A Probability–Quality Trade-off in Aligned Language Models and its Relation to Sampling Adaptors

Naaman Tan🌴 **Josef Valvoda**☂ **Tianyu Liu**🍫 **Anej Svete**🍫
Yanxia Qin🌴 **Kan Min-Yen**🌴 **Ryan Cotterell**🍫
🌴National University of Singapore ☂University of Copenhagen 🍫ETH Zürich
{tannaaman, knmnyn}@nus.edu.sg jval@di.ku.dk
{tianyu.liu, asvete,rcotterell}@inf.ethz.ch

Abstract

The relationship between the quality of a string, as judged by a human reader, and its probability, $p(\boldsymbol{y})$ under a language model undergirds the development of better language models. For example, many popular algorithms for sampling from a language model have been conceived with the goal of manipulating $p(\boldsymbol{y})$ to place higher probability on strings that humans deem of high quality (Fan et al., 2018; Holtzman et al., 2020). In this article, we examine the probability–quality relationship in language models explicitly aligned to human preferences, e.g., through reinforcement learning through human feedback. We show that, when sampling corpora from an aligned language model, there exists a trade-off between the strings' average reward and average log-likelihood under the prior language model, i.e., the same model before alignment with human preferences. We provide a formal treatment of this phenomenon and demonstrate how a choice of sampling adaptor allows for a selection of how much likelihood we exchange for the reward.

https://github.com/tanyjnaaman/probability-quality-paradox

1 Introduction

The relationship between the probability of a string and its quality as judged by a human reader is fundamental to the goal of producing high-quality text from probabilistic language models. For example, the belief that a string $\boldsymbol{y}$ with a higher probability under a language model p should be of higher quality (Graves, 2012; Fan et al., 2018; Holtzman et al., 2020; Zhang et al., 2021) has motivated the development of sampling adaptors like top-k (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2020). The intuition behind this belief is that, under a well-calibrated language model trained primarily on human-written text, strings that occur with high probability should be more human-like, and thus judged by humans to be of higher quality. Indeed, sampling methods that skew a language model towards high-probability strings have been shown to dramatically improve the quality of text sampled from a model (Wiher et al., 2022). Over the years, several studies have since contributed to a more nuanced understanding of the probability–quality relationship (Holtzman et al., 2020; Zhang et al., 2021; Basu et al., 2021) and led to the development of more sophisticated sampling methods (Hewitt et al., 2022; Meister et al., 2022).

This paper seeks to explain the relationship between probability and quality in the specific case of **aligned** language models, i.e., language models explicitly fine-tuned to better match human preferences, e.g., through reinforcement learning with human feedback (RLHF; Christiano et al., 2017; Leike et al., 2018; Ziegler et al., 2020; Stiennon et al., 2020; Ouyang et al., 2022a,b; Korbak et al., 2022, 2023). In particular, we provide a formal argument and empirical evidence that in RLHF-tuned models, there is an inherent *anti*-correlation, i.e., a *trade-off*, between probability and quality.

In the theoretical portion of this paper, we formalize the probability–quality trade-off in aligned language models. Specifically, we show that for corpora of generated strings of a large enough size, the average log-probability under the **prior** (unaligned) language model p trades off with the average score assigned by a reward model (Gao et al., 2023). We refer to this phenomenon as a probability–quality trade-off because, in the context of aligned language models, quality is often operationalized by the reward model (Perez et al., 2022; Lee et al., 2024). This trade-off follows straightforwardly from an application of a concentration inequality and can be seen as a direct corollary of the asymptotic equipartition property (AEP; Cover and Thomas, 2006). In addition to the standard version of the AEP based on Chebyshev's inequality, we also prove a tighter version of the AEP based on a Chernoff bound that applies to certain language models, including Transformer-based

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 14805–14829
November 12-16, 2024 ©2024 Association for Computational Linguistics

models (Vaswani et al., 2017). Lastly, we show how an appropriate choice of a globally normalized sampling adaptor can control the probability of a generated string under the prior p. This finding implies that they can be used to choose a point on the probability–quality trade-off and control the average quality of the generated text. Interestingly, the trade-off predicts the emergence of Simpson's paradox, which we also observe in our experiments.

In the empirical part of our paper, we present two sets of experiments. First, with synthetic data, we construct toy language and reward models to further demonstrate our theoretical results. This gives easily reproducible empirical evidence for our claim. Then, with a second set of experiments, we show that this trade-off exists in practice with open-sourced RLHF-tuned models and that globally normalized sampling adaptors do allow us to control where a corpus of generated text will lie on the trade-off. Notably, we find that in practice, **locally normalized sampling adaptors** (e.g., nucleus sampling; Holtzman et al., 2020; Meister et al., 2023a) are sufficiently close approximations of their globally normalized counterparts, and can be effectively used to control the trade-off.

2 The Probability–Quality Relationship

Let Σ be an **alphabet**. A **string** $\boldsymbol{y} \in \Sigma^*$ is a finite list of symbols drawn from Σ. The set of all strings is called Σ's **Kleene closure** and is denoted as Σ^*. A **language model** p is a probability distribution over Σ^*. It is often assumed that there is a *positive* correlation between the quality of a string and its probability under a language model trained on human text $p(\boldsymbol{y})$ (Graves, 2012; Fan et al., 2018; Holtzman et al., 2020; Zhang et al., 2021).[1] The idea behind this assumption is that under such a model, a higher probability string should be more human-like, and thus judged by humans to be of higher quality. Because of this positive correlation, string probability is commonly used to reason about text quality in the context of natural language processing. For example, a number of studies on language modeling methods report measures of perplexity to quantify the quality of text produced by the model (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020; Pillutla et al., 2023).

More recently, several authors have observed evidence that this positive correlation breaks down at extremes, finding instead that a string's probability is positively correlated with its quality up to an inflection point, after which it becomes negatively correlated (Yang et al., 2018; Stahlberg and Byrne, 2019; Holtzman et al., 2020; Zhang et al., 2021; Meister et al., 2022). Meister et al. (2022) show that this inflection point lies near the entropy of human-written text, and hypothesize that a string should encode a similar amount of information to natural language strings to be considered high-quality. This finding has since inspired various sampling adaptors, e.g., locally typical (Meister et al., 2023b) and η-sampling (Hewitt et al., 2022).

In this paper, we investigate the probability–quality relationship in aligned language models. There are two reasons for this choice. First, aligned models have an additional constraint—they are fine-tuned to *only* produce high-quality text (as judged by a reward model). And, *a priori*, it is unclear how alignment might influence the probability–quality relationship. Second, aligned language models are mathematically similar to the conditional language models often found in machine translation and controlled text generation, for which prior work has found relationships not seen in unconditional language models (Callison-Burch et al., 2007; Banchs et al., 2015; Teich et al., 2020; Sulem et al., 2020; Lim et al., 2024).

3 Learning from Human Feedback

Because our investigation focuses on aligned language models, we now introduce RLHF, a popular algorithm for alignment. Given a task of interest, we are interested in producing strings that are aligned with human preferences for that task. Let $\mathcal{A} = \{-, +\}$ denote binary judgments of alignment, and A be an $\mathcal{A}$-valued random variable. Then, we assume the existence of a true human-aligned distribution over strings $p_+(\boldsymbol{y}) \overset{\text{def}}{=} p(\boldsymbol{y} \mid A = +)$, such that strings with high probability under p_+ also receive positive scores from human annotators. For example, we may desire that strings that are offensive should have a lower probability under p_+ for chat-related tasks. With this, we can formalize the goal of alignment as obtaining an aligned language model $q_+(\boldsymbol{y})$ such that $q_+(\boldsymbol{y})$ is a good approximation to $p_+(\boldsymbol{y})$.

RLHF is a widely used method of finding such a model. At the core of RLHF is a **reward function**, which models preferences of human annotators,

[1] Probability here refers to string likelihood under a general-purpose, *unconditional* language model. The distinction is important since we will examine the relationship between this probability and the quality of strings in an *aligned* language model.

denoted as $r\colon \Sigma^* \to (-\infty, B]$ where B is a bound in $\in \mathbb{R}_{>0}$.[2] In practice, the reward function is itself typically parameterized by a neural network and derived by modeling preferences with a Bradley–Terry model (Bradley and Terry, 1952) and a ranked dataset. The human-aligned language model $p_+(\boldsymbol{y})$, reward function $r(\boldsymbol{y})$ and prior language model $p(\boldsymbol{y})$ can be related as follows (Korbak et al., 2022):

$$p_+(\boldsymbol{y}) = \frac{p(\boldsymbol{y}) \exp\left(\frac{1}{\beta} r(\boldsymbol{y})\right)}{Z(+)}, \tag{1}$$

where $\beta \in \mathbb{R}_{>0}$ is a scaling factor and

$$Z(+) \overset{\text{def}}{=} \sum_{\boldsymbol{y} \in \Sigma^*} p(\boldsymbol{y}) \exp\left(\frac{1}{\beta} r(\boldsymbol{y})\right). \tag{2}$$

is the normalizing constant.

If we take a variational inference perspective of RLHF (Levine, 2018; Korbak et al., 2022), then the goal of RLHF is to find an aligned language model $q_+(\boldsymbol{y})$ that minimizes the backward Kullback–Leibler (KL) divergence between q_+ and the ground truth aligned distribution over strings p_+:

$$\mathrm{KL}(q_+ \mid\mid p_+) \tag{3a}$$

$$= \sum_{\boldsymbol{y} \in \Sigma^*} q_+(\boldsymbol{y}) \log \frac{q_+(\boldsymbol{y})}{p_+(\boldsymbol{y})} \tag{3b}$$

$$= \sum_{\boldsymbol{y} \in \Sigma^*} q_+(\boldsymbol{y}) \log \frac{q_+(\boldsymbol{y}) Z(+)}{p(\boldsymbol{y}) \exp\left(\frac{1}{\beta} r(\boldsymbol{y})\right)} \tag{3c}$$

$$= \log Z(+) + \mathrm{KL}(q_+ \mid\mid p) - \frac{1}{\beta} \mathop{\mathbb{E}}_{\boldsymbol{y} \sim q_+} [r(\boldsymbol{y})]. \tag{3d}$$

This objective can be seen as KL-regularized reinforcement learning (Stiennon et al., 2020).

Notably, *any* preference-aligned language model can be expressed in the framework of RLHF, even if no explicit reward function was used in training the model. As shown by Rafailov et al. (2023), for *any* aligned language model q_+ that minimizes the backward-KL objective, we can always construct a "secret" reward function r_{q_+} with:

$$r_{q_+}(\boldsymbol{y}) = \beta \left(\log \frac{q_+(\boldsymbol{y})}{p(\boldsymbol{y})} + \log Z(+) \right). \tag{4}$$

The implication of this is that the result we prove for RLHF-tuned models can be trivially extended to *any* conditionally aligned language model, like the ones often deployed in controlled text generation (Hu et al., 2017; Krause et al., 2021; Yang and Klein, 2021; Liu et al., 2021; Zhang et al., 2023).

4 Sampling Adaptors

We are often interested in sampling strings from a language model $p(\boldsymbol{y})$. Sampling is usually performed autoregressively, i.e., we sample a symbol $\overline{y} \in \overline{\Sigma} \overset{\text{def}}{=} \Sigma \cup \{\textsc{eos}\}$ iteratively from $p(\cdot \mid \boldsymbol{y}_{<t})$ at each time step t until the special end-of-sequence EOS symbol is reached. Locally normalized sampling adaptors (Meister et al., 2023a) are post-hoc alterations of $p(\cdot \mid \boldsymbol{y}_{<t})$ that have been shown to dramatically improve the quality of text produced by language models, and are often considered an integral part of a text generation pipeline (Wiher et al., 2022). Common examples include top-k (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2020).

