantly, $\mu_{\text{RB}}^{\text{PPO}}$ does *not* give an unbiased estimate of $\mathbb{E}_{\boldsymbol{Y} \sim p_{\boldsymbol{\theta}}} \left[\log \frac{p_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{Y})}{q(\boldsymbol{Y})} \right]$, i.e.,

$$\mathbb{E}\left[\mu_{\text{RB}}^{\text{PPO}}\right] = \mathop{\mathbb{E}}_{\boldsymbol{Y} \sim p_{\boldsymbol{\theta}}} \left[\lim_{N \to \infty} \sum_{n=1}^{N} \mathop{\mathbb{E}}_{\overline{Y}_n \sim p_{\boldsymbol{\theta}_{\text{old}}}} \left[\log \frac{\vec{p}_{\boldsymbol{\theta}_{\text{old}}}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n})}{\vec{q}(\overline{Y}_n \mid \overline{\boldsymbol{Y}}_{<n})} \right] \right] \neq \mathop{\mathbb{E}}_{\boldsymbol{Y} \sim p_{\boldsymbol{\theta}}} \left[\log \frac{p_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{Y})}{q(\boldsymbol{Y})} \right]. \tag{59}$$

Therefore, we caution the reader against using this estimator as a replacement for $\mu_{\text{MC}}^{\text{PPO}}$ in practice.

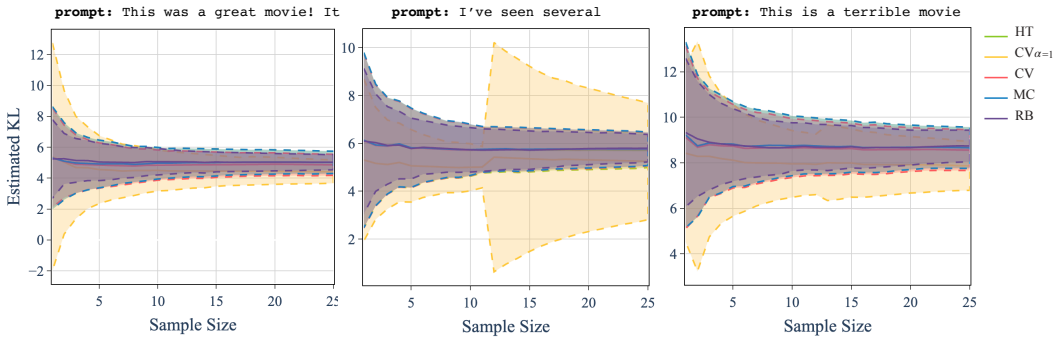

Figure 4: Comparing the bias, variance, and consistency of the estimators as the sample size increases. The μ_{CV} estimator with $\alpha = 1$ exhibits a higher standard deviation, particularly for neutral and negative prompts, where the variance becomes extremely large. In contrast, the RB estimator, μ_{RB}, achieves the lowest standard deviation.

E Additional Experiments

In Fig. 4, we visualize the KL estimates for three different prompts: (left) a positive prompt, (middle) a neutral prompt, and (right) a negative adversarial prompt. The traces represent the average estimates from all the repetitions, while the shaded regions indicate the standard deviation. Except $\mu_{\text{CV}}, \alpha = 1$, all other estimators are unbiased and consistent.

As the sample size increases, the chance of sampling from the tail of $p_{\boldsymbol{\theta}}$ also increases. These tail samples often correspond to negative movie reviews that had a high probability under the language model prior to fine-tuning, i.e., q, leading to extremely large values of $g(\boldsymbol{Y})$ and, consequently, a high standard deviation. This effect indeed depends on the prompt and is particularly pronounced for neutral and adversarial prompts.

We conducted an additional experiment to evaluate the estimators on a random subset of prompts from the UltraFeedback dataset [8]. We compute the KL divergence between `Zephyr-7B-Beta` [38] and its reference model, `Mistral-7B-v0.1` [18]. Zephyr is fine-tuned from Mistral using DPO on UltraFeedback, and as part of this fine-tuning, it is desirable not to diverge significantly from the base model.

We randomly sampled 512 prompts and generated 100 responses per prompt. For estimation, we used subsets of 1, 5, and 10 samples, reserving the remaining samples to estimate each method's standard deviation. Tab. 2 reports the KL estimate ± standard deviation for each estimator. Unlike our GPT-2 experiments, we had to use half-precision to perform inference and forward passes on a single GPU, which introduces bias in the HT and CV estimators. Consistent with our findings on the IMDB dataset, our proposed RB estimator consistently achieves the lowest standard deviation across all settings, reaffirming its stability and reliability.