Applying a locally normalized sampling adaptor to $p(\cdot \mid \boldsymbol{y}_{<t})$ at each t is commonly formulated as the composition of two steps. First, the application of a **transform function** $\gamma\colon \Delta^{|\overline{\Sigma}|-1} \to \mathbb{R}_{>0}{}^{|\overline{\Sigma}|}$ that maps the distribution over $\overline{\Sigma}$ to a vector of non-negative values. The transform function is responsible for the core logic of the sampling adaptor, e.g., assigning symbols outside the top-k zero probability. Second, a normalization step is performed to ensure a valid distribution over $\overline{\Sigma}$, i.e, one where $\sum_{\overline{y} \in \overline{\Sigma}} \gamma(p(\overline{y} \mid \boldsymbol{y}_{<t})) = 1$. When a locally normalized sampling adaptor is applied to a language model p, we can express the induced distribution $\ddot{p}$ as

$$\ddot{p}(\boldsymbol{y}) = \tag{5}$$
$$\frac{\gamma(p(\textsc{eos} \mid \boldsymbol{y}))}{\sum_{\overline{y} \in \overline{\Sigma}} \gamma(p(\overline{y} \mid \boldsymbol{y}))} \frac{\prod_{t=1}^{|\boldsymbol{y}|} \gamma(p(y_t \mid \boldsymbol{y}_{<t}))}{\prod_{t=1}^{|\boldsymbol{y}|} \sum_{\overline{y} \in \overline{\Sigma}} \gamma(p(\overline{y} \mid \boldsymbol{y}_{<t}))}$$

Despite their empirical success, locally normalized sampling adaptors have several undesirable theoretical properties. First, they can induce a language model that is not **tight**, i.e., one where probability mass can be placed on infinite-length strings (Welleck et al., 2020; Du et al., 2023). For instance, this can occur if zero probability is placed on EOS at every step t. Second, the normalization operation at each t can produce unexpected behavior. Specifically, because the denominator used to normalize the local distribution at each t is dependent on the probability mass assigned to *other* symbols, a string assigned a higher score with the transform function is not necessarily assigned a higher probability under the induced language model $\ddot{p}$. This can lead to less desirable strings being sampled more frequently. See App. A for an example.

[2] We assume that the reward function is bounded, following Levine (2018); Korbak et al. (2022).

We thus introduce **globally normalized sampling adaptors**, a procedure to sample from a language model that circumvents these issues. Globally normalized sampling adaptors are also defined with respect to a transform function $\boldsymbol{\gamma}$, and thus every existing locally normalized sampling adaptor can be mapped to its globally normalized counterpart. The transform function is similarly applied at each time step to $p(\cdot \mid \boldsymbol{y}_{<t})$. But without the normalization step, we define the probability of a string under the induced model $\widetilde{p}$ as

$$\widetilde{p}(\boldsymbol{y}) = \frac{\gamma\big(p(\cdot \mid \boldsymbol{y})\big)(\text{EOS})\prod_{t=1}^{|\boldsymbol{y}|}\gamma\big(p(\cdot \mid \boldsymbol{y}_{<t})\big)(y_t)}{Z_{\gamma}(\Sigma)}. \tag{6}$$

where the normalizing constant $Z_{\gamma}(\Sigma)$ is defined

$$Z_{\gamma}(\Sigma) = \tag{7}$$
$$\sum_{\boldsymbol{y}\in\Sigma^*}\gamma\big(p(\cdot \mid \boldsymbol{y})\big)(\text{EOS})\prod_{t=1}^{|\boldsymbol{y}|}\gamma\big(p(\cdot \mid \boldsymbol{y}_{<t})\big)(y_t).$$

The language model induced by a globally normalized sampling adaptor is always tight. Under the assumption that $Z_{\gamma}(\Sigma)$ is finite (Cotterell et al., 2023), it is easy to see that $\sum_{\boldsymbol{y}\in\Sigma^*}\widetilde{p}(\boldsymbol{y}) = 1$. It is also easy to see that

$$\widetilde{p}(\boldsymbol{y}) \propto \gamma\big(p(\cdot \mid \boldsymbol{y})\big)(\text{EOS})\prod_{t=1}^{|\boldsymbol{y}|}\gamma\big(p(\cdot \mid \boldsymbol{y}_{<t})\big)(y_t). \tag{8}$$

That is, the probability of a string under the adapted model is determined only by the transform function without additional renormalization, which makes it convenient to reason about how a choice of transform function can modify the properties of the induced distribution $\widetilde{p}$.

We show in §5.2 how globally normalized sampling adaptors can be used to control the probability–quality trade-off. To generate strings from a language model using a globally normalized sampling adaptor, we use the Independent Metropolis–Hastings Algorithm (IMHA; Metropolis et al., 1953; Hastings, 1970; Wang, 2022).[3] The IMHA is a Markov Chain Monte Carlo (MCMC) method that simulates sampling from a target distribution using a proposal distribution. We detail how the IMHA's acceptance–rejection protocol can be used to generate text with a globally normalized sampling adaptor in Algorithm 1. The idea is that by sampling sequentially from the proposal (i.e., the distribution induced by a locally normalized sampling adaptor, $\ddot{p}$) and appropriately accepting or rejecting samples, the generated Markov chain (i.e., the sequence of samples) converges to a stationary distribution equal to the target distribution (i.e., the one induced by a globally normalized sampling adaptor, $\widetilde{p}$). Convergence is often diagnosed by the absence of autocorrelation between consecutive samples (Deonovic and Smith, 2017). In the case of categorical variables like strings, this can be measured with Cramér's V (Ialongo, 2016).

Algorithm 1 IMH Algorithm.

```
def IMHA $(N, \widetilde{p}, \ddot{p})$:
    $\boldsymbol{y} \sim \ddot{p}$
    $\mathcal{Y} \leftarrow \{\}$
    for $n \leftarrow 1 \ldots N$ :
        $\boldsymbol{y}' \sim \ddot{p}$
        $a(\boldsymbol{y}, \boldsymbol{y}') \leftarrow \frac{\widetilde{p}(\boldsymbol{y}')\,\ddot{p}(\boldsymbol{y})}{\widetilde{p}(\boldsymbol{y})\,\ddot{p}(\boldsymbol{y}')}$
        $r \sim U(0,1)$
        if $a > r$ :
            $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{\boldsymbol{y}'\}$
            $\boldsymbol{y} \leftarrow \boldsymbol{y}'$
        else
            $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{\boldsymbol{y}\}$
    return $\mathcal{Y}$
```

5 Theoretical Results

We are now ready to discuss the theoretical contributions of this paper. In §5.1 we begin with a formal argument that there exists a fundamental trade-off between the average log-probability under the prior and the average reward for corpora sampled from an aligned language model. Then, in §5.2, we show how sampling adaptors, by shifting probability mass, can be used to choose any point on this trade-off. In §5.3, we conclude the section with a discussion of how this trade-off leads to an emergence of Simpson's paradox.

5.1 A Fundamental Trade-off

Let $q_{+}(\boldsymbol{y})$ be an aligned language model such that $q_{+}(\boldsymbol{y}) = p(\boldsymbol{y} \mid A = +)$.[4] Also, making use of a Σ^*-valued random variable $\boldsymbol{Y}$ distributed according to q_{+}, we define the **pointwise joint**

[3] Directly sampling from $\widetilde{p}$ is not possible because it requires computing the constant $Z_{\gamma}(\Sigma)$. This sum over the countably infinite set of finite strings is in practice unknown. Using IMHA avoids this issue as it only requires the knowledge of the target distribution up to a *multiplicative constant*.

[4] This assumption is not strictly necessary, but allows us to discuss the trade-off in terms of the true reward function r, rather than $q_{+}(\boldsymbol{y})$'s "secret" reward function $r_{q_{+}}$.

entropy of $q_+(\boldsymbol{y})$ as follows:

$$\mathrm{H}(\boldsymbol{Y}, A=+) \stackrel{\text{def}}{=} -\sum_{\boldsymbol{y}\in\Sigma^*} q_+(\boldsymbol{y}) \log q_+(\boldsymbol{y}). \quad (9)$$

Now, we can introduce the (N, ε)-**typical set** of q_+:

$$T_N^\varepsilon(q_+) \stackrel{\text{def}}{=} \Big\{ \mathcal{Y} \in (\Sigma^*)^N \mid \left| \mathrm{H}(\boldsymbol{Y}, A=+) + \frac{\log q_+(\mathcal{Y})}{N} \right| < \varepsilon \Big\}, \quad (10)$$

where $\mathcal{Y}$ is a **corpus** of strings and $\log q_+(\mathcal{Y}) = \sum_{n=1}^N \log q_+(\boldsymbol{y}^{(n)})$ for some $\varepsilon > 0$. In words, $T_N^\varepsilon(q_+)$ is the set of corpora of size N, i.e., bags of strings, each sampled from q_+ with average information content $\frac{\log q_+(\mathcal{Y})}{N}$ close to the pointwise joint entropy $\mathrm{H}(\boldsymbol{Y}, A=+)$.

This notion of typicality is useful because it can be shown that a sampled corpus $\mathcal{Y} = \{\boldsymbol{y}^{(n)}\}_{n=1}^N$ where $\boldsymbol{y}^{(n)} \sim q_+$ falls in $T_N^\varepsilon(q_+)$ with high probability. Let $I = -\log q_+(\boldsymbol{Y})$ be a random variable that denotes the information content of a string, and let $\mathbb{V}(I)$ denote its variance.[5] Then, with Chebyshev's inequality we can show that:

$$\mathbb{P}(\mathcal{Y} \notin T_N^\varepsilon(q_+)) \le \frac{\mathbb{V}(I)}{N\varepsilon^2} = \mathcal{O}\left(\frac{1}{N}\right). \quad (11)$$

The full derivation can be found in App. B.1. What Eq. (11) says is that if we observe a set of N strings, the probability that the corpus lies outside $T_N^\varepsilon(q_+) \to 0$ as $N \to \infty$. Equivalently, this is to say that the **sample entropy** $-\frac{\log q_+(\mathcal{Y})}{N}$ collapses around the entropy $\mathrm{H}(\boldsymbol{Y}, A=+)$ with high probability when N is large.

The above derivation is standard. However, what is less standard is the application of Bayes' rule to show that strings in the typical set display a fundamental trade-off.

Proposition 1 (Probability–quality trade-off)**.**

$$\mathbb{P}\left(\left| C + \frac{\log p(\mathcal{Y})}{N} + \frac{r(\mathcal{Y})}{\beta N} \right| < \varepsilon \right) > 1 - \delta \quad (12)$$

where $\delta = \mathcal{O}(\frac{1}{N})$ *and* $C \stackrel{\text{def}}{=} \mathrm{H}(\boldsymbol{Y} \mid A = +) - \log Z(+)$ *is a constant, and we use the shorthands* $\log p(\mathcal{Y}) = \sum_{n=1}^N \log p(\boldsymbol{y}^{(n)})$ *and* $r(\mathcal{Y}) = \sum_{n=1}^N r(\boldsymbol{y}^{(n)})$.

Prop. 1 says that a corpus $\mathcal{Y}$ of size N sampled from q_+ will have its average log-probability $\frac{\log p(\mathcal{Y})}{N}$ and average reward $\frac{r(\mathcal{Y})}{\beta N}$ bound by a constant *with high probability*. The implication of this is that the two quantities will trade off *linearly*.

Proof Sketch. Prop. 1 follows relatively straightforwardly from Eq. (10) and Eq. (11), when one observes from Eq. (4) that $q_+(\boldsymbol{y}) \propto p(\boldsymbol{y}) \exp\left(\frac{1}{\beta} r(\boldsymbol{y})\right)$, which implies $\log(q_+) = \log p(\boldsymbol{y}) + \frac{1}{\beta} r(\boldsymbol{y}) + \text{constant}$. Recall that corpora in the typical set $T_N^\varepsilon(q_+)$ have average information content $\frac{\log q_+(\boldsymbol{y})}{N}$ close to the *constant* pointwise joint entropy $\mathrm{H}(\boldsymbol{Y}, A=+)$ (Eq. (10)). That is, typical corpora *by definition* exhibit a trade-off between average log-probability under the prior $\frac{\log p(\boldsymbol{y})}{N}$ and average reward $\frac{r(\boldsymbol{y})}{N}$. Then, due to Chebyshev's inequality in Eq. (11), we have $\left| C + \frac{\log p(\mathcal{Y})}{N} + \frac{r(\mathcal{Y})}{\beta N} \right| < \varepsilon$ with probability at least $\left(1 - \frac{\mathbb{V}(I)}{N\varepsilon^2}\right)$ for all N and $\varepsilon > 0$. When $N \ge \frac{\mathbb{V}(I)}{\delta\varepsilon^2}$ for some $\delta = \mathcal{O}(\frac{1}{N})$, the above holds with probability at least $1 - \delta$, i.e., *with high probability*, and we arrive at the proposition. The full proof is provided in App. B.2. ■

Strictly speaking, Prop. 1 describes a trade-off between the average log prior probability and the average reward. However, because the reward function is often used to reflect human preferences for various notions of quality, e.g., helpfulness or concision (Perez et al., 2022; Ethayarajh et al., 2022), we can interpret this result as a probability–quality trade-off.

Assumptions. Prop. 1 relies on two key assumptions. First, that q_+ has finite entropy.[6] Second, that the variance of information content of a string is also finite, i.e., $\mathbb{V}(I) < +\infty$. We argue that neither of these assumptions are limiting in practice because we show in Prop. 3 and Prop. 5 that they hold for all Transformer-based language models, which constitute the base architecture for most modern models (Brown et al., 2020; Touvron et al., 2023).

A Tighter Bound. We remark that there exists a tighter bound for the probability–quality trade-off than the $\mathcal{O}(\frac{1}{N})$ one in Eq. (11) for specific types of language models. Specifically, we show in App. D that for transformer and n-gram based language models, the probability that sampled corpora land in the typical set $T_N^\varepsilon(q_+)$ and exhibit the trade-off grows *exponentially* quickly, i.e., the bound is $\mathcal{O}(\exp(-cN))$ for some constant $c \in \mathbb{R}_{>0}$.

5.2 Controlling the Trade-off

Ideally, we would like to choose how much probability we trade for quality when sampling corpora

[5] $\mathbb{V}(I)$, in the few papers that treat it directly, is often called the **varentropy** (Fradelizi et al., 2016).

[6] In general, this is not true. See App. C for an example.

from an aligned language model. After all, depending on the context, it may be desirable to extract higher-reward text (e.g., to improve alignment) or lower-reward text (e.g., to combat overfitting of the reward function; Azar et al., 2023; Gao et al., 2023; Wang et al., 2024; He et al., 2024).

We can leverage sampling adaptors (§4) to exercise this control. The intuition for this comes from analyzing the effect of a specific sampling adaptor on the prior string probability. That is, in App. E, we show for particular sampling adaptor that

$$\widetilde{p}_{+}(\boldsymbol{y}) \propto F_{\gamma}[p(\boldsymbol{y})] p(\boldsymbol{y}) \exp\left(\frac{1}{\beta} r(\boldsymbol{y})\right) \tag{13a}$$

$$= \widetilde{p}(\boldsymbol{y}) \exp\left(\frac{1}{\beta} r(\boldsymbol{y})\right), \tag{13b}$$

where $\widetilde{p}_{+}$ is the induced distribution when applying a sampling adaptor to an aligned model p_{+} and $F_{\gamma}[p(\boldsymbol{y})]$ refers to the aggregated series of truncation functions coming from the transform function γ that rely only on the *prior* $p(\boldsymbol{y})$. Eq. (13a) says that in some special cases, the effect of applying a globally normalized sampling adaptor to p_{+} is akin to applying it to the prior language model p and then multiplying the result by the likelihood that the generated string aligns with human preferences, i.e., the effects of the sampling adaptor can be pushed to the prior.

Given the probability–quality trade-off, this suggests that an appropriate choice of sampling adaptor can be used to control the average log-probability of sampled corpora, which then determines the average reward of generated text. For example, we could use the temperature sampling with a high temperature to produce lower probability (and thus higher reward) strings. This intuition proves to be useful and leads to an efficient way to control the trade-off, as we demonstrate in §6.2.

5.3 The Emergence of Simpson's Paradox

Following Lim et al. (2024), we now argue that the trade-off described in Prop. 1—under appropriate conditions—can lead to the emergence of Simpson's paradox. Specifically, the paradox emerges when the reward $r(\boldsymbol{y})$ is *a priori* positively correlated with string likelihood under the prior language model $p(\boldsymbol{y})$. This is not always the case, of course. However, we should expect it to be true when reward scores reflect the quality of text and the language model is well-calibrated. Thus, if we consider samples $\boldsymbol{y} \sim q_{+}$, we should expect $\log p(\boldsymbol{y})$ to be positively correlated with $r(\boldsymbol{y})$ *by assumption*. This correlation exists at the string level.

Simultaneously, in Prop. 1 we showed that an *anti*-correlation arises from the trade-off between the average log-probability $\frac{\log p(\boldsymbol{y})}{N}$ and average reward $\frac{r(\boldsymbol{y})}{\beta N}$. The positive correlation between probability and quality at the level of strings, reversed at the level of corpora, is precisely an instance of Simpson's paradox.

6 Experimental Setup

We conduct two experiments in the empirical portion of this paper. First, in §6.1 we validate the predictions of Prop. 1 with toy language and reward models. Then, in §6.2 we demonstrate that this trade-off exists in practice for open-sourced RLHF-tuned models and that globally normalized sampling adaptors can control where on this trade-off the corpus of generated text will lie. We also examine models aligned with Direct Preference Optimization (DPO; Rafailov et al., 2023) in App. F, as RLHF and DPO have the same objective.

6.1 A Toy Experiment

The trade-off described in Prop. 1 fundamentally arises as a consequence of typicality and the fact that $p_{+}(\boldsymbol{y}) \propto p(\boldsymbol{y}) \exp(\frac{1}{\beta} r(\boldsymbol{y}))$. We aim to demonstrate these theoretical principles with an easily reproducible toy experiment, where we model p_{+} and p as language models with support only over a finite subset of Σ^{*}.

Modeling p_{+}, p and r. Let $\mathcal{D} \subset \Sigma^{*}$ be a finite set of $|\mathcal{D}| = 1,000$ strings.[7] With this, let us construct a toy aligned language model $p_{+}(\boldsymbol{y})$ by sampling a distribution over $\mathcal{D}$ from the Dirichlet distribution parameterized by a $|\mathcal{D}|$-sized vector where every value is set to 0.1. We create the prior $p(\boldsymbol{y})$ by applying the softmax to a scaled and noised version of this distribution. That is, we define $p(\boldsymbol{y}) \overset{\text{def}}{=} \text{softmax}\big(p_{+}(\cdot)^{\frac{1}{\tau}} + \epsilon\big)\big) (\boldsymbol{y})$ where $\epsilon \sim U(-\kappa, \kappa)$ and $\tau \in \mathbb{R}_{>0}$, $\kappa \in \mathbb{R}_{>0}$ are our hyperparameters. We then define $r(\boldsymbol{y})$ analogously to Eq. (4) with $r(\boldsymbol{y}) = \log \frac{p_{+}(\boldsymbol{y})}{p(\boldsymbol{y})}$. The distributions of $p_{+}(\boldsymbol{y})$, $p(\boldsymbol{y})$ and $r(\boldsymbol{y})$ over the domain $\mathcal{D}$ can be seen in App. G. To induce Simpson's paradox, we tune κ and τ such that they are positively correlated, shown in Fig. 1 on the left.

Constructing Corpora with Causal Bootstrapping. An important point about the trade-off in Prop. 1 is that it occurs *with high probability*.

[7] We arbitrarily identify $\mathcal{D}$ with $\{1, 2, \ldots 1000\}$.

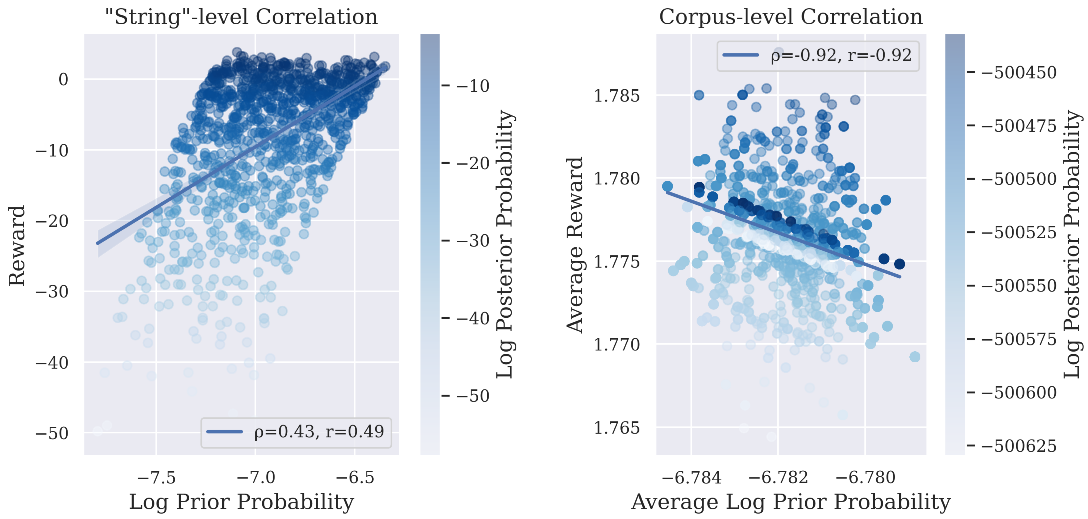

Figure 1: Illustration of the probability–quality trade-off with toy data, where quality is measured by the reward function. (Left) "String"-level correlations between probability and reward, where strings are mimicked by arbitrary objects. (Right) Corpus-level correlations between average log-probability and average reward. We include a best-fit line for corpora in the typical set, i.e., those with sample entropy close to $\mathrm{H}(p_{+})$. In both figures, the log-probability of each string or corpus is coloured according to high (dark) and low (light).

To illustrate this, we use causal bootstrapping (Little and Badawy, 2020) to construct corpora that are uniformly distributed across 10 bands of average log-probability under the prior.[8] Then, we compute and visualize the corpus probabilities, i.e., $\log p_{+}(\mathcal{Y})$ where $\mathcal{Y} \overset{\text{def}}{=} \{\boldsymbol{y}^{(n)}\}_{n=1}^{N}$. Due to Prop. 1, we expect to see that corpora exhibiting the trade-off have much higher probability than those that do not. We examine 1,000 corpora sampled this way, each with 100k samples.

6.2 The Trade-off in Practice

Here we demonstrate the existence of the probability–quality trade-off with an open-sourced aligned language model based on the Llama 2 family (Touvron et al., 2023). Using locally normalized sampling adaptors, we sample a corpus $\mathcal{Y}$ of 2,000 texts from an RLHF-tuned model q_{+}. Towards this, we randomly choose 1,000 prompts using the helpfulness dataset from Perez et al. (2022) and for each prompt, we produce two generations. Then, for every string in this corpus, we obtain its log-probability under the prior language model $\log p(\boldsymbol{y})$ and its reward $r(\boldsymbol{y})$. The prior and reward models are the same as those used to train q_{+} in an RLHF scheme. We repeat this using five locally normalized sampling adaptors at five temperatures, totaling 25 sampling schemes and thus $50,000$ $(\log p(\boldsymbol{y}), r(\boldsymbol{y}))$ pairs. To observe the trade-off, we compute the Pearson and Spearman's correlation between $\log p(\boldsymbol{y})$ and $r(\boldsymbol{y})$ at the string level, and between $\frac{\log p(\mathcal{Y})}{N}$ and $\frac{r(\mathcal{Y})}{N}$ at the corpus level.

Resampling Corpora with the IMHA. To compute corpus-level correlations we require a lot of data points of the average log-probability and average reward. However, because sampling multiple corpora is prohibitively expensive, we use standard bootstrap resampling (Efron and Tibshirani, 1994) to create multiple corpora for each of the 25 sampling schemes. Given a corpus of strings generated from q_{+} with a locally normalized sampling adaptor defined by the transform function $\boldsymbol{\gamma}$, we resample uniformly with replacement N times, accepting and rejecting each as described in Algorithm 1. This gives us a resampled corpus $\mathcal{Y}'$. Then, we compute the average log-likelihood $\frac{\log p(\mathcal{Y}')}{N}$ and average reward $\frac{r(\mathcal{Y}')}{N}$. We do this 2,000 times per sampling scheme, giving us a total of 50,000 $\left(\frac{\log p(\mathcal{Y})}{N}, \frac{r(\mathcal{Y}')}{N}\right)$ pairs, which we then use to we compute the corpus-level correlations. We set $N = 200,000$ as preliminary experiments showed that for $N \geq 200,000$ the IMHA converges, i.e., the autocorrelation measured with Cramér's V falls to < 0.10.