Table 2: Estimated value ± empirical standard deviation of different estimators. RB estimator consistently achieves the lowest standard deviation.

	$M = 1$	$M = 5$	$M = 10$
μ_{MC}	18.05 ± 3.19	18.05 ± 1.63	$18.05 \pm \mathbf{0.75}$
μ_{HT}	18.05 ± 12.64	18.56 ± 5.34	19.24 ± 3.62
μ_{CV1}	17.17 ± 3.17	17.17 ± 1.62	17.17 ± 0.75
μ_{CV}	17.80 ± 3.18	17.80 ± 1.63	17.80 ± 0.75
μ_{RB}	$18.05 \pm \mathbf{3.16}$	$18.05 \pm \mathbf{1.61}$	$18.05 \pm \mathbf{0.75}$

F Experimental Details

F.1 Code Snippet

```python
def compute_kl(logprobs, ref_logprobs, logits, ref_logits):
    """
    Compute KL divergence using two estimators.

    Args:
        logprobs: Log probabilities of sampled actions from policy
        ref_logprobs: Log probabilities of same actions from reference
        logits: Full distribution logits from policy (all actions)
        ref_logits: Full distribution logits from reference (all actions)

    Example:
        # Policy samples action 3 from 1000 possible actions
        logprobs = [-2.3]           # Only action 3's log prob
        logits = [0.1, -0.5, ...]   # All 1000 action logits (raw)

        # MC: uses only sampled action
        # RB: uses all 1000 actions for exact expectation
    """
    # Monte Carlo: unbiased but higher variance
    kl_mc = mean(logprobs - ref_logprobs)

    # Rao-Blackwell: lower variance, uses full distribution
    log_p = log_softmax(logits)          # Normalize to log probs
    log_q = log_softmax(ref_logits)      # Normalize to log probs
    kl_rb = mean(sum(exp(log_p) * (log_p - log_q)))

    return kl_mc, kl_rb
```

F.2 RLHF Experiments

In App. F.2, we include the hyperparameters used with the RLOO algorithm for the sentiment control experiment. Each experiment takes approximately 20 minutes on a single `rtx_4090` GPU.

Hypterparameter	**Value**
Optimizer	AdamW ($\epsilon = 1\text{e}{-5}$, $\texttt{lr}= 3\text{e}{-6}$)
Scheduler	Linear
Batch Size	32
β	0.07
k	2
Number of RLOO Updates Iteration Per Epoch	4
Clip range	0.2
Sampling Temperature	1

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: The main contributions are that our proposed estimator is unbiased and has variance less than or equal to the variance of the MC estimator, and it improves the stability of RLHF training. All these claims are clearly stated in the abstract and introduction.
   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: We explain the limitations in §6.
   Guidelines:
   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.
3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [Yes]
   Justification: Yes, the main theoretical claim is that the RB estimator is unbiased and has variance less than or equal to the variance of the MC estimator, this is supported by theorem 2, and the proof is in appendix E.3. The rest of the theoretical claims regarding other estimators are proved in appendix A-F.
   Guidelines:
   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**
   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?
   Answer: [Yes]
   Justification: We provide a simple code snippet to implement our RB estimator in App. F.1, which also highlights the difference between the RB and MC estimator.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
   Answer: [Yes]
   Justification: We have released the code but, to preserve anonymity, we have not included the link to the public repository in the paper. Instead, the code is attached as supplementary material to the submission.
   Guidelines:
   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**
   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?
   Answer: [Yes]
   Justification: The experimental details is explained in App. F.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.
7. **Experiment statistical significance**
   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
   Answer: [Yes]
   Justification: When evaluating estimators in §5.1, we report the standard deviation across runs. Furthermore, for RLHF Pareto plot, we perform a permutation test in §5.2 to test significance of the results.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.
8. **Experiments compute resources**
   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
   Answer: [Yes]
   Justification: We provide this in App. F.
   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS code of ethics and we confirm that the paper conforms to the NeurIPS code of ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss this in §6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: We do not release any new models or datasets, therefore, the paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**
    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?
    Answer: [Yes]
    Justification: We cite the dataset in §5.1. While the original paper doesn't specify a particular license, it emphasizes that the data is intended for academic and non-commercial use.
    Guidelines:
    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.
13. **New assets**
    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?
    Answer: [NA]
    Justification: We do not release new assets.
    Guidelines:
    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.
14. **Crowdsourcing and research with human subjects**
    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
    Answer: [NA]
    Justification: This paper does not involve crowdsourcing nor research with human subjects.
    Guidelines:
    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.
15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**
    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?
    Answer: [NA]
    Justification: This paper does not involve crowdsourcing nor research with human subjects.
    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [NA]

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

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.