Sampling adaptors. The five sampling schemes we examine are: top-k sampling (Fan et al., 2018) for $k \in \{30, 50\}$, nucleus sampling (Holtzman et al., 2020) for $\pi \in \{0.90, 0.95\}$, η-sampling (Hewitt et al., 2022) and locally-typical sampling (Meister et al., 2023b). For

[8] This scheme allows us to observe low-probability corpora.

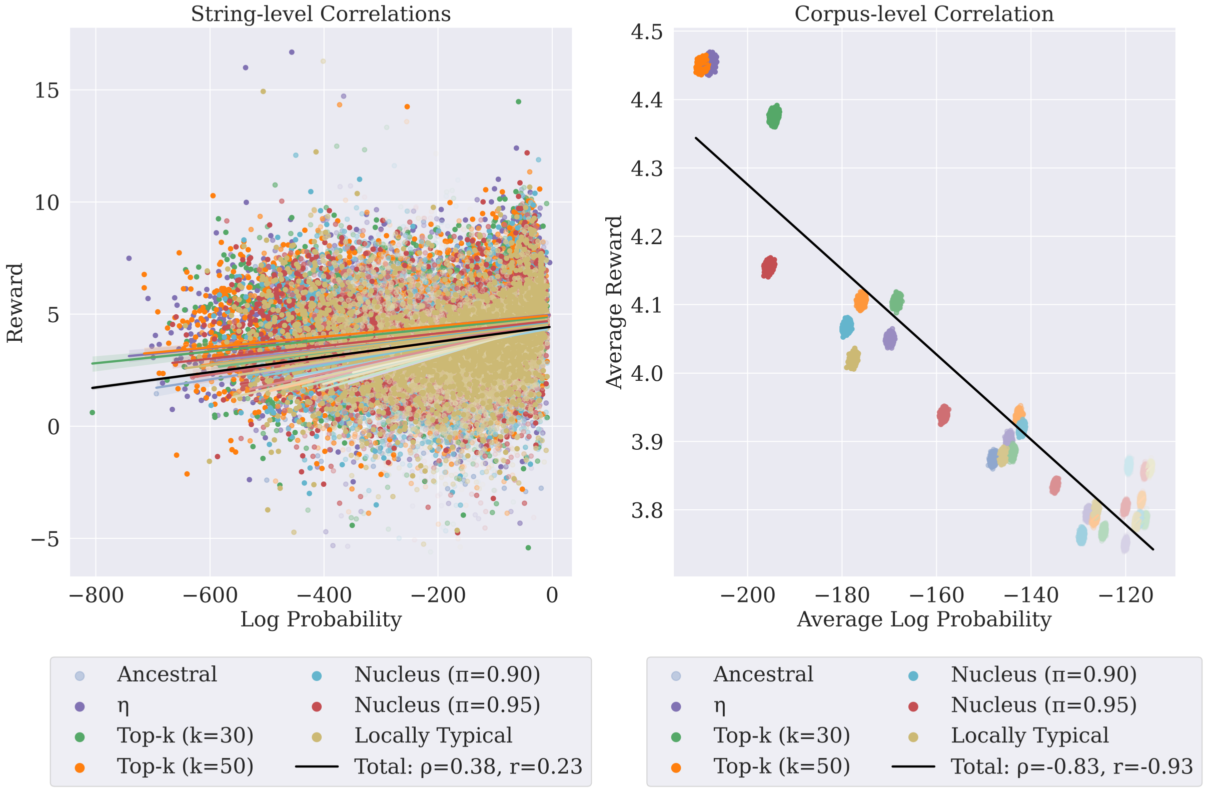

Figure 2: The probability–quality relationship, where quality is measured by the reward function. (Left) String-level correlations between log-probability and quality. (Right) Corpus-level correlations between average log-probability and average quality, with corpora created by different sampling adaptors. Higher intensity of the colours denote higher temperatures used with the sampling adaptor.

each setting, we examine five temperatures $\tau \in \{0.5, 0.75, 1.0, 1.25, 1.5\}$. This gives us a total of 25 settings that cover various real-world use cases. As a baseline, we also include ancestral sampling.

Models. We utilize the family of 7B reward, RLHF-tuned and prior language models from Rando and Tramèr (2024) based on Llama 2 7B (Touvron et al., 2023). Specifically, we use the baseline reward and RLHF-tuned models trained on the helpfulness dataset from Perez et al. (2022).

7 Results

Our results confirm our theoretical findings in §5.

A Strong Anti-correlation. In both toy and empirical settings, we observe at the corpus level a strong linear anti-correlation between the average log-probability $\frac{\log p(\mathcal{Y})}{N}$ and reward $\frac{r(\mathcal{Y})}{N}$. In the toy experiment, as shown in Fig. 1 corpora in the typical set have average log-probabilities and average rewards that exhibit a Pearson correlation of $r = -0.92$. And, importantly, these typical corpora occur with significantly higher probability than those that do not. For example, the median log-probability difference between a corpus in and out of the typical set is 10^{84} fold.[9] The results in the empirical experiment with real language models are generally consistent with the toy experiment. We observe a Pearson correlation between the average log-probability and average reward of $r = -0.93$ with relatively few outliers.

Globally Normalized Sampling Adaptors Control the Trade-off. We observe in Fig. 2 that corpora sampled with various globally normalized sampling adaptors are centered at different points on the trade-off and follow qualitatively expected trends. For example, corpora sampled with different temperatures have average log-probability that follow the expected trend—high-temperature corpora have lower average log-probability and higher average reward. And, at $\tau = 1.0$ all sampling adaptors produce corpora with higher average log-probability and lower average reward than ancestral sampling. These results are expected since lower temperature and the examined sampling adaptors skew the sampling distribution towards high probability strings. The behaviour

[9] $-500,523$ vs. $-500,718$; the difference is somewhat masked by the log-scale.

when comparing sampling adaptors with different degrees of truncation also follows our expectation, e.g., corpora sampled with nucleus sampling for $\pi = 0.95$ have lower average reward than those sampled with $\pi = 0.90$. These findings are in line with our theoretical exposition in §5.2 and suggest that we can use sampling adaptors to control the average reward of sampled corpora.

Local vs. Global Normalization. In practice, we find that the tradeoff can be controlled by directly using locally normalized sampling adaptors, i.e., without applying the IMHA post-hoc to derive a globally normalized sampling adaptor. First, we see similar results when we repeat the experiment in §6.2 without applying the IMHA's acceptance-rejection protocol. Second, for every locally normalized sampling adaptor over 95% of the samples are accepted by the IMHA. An acceptance rate of 100% implies that the proposal distribution (in this case, the distribution induced by the locally normalized sampling adaptor) is equal to the target distribution (Metropolis et al., 1953; Hastings, 1970; Deonovic and Smith, 2017; Wang, 2022).

Simpson's Paradox. We observe in both toy (Fig. 1) and empirical data (Fig. 2) the emergence of Simpson's paradox. At the string-level, we measure rank correlations of $\rho = 0.43$ in the toy setting $\rho = 0.38$ in the empirical setting. In the latter case, this positive correlation is probably explained by the fact that the reward model is trained to model preferences of *helpfulness* and the prior Llama 2 7B model has likely seen related texts in its training data. In both settings, we simultaneously find an anti-correlation at the corpus level between average log-probability and average reward. These results are consistent with our expectations in §5.3—the reversal emerges because the trade-off arises out of typicality, independently of the true relationship between probability and quality at the string level.

8 Conclusion

Our work examines the relationship between probability and reward in sampling from RLHF-tuned language models. We provide a formal argument and empirical evidence of a trade-off between these two quantities when generating text at scale. Notably, this trade-off exists as a consequence of typicality, is independent of the relationship between reward and probability at the string level, and applies to *any* conditionally aligned language model, not just those aligned with RLHF.

Moreover, we have proposed globally normalized sampling adaptors, and demonstrate their utility for selecting how much likelihood we exchange for reward. We also find that locally normalized sampling adaptors are good approximations of their globally normalized counterparts in practice, and can be directly used to control the trade-off. Altogether, these findings present a new direction of research for improving reward alignment or mitigating reward overfitting in RLHF-tuned models, and the development of sampling methods for conditional text generation.

Limitations

There are three main limitations to our work. First, is that we only conduct empirical analysis for English and Transformer-based language models. Second, we don't experiment over all sampling adaptors, e.g., we did not consider Mirostat-sampling (Basu et al., 2021) or contrastive search decoding (Su et al., 2022) in our experiments. These choices were made because the theory holds independently of these factors, though further work should consider other model architectures, sampling adaptors and models that span a variety of languages and domains. Finally, we have only examined the probability–quality relationship under the paradigm of RLHF (and equivalently, DPO, as we show in App. F), but not other alignment methods like ORPO (Hong et al., 2024), ODPO (Amini et al., 2024a) or variational BoN (Amini et al., 2024b). We leave those to future work.

Ethics Statement

We do not foresee any ethical implications.

Acknowledgments

This research is supported by the Singapore Ministry of Education Academic Research Fund Tier 1 (T1 251RES2216). Josef Valvoda is funded by the Nordic Programme for Interdisciplinary Research Grant 105178 and the Danish National Research Foundation Grant no. DNRF169. Anej Svete is supported by the ETH AI Center Doctoral Fellowship.

References

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024a. Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571*.

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024b. Variational best-of-n alignment.

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

Michael Baer. 2008. A simple countable infinite-entropy distribution. Https://www.mbbaer.com/Hinf.pdf.

Rafael E. Banchs, L. F. D'Haro, and Haizhou Li. 2015. Adequacy–fluency metrics: Evaluating MT in the continuous space model framework. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23:472–482.

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. 2021. Mirostat: A neural text decoding algorithm that directly controls perplexity. In *International Conference on Learning Representations*.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (Meta-)evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic. Association for Computational Linguistics.

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*, volume 30. Curran Associates, Inc.

Ryan Cotterell, Anej Svete, Clara Meister, Tianyu Liu, and Li Du. 2023. Formal aspects of language modeling. *arXiv preprint arXiv:2311.04329*.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA.

Benjamin E. Deonovic and Brian J. Smith. 2017. Convergence diagnostics for MCM draws of a categorical variable. *arXiv preprint arXiv:1706.04919*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Li Du, Lucas Torroba Hennigen, Tiago Pimentel, Clara Meister, Jason Eisner, and Ryan Cotterell. 2023. A measure-theoretic characterization of tight language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9744–9770, Toronto, Canada. Association for Computational Linguistics.

Bradley Efron and R. J. Tibshirani. 1994. *An Introduction to the Bootstrap*, 1st edition. Chapman and Hall/CRC.

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

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Matthieu Fradelizi, Mokshay Madiman, and Liyao Wang. 2016. Optimal concentration of information content for log-concave densities. In *High Dimensional Probability VII*, pages 45–60, Cham. Springer International Publishing.

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

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

W. K. Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Zhiwei He, Xing Wang, Wenxiang Jiao, Zhuosheng Zhang, Rui Wang, Shuming Shi, and Zhaopeng Tu. 2024. Improving machine translation with human feedback: An exploration of quality estimation as a reward model. *arXiv preprint arXiv:2401.12873*.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Cristiano Ialongo. 2016. Understanding the effect size and its measures. *Biochem Med (Zagreb)*.

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. *arXiv preprint arXiv:2310.06825*.

Tomasz Korbak, Ethan Perez, and Christopher Buckley. 2022. RL with KL penalties is better viewed as Bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1083–1091, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L. Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Seongyun Lee, Sue Hyun Park, Seungone Kim, and Minjoon Seo. 2024. Aligning to thousands of preferences via system message generalization. *arXiv preprint arXiv:2405.17977*.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. 2018. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*.

Sergey Levine. 2018. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*.

Zheng Wei Lim, Ekaterina Vylomova, Trevor Cohn, and Charles Kemp. 2024. Simpson's paradox and the accuracy-fluency tradeoff in translation. *arXiv preprint arXiv:2402.12690*.

Max A. Little and Reham Badawy. 2020. Causal bootstrapping. *arXiv preprint arXiv:1910.09648*.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Clara Meister, Tiago Pimentel, Luca Malagutti, Ethan Wilcox, and Ryan Cotterell. 2023a. On the efficacy of sampling adapters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1455, Toronto, Canada. Association for Computational Linguistics.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023b. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121.

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

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

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. 2022a. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

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. 2022b. Training language models to follow instructions with human feedback. In *Advances in Neural Information*

Processing Systems, volume 35, pages 27730–27744. Curran Associates, Inc.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Krishna Pillutla, Lang Liu, John Thickstun, Sean Welleck, Swabha Swayamdipta, Rowan Zellers, Sewoong Oh, Yejin Choi, and Zaid Harchaoui. 2023. MAUVE scores for generative models: Theory and practice. *Journal of Machine Learning Research*, 24(356):1–92.

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

Javier Rando and Florian Tramèr. 2024. Universal jailbreak backdoors from poisoned human feedback. In *The Twelfth International Conference on Learning Representations*.

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

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

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems*.

Elior Sulem, Omri Abend, and Ari Rappoport. 2020. Semantic structural decomposition for neural machine translation. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 50–57, Barcelona, Spain (Online). Association for Computational Linguistics.

Elke Teich, José Martínez Martínez, and Alina Karakanta. 2020. *Translation, Information Theory and Cognition*. Routledge.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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*, volume 30. Curran Associates, Inc.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, Songyang Gao, Nuo Xu, Yuhao Zhou, Xiaoran Fan, Zhiheng Xi, Jun Zhao, Xiao Wang, Tao Ji, Hang Yan, Lixing Shen, Zhan Chen, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. 2024. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.

Guanyang Wang. 2022. Exact convergence analysis of the independent Metropolis-Hastings algorithms. *Bernoulli*, 28(3):2012 – 2033.

Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. 2020. Consistency of a recurrent language model with respect to incomplete decoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5553–5568, Online. Association for Computational Linguistics.

Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On decoding strategies for neural text generators. *Transactions of the Association for Computational Linguistics*, 10:997–1012.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3054–3059, Brussels, Belgium. Association for Computational Linguistics.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

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

A Locally Normalized Sampling Adaptors

Despite their empirical success, the normalization performed in locally normalized sampling adaptors can lead to strings that are scored higher by the transform function γ to have lower probability under the induced language model $\ddot{p}$. This behaviour stems from the fact that the normalization is dependent on the weights assigned to *other* symbols at a given time step, and thus leads to inconsistencies at the global level. This is arguably undesirable since it makes it difficult to formally reason about how the transform function—which embodies the core logic of the sampling adaptor—influences the properties of strings sampled from $\ddot{p}$. To make this clear, we provide an example of this behaviour in top-k sampling (Fan et al., 2018).

Example 1. *Consider an alphabet $\overline{\Sigma} = \{a, b, c, \text{EOS}\}$. Let us define a language model*[10] *such that*

$$p(\boldsymbol{y}) = p(\text{EOS} \mid \boldsymbol{y}) \prod_{t=1}^{|\boldsymbol{y}|} p(y_t \mid \boldsymbol{y}_{<t}) \tag{14}$$

and

$$p(\cdot \mid \boldsymbol{y}_{<t}) = \begin{cases} [0.4, 0.1, 0.1, 0.4] & \textit{if } y_{t-1} = a \\ [0.1, 0.4, 0.2, 0.3] & \textit{if } y_{t-1} = b \\ [0.5, 0.5, 0.0, 0.0] & \textit{if } t = 1 \end{cases} \tag{15}$$

where the vector $[\ldots]$ denotes the probability assigned to $a, b, c,$ EOS, in that order. Let us now consider the probability of the strings aa and bb.

$$p(aaa) = 0.5 \times 0.4 \times 0.4 \times 0.4 = 0.032 \tag{16}$$

$$p(bbb) = 0.5 \times 0.4 \times 0.4 \times 0.3 = 0.024 \tag{17}$$

Let us now consider their probability when top-2 sampling is applied.

$$\ddot{p}(aaa) = 0.5 \times 0.5 \times 0.5 \times 0.5 = 0.0625 \tag{18}$$

$$\ddot{p}(bbb) = 0.5 \times \frac{0.4}{0.7} \times \frac{0.4}{0.7} \times \frac{0.3}{0.7} = 0.06997 \tag{19}$$

Because the transform function in top-k does not modify the symbol probability if the symbol is kept (see App. E.1), Eq. (16) is precisely the score assigned to the strings by the transform function, and Eq. (18) is the score after the normalization step. And we observe a reversal—the string "aa" has a higher score than "bb", but is later assigned a lower probability under the induced model $\ddot{p}$.

B Supplementary proofs for §5

B.1 Proof of Eq. (11)

Proof.

$$\mathbb{P}(\mathcal{Y} \notin T_N^{\varepsilon}(q_+)) = \mathbb{P}(\left|\mathrm{H}(\boldsymbol{Y}, A=+) + \frac{\log q_+(\mathcal{Y})}{N}\right| \geq \varepsilon) \tag{20a}$$

$$= \mathbb{P}(\left|\mathrm{H}(\boldsymbol{Y}, A=+) - \frac{-\log q_+(\mathcal{Y})}{N}\right| \geq \varepsilon) \tag{20b}$$

$$\leq \frac{\mathbb{V}(\frac{-\log q_+(\mathcal{Y})}{N})}{\varepsilon^2} = \frac{\mathbb{V}(-\log q_+(\boldsymbol{y}))}{N\varepsilon^2} = \frac{\mathbb{V}(I)}{N\varepsilon^2} \tag{20c}$$

Eq. (20c) holds due to Chebyshev's inequality. ■

[10] Note that this language model is not necessarily tight.

B.2 Proof of the Probability–Quality trade-off

Proof. Consider the (N, ε)-typical set

$$T_N^{\varepsilon}(q_+) \overset{\text{def}}{=} \left\{\mathcal{Y} \in (\Sigma^*)^N \mid \left|\mathrm{H}(\boldsymbol{Y}, A = +) + \frac{\log q_+(\mathcal{Y})}{N}\right| < \varepsilon\right\} \tag{21}$$

By rewriting Eq. (4)

$$\frac{r(\boldsymbol{y})}{\beta} = \log \frac{q_+(\boldsymbol{y})}{p(\boldsymbol{y})} + \log Z(+)$$

as

$$\log q_+(\boldsymbol{y}) = \frac{r(\boldsymbol{y})}{\beta} + \log p(\boldsymbol{y}) - \log Z(+),$$

and summing over all $\boldsymbol{y} \in \mathcal{Y}$, we get

$$\log q_+(\mathcal{Y}) = \frac{r(\mathcal{Y})}{\beta} + \log p(\mathcal{Y}) - N \log Z(+), \tag{22}$$

Substituting Eq. (22) into Eq. (12), we obtain

$$\begin{aligned} T_N^{\varepsilon}(q_+) &= \left\{\mathcal{Y} \in (\Sigma^*)^N \mid \left|\mathrm{H}(\boldsymbol{Y}, A = +) + \frac{r(\mathcal{Y})}{N\beta} + \frac{\log p(\mathcal{Y})}{N} - \log Z(+)\right| < \varepsilon\right\} \\ &= \left\{\mathcal{Y} \in (\Sigma^*)^N \mid \left|C + \frac{\log p(\mathcal{Y})}{N} + \frac{r(\mathcal{Y})}{N\beta}\right| < \varepsilon\right\} \end{aligned}$$

Due to Chebyshev's inequality, $\mathbb{P}(\mathcal{Y} \notin T_N^{\varepsilon}(q_+)) \leq \frac{\mathbb{V}(I)}{N\varepsilon^2}$, we have $\left|C + \frac{\log p(\mathcal{Y})}{N} + \frac{r(\mathcal{Y})}{N\beta}\right| < \varepsilon$ with probability at least $\left(1 - \frac{\mathbb{V}(I)}{N\varepsilon^2}\right)$ for all N and $\varepsilon > 0$. When $N \geq \frac{\mathbb{V}(I)}{\delta\varepsilon^2}$, the above holds with probability at least $1 - \delta$. ■

C Infinite-Entropy Language Models

A key assumption we have made in this paper is that all language models p under consideration have *finite* entropy. In general, this is not true. To make this point clear, we give an example of a simple language model whose entropy diverges.

Example 2 (A Tight LM with Infinite Entropy). *Let* $\Sigma \overset{\text{def}}{=} \{a\}$ *and define for* $t = 1, 2, \ldots$

$$p(\underbrace{a \cdots a}_{t \text{ times}}) \overset{\text{def}}{=} \frac{1}{\lg(t+1)} - \frac{1}{\lg(t+2)}. \tag{23}$$

Proposition 2. *The language model p from Eq. (23) is tight and has infinite entropy.*

Proof. The proof follows Baer (2008). We consider the language model:

$$p(\underbrace{a \cdots a}_{t \text{ times}}) \overset{\text{def}}{=} \frac{1}{\lg(t+1)} - \frac{1}{\lg(t+2)} \tag{24}$$

$p(\underbrace{a \cdots a}_{t \text{ times}})$ is positive over $t = 1, 2, \ldots$ and sums to 1 since it forms a telescoping sum with the only remaining term $\frac{1}{\lg(2)} = 1$. This proves that p is tight.

Furthermore, we can show that p's entropy is ∞. Let us denote $p_t \overset{\text{def}}{=} p(\underbrace{a \cdots a}_{t \text{ times}})$, and begin by pointing out several facts. First, the monotonicity and convexity of $\frac{1}{\lg x}$ is easily seen by noting that its first derivative is $-\frac{1}{x \lg^2 x \log 2}$ (negative for $x > 1$) and its second derivative is $\frac{\lg(x + \frac{2}{\log 2})}{x^2 \lg^3 x \log(2)}$ (positive for $x > 1$).

This will allow us to bound p_t from below with $p'_t \overset{\text{def}}{=} \frac{1}{t \lg^2 t \log 2}$, which is monotonically decreasing and less than $\frac{1}{2}$ for $t \geq 3$. Then, we point out that with basic calculus we can see that $p \lg p$ is monotonically decreasing with p for $p < \frac{1}{2}$. With these, we can say that $p'_t \lg p'_t$ is monotonically decreasing for $t \geq 2$ since $p'_t < \frac{1}{2}$ for these t. We are now ready to lower bound $\mathrm{H}(p)$ with an expression equal to infinity, thereby showing that $\mathrm{H}(p)$ is infinite:

$$\mathrm{H}(p) = -\sum_{t=1}^{\infty} p_t \lg p_t \tag{25a}$$

$$= -p_1 \lg p_1 - p_2 \lg p_2 - \sum_{t=3}^{\infty} p_t \lg p_t \tag{25b}$$

$$= -p_1 \lg p_1 - p_2 \lg p_2 - \sum_{t=3}^{\infty} \left(\frac{1}{\lg (t+1)} - \frac{1}{\lg (t+2)} \right) \lg \left(\frac{1}{\lg (t+1)} - \frac{1}{\lg (t+2)} \right) \tag{25c}$$

$$> \frac{1}{\log 2} \sum_{t=3}^{\infty} \frac{1}{t \lg^2 t} (\lg t + \lg \lg^2 t + \lg \log 2) \tag{25d}$$

$$> \frac{1}{\log 2} \sum_{t=3}^{\infty} \frac{1}{t \lg^2 t} \tag{25e}$$

$$> \frac{1}{\log 2} \int_3^{\infty} \frac{1}{t \lg t} dt \tag{25f}$$

$$> \lim_{n \to \infty} (\lg 2)(\lg \lg n - \lg \lg 3) = \infty \tag{25g}$$

■

D A Tighter (Chernoff) Bound

In this section, we give a tighter concentration inequality than the (standard) one derived with Chebyshev's inequality. The inequality displayed in Eq. (11) is weak in the sense that the average right hand size is $\mathcal{O}(\frac{1}{N})$—ideally, we desire a concentration inequality that is exponential, i.e., $\mathcal{O}(\exp(-cN))$ for some constant $c \in \mathbb{R}_{>0}$. To prove such a tighter concentration inequality, we define a class of language models that we term **sub-exponential** language models. We show that both classical n-gram language models as well as modern Transformer-based language models are sub-exponential under our definition. We further show that we can apply the Chernoff–Cramér method to argue that the sample entropy collapses around the mean exponentially quickly.

Before we delve into our derivation, we highlight what makes a direct application of a standard concentration bound, e.g., a Hoeffding bound, tricky. Consider a language model p with support everywhere on Σ^*. Furthermore, consider an enumeration $\{\boldsymbol{y}_n\}_{n=1}^{\infty}$ of Σ^* such that $n > m$ implies $p(\boldsymbol{y}_n) \leq p(\boldsymbol{y}_m)$. Observing the infinite sum $\sum_{n=1}^{\infty} p(\boldsymbol{y}_n) = 1$ is convergent, we must have that $p(\boldsymbol{y}_n) \to 0$ as $n \to \infty$. It follows by the continuity of log, that $-\log p(\boldsymbol{y}_n) \to \infty$ as $n \to \infty$. A simpler way of stating the above is that the random variable $I(\boldsymbol{y}) = -\log p(\boldsymbol{y})$, distributed according to

$$\mathbb{P}(I = \iota) = \sum_{\boldsymbol{y} \in \Sigma^*} p(\boldsymbol{y}) \mathbb{1}\{\iota = -\log p(\boldsymbol{y})\}, \tag{26}$$

is unbounded.

D.1 Prerequisites

We will now introduce several definitions and prove several results.

Definition 1 (Non-trivial Language Model)**.** *We call a language model over Σ* ***non-trivial*** *if its support is an infinite subset of Σ^*.*

Definition 2 (Rényi Entropy)**.** *Let p be a language model over Σ. The* ***Rényi entropy*** *of p is defined as*

$$\mathrm{H}_\gamma(p) = \begin{cases} \frac{1}{1-\gamma} \log \sum_{\boldsymbol{y} \in \Sigma^*} p(\boldsymbol{y})^\gamma & \gamma \in (0,1) \\ -\sum_{\boldsymbol{y} \in \Sigma^*} p(\boldsymbol{y}) \log p(\boldsymbol{y}) & \gamma = 1 \end{cases} \tag{27}$$

for $\gamma \in (0, 1]$

Definition 3. *A language model* p *is* EOS*-bounded if there exists* c *such that* $p(\text{EOS} \mid \boldsymbol{y}) > c > 0$ *for all* $\boldsymbol{y} \in \Sigma^*$.[11] *In other words,* $p(\boldsymbol{y} \oplus \text{EOS}) \leq (1-c)^{|\boldsymbol{y}|}$ *for all* $\boldsymbol{y} \in \Sigma^*$.

Proposition 3. *Let* p *be an* EOS*-bounded language model. Then,* $\mathrm{H}_\gamma(p) < +\infty$ *for* $\gamma \in (0, 1]$.

Proof. We divide the proof into two cases.
Case 1: $\gamma \in (0, 1)$**.** Consider the following manipulation

$$\sum_{\boldsymbol{y} \in \Sigma^*} p(\boldsymbol{y})^\gamma = \sum_{n=0}^{\infty} \sum_{\boldsymbol{y} \in \Sigma^n} p(\boldsymbol{y})^\gamma \tag{28a}$$

$$\leq \sum_{n=0}^{\infty} (1-c)^{n\gamma} \tag{28b}$$

$$= \sum_{n=0}^{\infty} [(1-c)^\gamma]^n \tag{28c}$$

$$= \frac{1}{1-(1-c)^\gamma} < +\infty \tag{28d}$$

The last inequality follows because $c \in (0, 1)$, and, thus, we have $0 < (1-c)^\gamma < 1$ and, thus, the geometric sum converges.
Case 2: $\alpha = 1$**.** In the case of $\alpha = 1$, Rényi entropy entropy turns into Shannon entropy. Because $\log(\cdot)$ is concave, we have

$$-\sum_{\boldsymbol{y} \in \Sigma^*} p(\boldsymbol{y}) \log p(\boldsymbol{y}) = \log \prod_{\boldsymbol{y} \in \Sigma^*} \frac{1}{p(\boldsymbol{y})^{p(\boldsymbol{y})}} \tag{29a}$$

$$= 2 \log \prod_{\boldsymbol{y} \in \Sigma^*} \left(\frac{1}{p(\boldsymbol{y})^{\frac{1}{2}}} \right)^{p(\boldsymbol{y})} \tag{29b}$$

$$\leq 2 \log \left(\sum_{\boldsymbol{y} \in \Sigma^*} \frac{p(\boldsymbol{y})}{p(\boldsymbol{y})^{\frac{1}{2}}} \right) \tag{29c}$$

$$= 2 \log \left(\sum_{\boldsymbol{y} \in \Sigma^*} p(\boldsymbol{y})^{\frac{1}{2}} \right) \tag{29d}$$

$$= \mathrm{H}_{\frac{1}{2}}(p) \tag{29e}$$

$$< +\infty \tag{29f}$$

Eq. (29c) holds due to GM–AM inequality $\prod_i x_i^{p_i} \leq \sum_i p_i x_i$, when $\sum_i p_i = 1$ and $x_i > 0 \; \forall i$. ■

Corollary 1. *Let* p *be a Transformer-based language model. Then,* $\mathrm{H}_\gamma(p) < +\infty$ *for* $\gamma \in (0, 1]$.

Proof. This follows from the proof in Du et al. (Prop. 4.7 and Thm. 5.9, 2023) that Transformer-based LMs are EOS-bounded. ■

Corollary 2. *Let* p *be a tight* n*-gram language model. Then,* $\mathrm{H}_\gamma(p) < +\infty$ *for* $\gamma \in (0, 1]$.

Proof. Tight n-gram LMs are trivially are EOS-bounded. ■

[11] Strictly speaking, autoregressive language models are not always language models, i.e., valid probability distributions over Σ^* where $\sum_{\boldsymbol{y} \in \Sigma^*} p(\boldsymbol{y}) = 1$. We highlight this difference in our exposition on locally vs. globally normalized sampling adaptors in §4.

Proposition 4. *Let p be an* EOS*-bounded language model. Then, over the interval $(0, 1]$, $\mathrm{H}_\gamma(p)$ is monotonically decreasing in γ. Moreover, if p is a non-trivial language model, then $\mathrm{H}_\gamma(p)$ is* strictly *monotonically decreasing in γ.*

Proof. Consider the following derivation

$$\begin{aligned}
\frac{\mathrm{dH}_\gamma(p)}{\mathrm{d}\gamma} =& \frac{1}{(1-\gamma)^2} \log \sum_{\boldsymbol{y}\in\Sigma^*} p(\boldsymbol{y})^\gamma + \frac{1}{1-\gamma} \sum_{\boldsymbol{y}\in\Sigma^*} \frac{p(\boldsymbol{y})^\gamma \log p(\boldsymbol{y})}{\sum_{\boldsymbol{y}'\in\Sigma^*} p(\boldsymbol{y}')^\gamma} && \text{(30a)}\\
=& \frac{1}{(1-\gamma)^2} \sum_{\boldsymbol{y}\in\Sigma^*} z(\boldsymbol{y}) \left(\log \sum_{\boldsymbol{y}'\in\Sigma^*} p(\boldsymbol{y}')^\gamma \right) + \frac{1}{(1-\gamma)} \sum_{\boldsymbol{y}\in\Sigma^*} z(\boldsymbol{y}) \log p(\boldsymbol{y}) && \text{(30b)}\\
=& \frac{1}{(1-\gamma)^2} \sum_{\boldsymbol{y}\in\Sigma^*} z(\boldsymbol{y}) \left(\log \sum_{\boldsymbol{y}'\in\Sigma^*} p(\boldsymbol{y}')^\gamma \right) + \frac{1}{(1-\gamma)^2} \sum_{\boldsymbol{y}\in\Sigma^*} z(\boldsymbol{y}) \log p(\boldsymbol{y})^{1-\gamma} && \text{(30c)}\\
=& \frac{1}{(1-\gamma)^2} \sum_{\boldsymbol{y}\in\Sigma^*} z(\boldsymbol{y}) \left(\log \sum_{\boldsymbol{y}'\in\Sigma^*} p(\boldsymbol{y}')^\gamma + \log p(\boldsymbol{y})^{1-\gamma} \right) && \text{(30d)}\\
=& \frac{1}{(1-\gamma)^2} \sum_{\boldsymbol{y}\in\Sigma^*} z(\boldsymbol{y}) \left(-\log \frac{p(\boldsymbol{y})^\gamma}{\sum_{\boldsymbol{y}'\in\Sigma^*} p(\boldsymbol{y}')^\gamma} + \log p(\boldsymbol{y}) \right) && \text{(30e)}\\
=& \frac{1}{(1-\gamma)^2} \sum_{\boldsymbol{y}\in\Sigma^*} z(\boldsymbol{y}) \log \frac{p(\boldsymbol{y})}{z(\boldsymbol{y})} && \text{(30f)}\\
=& -\frac{1}{(1-\gamma)^2} \mathrm{KL}(z \parallel p) \leq 0, && \text{(30g)}
\end{aligned}$$

where $z(\boldsymbol{y}) \overset{\text{def}}{=} \frac{p(\boldsymbol{y})^\gamma}{\sum_{\boldsymbol{y}\in\Sigma^*} p(\boldsymbol{y})^\gamma}$. Because the derivative of $\mathrm{H}_\gamma(p)$ with regard to γ is ≤ 0 on the interval $(0, 1]$, $\mathrm{H}_\gamma(p)$ is monotonically decreasing in γ. Moreover, when p is non-trivial, which implies p is not uniform nor a point mass[12], we have $z \neq p, \forall\gamma \in (0, 1)$. Thus $\frac{\mathrm{dH}_\gamma(p)}{\mathrm{d}\gamma} = -\frac{1}{(1-\gamma)^2}\mathrm{KL}(z \parallel p) < 0$, i.e., $\mathrm{H}_\gamma(p)$ is strictly monotonically decreasing on $(0, 1)$. ■

[12] i.e., the size of the support of p is 1.

Proposition 5. *Let p be an* EOS*-bounded language model. Then, $\mathbb{V}(I)$ is finite.*

Proof. We show that $\mathbb{V}(I)$ is bounded for EOS-bounded language models. Let $M \overset{\text{def}}{=} \sum_{\boldsymbol{y}\in\Sigma^*} p(\boldsymbol{y})^{\frac{1}{2}}, z(\boldsymbol{y}) \overset{\text{def}}{=} \frac{p(\boldsymbol{y})^{\frac{1}{2}}}{M}$. Note that $M = \exp(\frac{1}{2}\mathrm{H}_{\frac{1}{2}}(p)) < \infty$ due to Prop. 3. Then, we have

$$\mathbb{V}(I) = \sum_{\boldsymbol{y}\in\Sigma^*} p(\boldsymbol{y})\left(\log\frac{1}{p(\boldsymbol{y})}\right)^2 - \mathrm{H}(p)^2 \tag{31a}$$

$$\leq \left(\sum_{\boldsymbol{y}\in\Sigma^*} p(\boldsymbol{y})^{\frac{1}{2}}\log\frac{1}{p(\boldsymbol{y})}\right)^2 - \mathrm{H}(p)^2 \tag{31b}$$

$$= \left(2\sum_{\boldsymbol{y}\in\Sigma^*} p(\boldsymbol{y})^{\frac{1}{2}}\log\left(\frac{1}{p(\boldsymbol{y})}\right)^{\frac{1}{2}}\right)^2 - \mathrm{H}(p)^2 \tag{31c}$$

$$= \left(2\sum_{\boldsymbol{y}\in\Sigma^*} Mz(\boldsymbol{y})\log\frac{1}{Mz(\boldsymbol{y})}\right)^2 - \mathrm{H}(p)^2 \tag{31d}$$

$$= \left(-2M\log M + 2\sum_{\boldsymbol{y}\in\Sigma^*} Mz(\boldsymbol{y})\log\frac{1}{z(\boldsymbol{y})}\right)^2 - \mathrm{H}(p)^2 \tag{31e}$$

$$= (-2M\log M + 2M\mathrm{H}_1(z))^2 - \mathrm{H}(p)^2 \tag{31f}$$

$$\leq \left(|2M\log M| + |2M\mathrm{H}_{1/2}(z)|\right)^2 - \mathrm{H}(p)^2 \qquad \text{(monotonicity of Rényi Entropy)} \tag{31g}$$

$$= \left(2M\log M + 4M\log\left(\sum_{\boldsymbol{y}\in\Sigma^*} z(\boldsymbol{y})^{1/2}\right)\right)^2 - \mathrm{H}(p)^2 \tag{31h}$$

$$= \left(2M\log M + 4M\left(\frac{3}{4}\mathrm{H}_{1/4}(p) - \frac{1}{2}\log M\right)\right)^2 - \mathrm{H}(p)^2 \tag{31i}$$

$$< +\infty \tag{31j}$$

■

Definition 4 (Rényi Gap)**.** *Let p be a language model and let $\gamma \in (0, 1]$. The* ***Rényi gap*** *is defined as*

$$\Delta\gamma(p) = \mathrm{H}_\gamma(p) - \mathrm{H}(p) \tag{32}$$

Corollary 3. *Let p be a language model and let $\alpha \in (0, 1]$. Then, the Rényi gap $\Delta\gamma(p)$ is non-negative.*

Proof. This follows from Prop. 4. ■

Lemma 1. *Let p be a non-trivial,* EOS*-bounded language model. Then, for any $\varepsilon > 0$, there exists an $\gamma \in (0, 1)$ such that the Rényi gap $0 < \Delta\gamma < \varepsilon$.*

Proof. This follows from the $\Delta\gamma$ being a continuous monotonically decreasing function in γ and $\Delta\gamma = 0$ when $\gamma = 1$. ■

D.2 A Tighter Concentration Bound

We now introduce a sharper version of the AEP for EOS-bounded language models. As shown in App. D.1, this includes Transformer-based language models, which constitute the base architecture for most modern models (Brown et al., 2020; Touvron et al., 2023). The theorem is stated below.

Theorem 2. *Let p be an* EOS*-bounded, non-trivial language model. Then, there exists a function $s(\varepsilon) > 0$ such that, for any $\varepsilon > 0$, we have*

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^{N} I_n - \mathrm{H}(p)\right| \geq \varepsilon\right) \leq 2\exp(-s(\varepsilon)N) \tag{33}$$

with

$$s(\varepsilon) \overset{\text{def}}{=} -t(\varepsilon)(\Delta_{1-t(\varepsilon)}(p) - \varepsilon) \tag{34}$$

Proof. To prove the result, we apply a Chernoff bound. This is a one-sided bound and the other will follow by symmetry.

$$\begin{align}
\mathbb{P}\left(\frac{1}{N}\sum_{n=1}^{N} I_n - \mathrm{H}(p) \geq \varepsilon\right) &\leq \inf_{t>0} \exp(-t\varepsilon)\,\mathbb{E}\prod_{n=1}^{N}\exp\left(\frac{t}{N}(I_n - \mathrm{H}(p))\right) \tag{35a}\\
&= \inf_{t>0} \exp(-t\varepsilon)\exp(-t\mathrm{H}(p))\,\mathbb{E}\prod_{n=1}^{N}\exp\left(\frac{t}{N}I_n\right) \tag{35b}\\
&= \inf_{t>0} \exp(-t\varepsilon)\exp(-t\mathrm{H}(p))\prod_{n=1}^{N}\mathbb{E}\exp\left(\frac{t}{N}I_n\right) \tag{35c}\\
&= \inf_{t>0} \exp(-t\varepsilon)\exp(-t\mathrm{H}(p))\prod_{n=1}^{N}\sum_{\boldsymbol{y}\in\Sigma^*} p(\boldsymbol{y})\exp\left(-\frac{t}{N}\log p(\boldsymbol{y})\right) \tag{35d}\\
&= \inf_{t>0} \exp(-t\varepsilon)\exp(-t\mathrm{H}(p))\prod_{n=1}^{N}\sum_{\boldsymbol{y}\in\Sigma^*} p(\boldsymbol{y})^{1-\frac{t}{N}} \tag{35e}\\
&= \inf_{t>0} \exp(-t\varepsilon)\exp(-t\mathrm{H}(p))\prod_{n=1}^{N}\exp\left(\frac{t}{N}\mathrm{H}_{1-\frac{t}{N}}(p)\right) \tag{35f}\\
&= \inf_{t>0} \exp(-t\varepsilon)\exp(-t\mathrm{H}(p))\exp\left(t\mathrm{H}_{1-\frac{t}{N}}(p)\right) \tag{35g}\\
&= \inf_{t>0} \exp(-t\varepsilon)\exp(t(\mathrm{H}_{1-\frac{t}{N}}(p) - \mathrm{H}(p))) \tag{35h}\\
&= \inf_{t>0} \exp(-t\varepsilon)\exp(t(\Delta_{1-\frac{t}{N}}(p))) \tag{35i}\\
&= \inf_{t>0} \exp\left(t(\Delta_{1-\frac{t}{N}}(p) - \varepsilon)\right) \tag{35j}\\
&= \inf_{t'>0} \exp\big(Nt'(\Delta_{1-t'}(p) - \varepsilon)\big) \tag{35k}
\end{align}$$

Now, by Lemma 1, for any $\epsilon > 0$ we can find a $0 < t(\varepsilon) < 1$ such that $\Delta_{1-t(\varepsilon)}(p) - \varepsilon < 0$. Thus, we have

$$\mathbb{P}\left(\frac{1}{N}\sum_{n=1}^{N} I_n - \mathrm{H}(p) \geq \varepsilon\right) \leq \exp(Nt(\varepsilon)(\Delta_{1-t(\varepsilon)}(p) - \varepsilon)). \tag{36}$$

Similarly, we have:

$$\mathbb{P}\left(\frac{1}{N}\sum_{n=1}^{N} I_n - \mathrm{H}(p) \leq -\varepsilon\right) \leq \exp(Nt(\varepsilon)(-\Delta_{1-t(\varepsilon)}(p) - \varepsilon)) \leq \exp(Nt(\varepsilon)(\Delta_{1-t(\varepsilon)}(p) - \varepsilon)) \tag{37}$$

And, finally, we get:

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^{N} I_n - \mathrm{H}(p)\right| \geq \varepsilon\right) \leq \mathbb{P}\left(\frac{1}{N}\sum_{n=1}^{N} I_n - \mathrm{H}(p) \geq \varepsilon\right) + \mathbb{P}\left(\frac{1}{N}\sum_{n=1}^{N} I_n - \mathrm{H}(p) \leq -\varepsilon\right)$$
$$\leq 2\exp(Nt(\varepsilon)(\Delta_{1-t(\varepsilon)}(p) - \varepsilon)). \tag{38}$$

Substituting in $s(\varepsilon) = -t(\varepsilon)(\Delta_{1-t(\varepsilon)}(p) - \varepsilon) > 0$, we arrive at

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^{N} I_n - \mathrm{H}(p)\right| \geq \varepsilon\right) \leq 2\exp(-s(\varepsilon)N), \tag{39}$$

which tends to 0 exponentially quickly as $N \to \infty$. Note that $2\exp(-s(\varepsilon)N)$ is $\mathcal{O}(\exp(-cN))$ for $c = s(\varepsilon)$, which proves the result. ■

In words, with respect to Transformer-based language models, Theorem 2 says that if we have a model p and randomly sample N strings $\mathcal{Y} \sim p$, when we average their surprisal values we approach the entropy of p exponentially quickly. One caveat is that the constant in the exponential is not a universal constant, i.e., it depends on ε. This is less desirable, of course, but it is an improvement over the $\mathcal{O}(\frac{1}{N})$ rate given by an application of the standard AEP. We leave finding a universal constant for EOS-bounded language models to future work.

E Sampling Adaptors and String Probability

In this section, we provide, by means of a simple example, an intuition of how an appropriate choice of globally normalized sampling adaptor can be used to control generated strings' average probability under the prior $p(\boldsymbol{y})$. Inspired by Meister et al. (2023a), we note that most $\boldsymbol{\gamma}$ can be formulated as the composition of truncation and scaling functions. The **truncation function** $C\colon \Sigma^* \to \mathcal{P}(\overline{\Sigma})$ is a function used to find the set of symbols that meets specified criteria given the prior context, so that symbols deemed likely to lead to undesirable text can have their probability reassigned to other symbols, e.g., to only keep the top-k symbols. The **scaling function** $\kappa\colon \mathbb{R}_{>0}{}^{|\overline{\Sigma}|} \to \mathbb{R}_{>0}{}^{|\overline{\Sigma}|}$ is a simple scaling of the symbol probability, e.g., to the power of $\frac{1}{\tau}$ for some temperature parameter $\tau \in \mathbb{R}_{>0}$. With these definitions we can express a transform function $\boldsymbol{\gamma}$ as

$$\boldsymbol{\gamma}(p(\cdot \mid \boldsymbol{y}_{<t}))(y) = \kappa(p(y \mid \boldsymbol{y}_{<t}))\mathbb{1}\{y \in C(p(\cdot \mid \boldsymbol{y}_{<t}))\}. \tag{40}$$

That is, given a symbol distribution $p(\cdot \mid \boldsymbol{y}_{<t})$, we apply the scaling function to scale symbol probabilities as needed and then remove symbols according to the truncation function to arrive at the output unnormalized distribution. For instance, we can express the transform function in nucleus sampling (Holtzman et al., 2020) with:

$$\kappa_{\text{nucleus}}\big((p(y \mid \boldsymbol{y}_{<t})\big) = \mathbb{I}\big(p(y \mid \boldsymbol{y}_{<t})\big) \tag{41a}$$
$$C_{\text{nucleus}}\big((p(y \mid \boldsymbol{y}_{<t})\big) = \operatorname*{argmin}_{\overline{\Sigma}' \subseteq \overline{\Sigma}} |\overline{\Sigma}'| \qquad s.t. \sum_{y \in \overline{\Sigma}'} p(y \mid \boldsymbol{y}_{<t}) \geq \pi \tag{41b}$$

where $\mathbb{I}$ denotes the identity function and $\pi \in \mathbb{R}_{>0}$ is a hyperparameter. See App. E.1 for more examples.

Let p be a language model and p_+ its aligned counterpart. We consider the probability of a string under the induced distribution $\widetilde{p}_+$ when a globally normalized sampling adaptor is used with an aligned model p_+. Since our goal is simply to provide an intuition behind what might happen by analyzing the effects of the global sampling adaptor, we make the following simplifying assumptions. First, we assume that the alignment does not modify the truncation function C: $C(p(\cdot \mid \boldsymbol{y})) = C(p_+(\cdot \mid \boldsymbol{y}))$ for all $\boldsymbol{y} \in \Sigma^*$. Second, we assume that $\kappa(\cdot) = \mathbb{I}(\cdot)$. With this, we can derive

$$\widetilde{p}_+(\boldsymbol{y}) \propto \gamma\big(p_+(\cdot \mid \boldsymbol{y})\big)(\text{EOS})\prod_{t=1}^{|\boldsymbol{y}|}\gamma\big(p_+(\cdot \mid \boldsymbol{y}_{<t})\big)(y_t)$$

$$= \kappa(p_+(\text{EOS} \mid \boldsymbol{y}))\mathbb{1}\{\text{EOS} \in C(p_+(\cdot \mid \boldsymbol{y}))\}\prod_{t=1}^{|\boldsymbol{y}|}\kappa(p_+(y_t \mid \boldsymbol{y}_{<t}))\mathbb{1}\{y_t \in C(p_+(\cdot \mid \boldsymbol{y}_{<t}))\} \tag{42a}$$

$$= p_+(\text{EOS} \mid \boldsymbol{y})\mathbb{1}\{\text{EOS} \in C(p_+(\cdot \mid \boldsymbol{y}))\}\prod_{t=1}^{|\boldsymbol{y}|}p_+(y_t \mid \boldsymbol{y}_{<t})\mathbb{1}\{y_t \in C(p_+(\cdot \mid \boldsymbol{y}_{<t}))\} \tag{42b}$$

$$= p_+(\text{EOS} \mid \boldsymbol{y})\mathbb{1}\{\text{EOS} \in C(p_+(\cdot \mid \boldsymbol{y}))\}\prod_{t=1}^{|\boldsymbol{y}|}p_+(y_t \mid \boldsymbol{y}_{<t})\mathbb{1}\{y_t \in C(p_+(\cdot \mid \boldsymbol{y}_{<t}))\} \tag{42c}$$

$$= p_+(\text{EOS} \mid \boldsymbol{y})\mathbb{1}\{\text{EOS} \in C(p(\cdot \mid \boldsymbol{y}))\}\prod_{t=1}^{|\boldsymbol{y}|}p_+(y_t \mid \boldsymbol{y}_{<t})\mathbb{1}\{y_t \in C(p(\cdot \mid \boldsymbol{y}_{<t}))\} \tag{42d}$$

$$= \mathbb{1}\{\text{EOS} \in C(p(\cdot \mid \boldsymbol{y}))\}\prod_{t=1}^{|\boldsymbol{y}|}\mathbb{1}\{y_t \in C(p(\cdot \mid \boldsymbol{y}_{<t}))\}p_+(\text{EOS} \mid \boldsymbol{y})\prod_{t=1}^{|\boldsymbol{y}|}p(y_t \mid \boldsymbol{y}_{<t}) \tag{42e}$$

$$= \mathbb{1}\{\text{EOS} \in C(p(\cdot \mid \boldsymbol{y}))\}\prod_{t=1}^{|\boldsymbol{y}|}\mathbb{1}\{y_t \in C(p(\cdot \mid \boldsymbol{y}_{<t}))\}p_+(\boldsymbol{y}) \tag{42f}$$

$$\propto \underbrace{\mathbb{1}\{\text{EOS} \in C(p(\cdot \mid \boldsymbol{y}))\}\prod_{t=1}^{|\boldsymbol{y}|}\mathbb{1}\{y_t \in C(p(\cdot \mid \boldsymbol{y}_{<t}))\}}_{F_\gamma[p(\boldsymbol{y})]}\,p(\boldsymbol{y})\exp\left(\frac{1}{\beta}r(\boldsymbol{y})\right) \tag{42g}$$

$$= F_\gamma[p(\boldsymbol{y})]p(\boldsymbol{y})\exp\left(\frac{1}{\beta}r(\boldsymbol{y})\right) \tag{42h}$$

$$\tag{42i}$$

We see that, in this simplified setting, the resulting probability of a string under the adapted and aligned model $\widetilde{p}_+$ equals the prior probability scaled by the reward function and by the truncation factors. Since different sampling adaptors affect this relationship differently, this simplified example suggests that an appropriate choice of globally normalized sampling adaptor can be used to effectively select strings based on their probability under the prior. For example, using the transform function from top-k sampling will lead to the generation of corpora with higher average probability under the prior,[13] and thus, by the anti-correlation derived in our paper, with lower average reward.

E.1 Examples of Sampling Adaptors

We note that these largely correspond to the examples in Meister et al. (2023a).

Example 3. *We recover **ancestral sampling** when $\kappa(\cdot) = \mathbb{I}(\cdot)$ and $C(p(\cdot \mid \boldsymbol{y}_{<t})) = \overline{\Sigma}$.*

Example 4. *We recover **temperature sampling** when $\kappa(p(y_t \mid \boldsymbol{y}_{<t})) \propto p(y_t \mid \boldsymbol{y}_{<t})^{\frac{1}{\tau}}$ and $C(p(\cdot \mid \boldsymbol{y}_{<t})) = \overline{\Sigma}$.*

Example 5. *We recover **top-k** sampling (Fan et al., 2018) when $\kappa(\cdot) = \mathbb{I}(\cdot)$ and*

$$C(p(\cdot \mid \boldsymbol{y}_{<t})) = \operatorname*{argmax}_{\overline{\Sigma}' \subseteq \overline{\Sigma}} \sum_{y \in \overline{\Sigma}'} p(y \mid \boldsymbol{y}_{<t}) \qquad s.t.\ |\overline{\Sigma}'| = k \tag{43}$$

i.e., the set of top-k most probable symbols.

[13] For $k < |\overline{\Sigma}|$.

Example 6. *We recover* ***locally typical sampling*** *(Meister et al., 2023b) when when* $\kappa(\cdot) = \mathbb{I}(\cdot)$ *and*

$$C(p(\cdot \mid \boldsymbol{y}_{<t})) = \operatorname*{argmin}_{\overline{\Sigma}' \subseteq \overline{\Sigma}} \sum_{y \in \overline{\Sigma}'} \left| \mathrm{H}(p(\cdot \mid \boldsymbol{y}_{<t})) + \log p(y \mid \boldsymbol{y}_{<t}) \right| \qquad s.t. \sum_{y \in \overline{\Sigma}'} p(y \mid \boldsymbol{y}_{<t}) \geq \pi \tag{44}$$

i.e., the set of items with log-probability closest to the symbol-level entropy that collectively have probability mass $\geq \pi$.

Example 7. *We recover* η***-sampling*** *(Hewitt et al., 2022) when* $\kappa(\cdot) = \mathbb{I}(\cdot)$ *and*

$$C(p(\cdot \mid \boldsymbol{y}_{<t})) = \{y \in \overline{\Sigma} \mid p(y \mid \boldsymbol{y}_{<t}) > \eta\} \tag{45}$$

that is, the set of symbols with probability greater than η*, where* $\eta = \min(\epsilon, \sqrt{\epsilon} \exp(-\mathrm{H}(p(\cdot \mid \boldsymbol{y}_{<t}))))$.

F The Probability–Quality Trade-off in DPO-aligned Language Models

The probability–quality trade-off also applies to models aligned with direct preference optimization (DPO; Rafailov et al., 2023). Though an explicit reward function is not needed to train a language model with DPO, the training scheme maximises the same backward KL divergence objective as RLHF (Eq. (3); Korbak et al., 2023; Rafailov et al., 2023; Azar et al., 2023). We should thus expect that Prop. 1 applies to these models and observe the trade-off when we construct a reward function r_{q_+} using the prior p and aligned model q_+ as in Eq. (4). We employ the same setup as in §6.2.

Models. We use the 7B DPO-aligned and prior language models from Lee et al. (2024), both of which are based on Mistral 7B v0.2 (Jiang et al., 2023). The DPO-aligned model is fine-tuned on the Multifaceted Collection (Lee et al., 2024), a dataset with 192k samples capturing preferences of style (e.g., clarity, tone), informativeness, and harmlessness, among others. We construct the "secret" reward function as $r_{q_+}(\boldsymbol{y}) = \frac{q_+(\boldsymbol{y})}{p(\boldsymbol{y})}$, omitting the constant term.

Results. We observe results identical to the setting with RLHF-tuned models. Specifically, we observe a strong anti-correlation (Pearson correlation of $r = -0.97$), trade-off control using sampling adaptors, and the emergence of Simpson's paradox. These are expected since RLHF and DPO have the same minimization objective, thus supporting our formal arguments in §5.

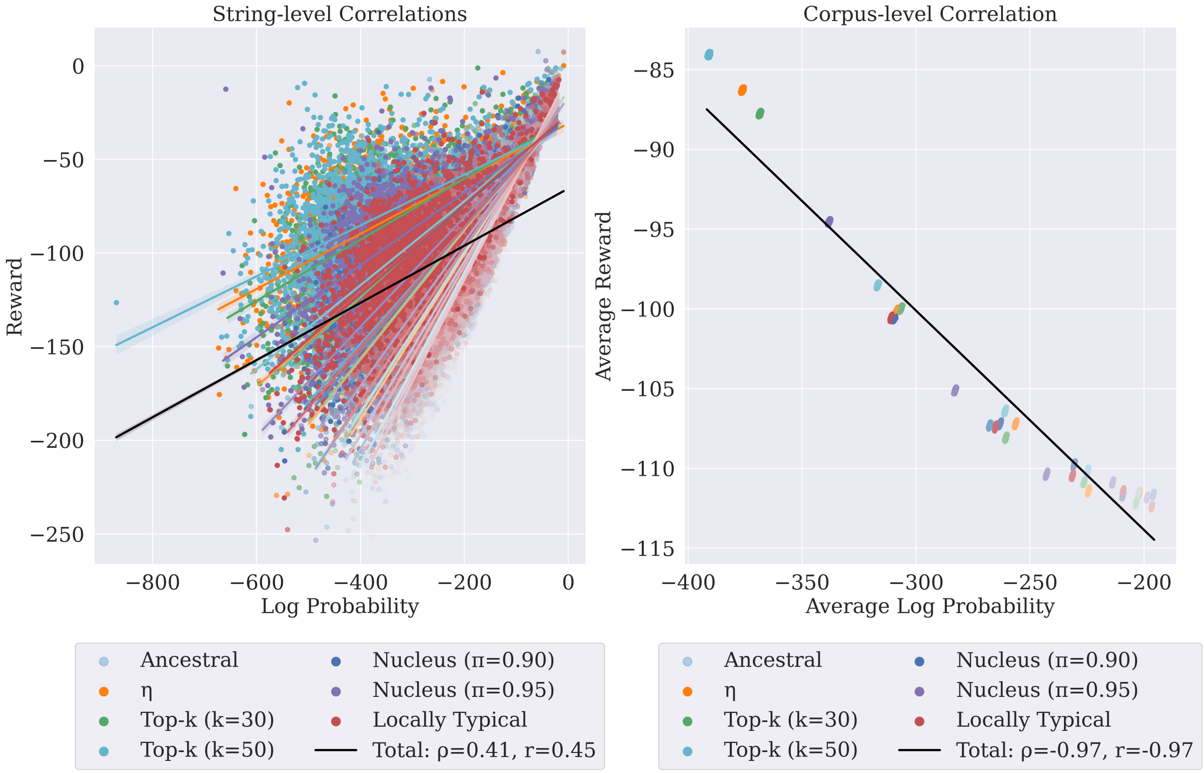

Figure 3: The probability–quality relationship in DPO-tuned models, where quality is measured by the secret reward function. (Left) String-level correlations between log-probability and quality. (Right) Corpus-level correlations between average log-probability and average quality, with corpora created by different sampling adaptors. Higher intensity of the colours denote higher temperatures used with the sampling adaptor.

G Toy Experiment Distributions

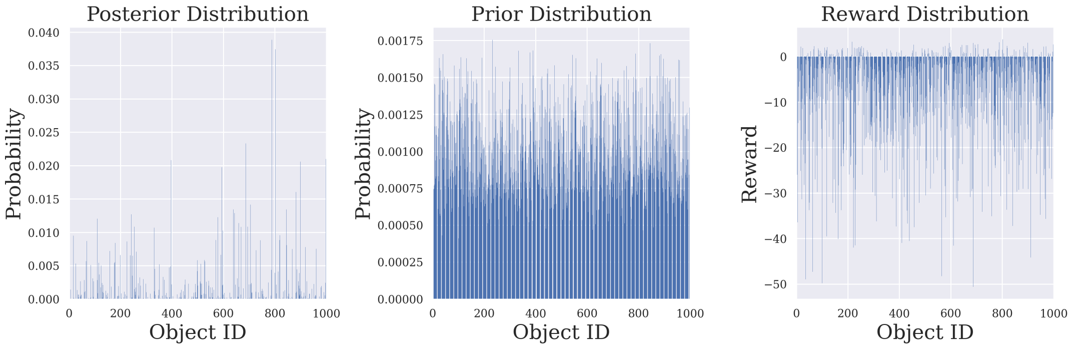

Figure 4: Toy models of $p_+(x)$, $p(x)$ and $r(x)$ analogous to the distributions over strings.