ENTROPY-ALIGNED DECODING OF LMS FOR BETTER WRITING AND REASONING

Kareem Ahmed
Department of Computer Science
University of California, Irvine
ahmedky@uci.edu

Sameer Singh
Department of Computer Science
University of California, Irvine
sameer@uci.edu

ABSTRACT

Language models (LMs) are trained on billions of tokens in an attempt to recover the true language distribution. Still, vanilla random sampling from LMs yields low quality generations. *Decoding algorithms* attempt to restrict the LM distribution to a set of high-probability continuations, but rely on greedy heuristics that introduce myopic distortions, yielding sentences that are homogeneous, repetitive and incoherent. In this paper, we introduce EPIC, a hyperparameter-free decoding approach that incorporates the entropy of future trajectories into LM decoding. EPIC explicitly regulates the amount of uncertainty expressed at every step of generation, aligning the sampling distribution's entropy to the aleatoric (data) uncertainty. Through ENTROPY-AWARE LAZY GUMBEL-MAX sampling, EPIC manages to be exact, while also being efficient, requiring only a sublinear number of entropy evaluations per step. Unlike current baselines, EPIC yields sampling distributions that are empirically well-aligned with the entropy of the underlying data distribution. Across creative writing and summarization tasks, EPIC consistently improves LM-AS-JUDGE preference win-rates over widely used decoding strategies. These preference gains are complemented by automatic metrics, showing that EPIC produces more diverse generations and more faithful summaries. We also evaluate EPIC on mathematical reasoning, where it outperforms all baselines.

1 INTRODUCTION

Despite the unprecedented capabilities of LMs at modeling natural language, generating diverse, non-repetitive, and coherent text remains elusive. Stochastic sampling tends to produce low quality text due to sampling the unreliable tails of the distribution.[1] Consequently, many *decoding algorithms* have been proposed that either statically (Fan et al., 2018b) or dynamically (Holtzman et al., 2020b; Freitag & Al-Onaizan, 2017a; Minh et al., 2025; Hewitt et al., 2022a; Meister et al., 2023) intervene on the next-token distribution, in an effort to truncate, or restrict, the LM distribution to a *nucleus set* of the most promising sentences. Unfortunately, these heuristics are often ad hoc, based only on empirical observations. Furthermore, these ad hoc heuristics are often implemented *greedily* at every step of the generation, yielding a *myopic* rendition of the desired target distribution.

Instinctively, one might be tempted to associate with high-quality *human-like* generations a high probability under the LM distribution. Paradoxically, that does not seem to be the case. In fact, empirical studies have shown (Holtzman et al., 2020b; Zhang et al., 2021; Meister et al., 2022) the quality of a generation to exhibits a peculiar relationship with regards to its probability under the LM distribution: there is indeed a positive correlation between the probability prescribed by a LM to a generation and its perceived quality *up to an inflection point* after which the perceived quality of the generation negatively correlates with its probability. This empirical observation offers a heuristic explanation for the phenomenon whereby lower probability generations returned by stochastic decoding approaches appear to outperform text generated using probability-maximizing approaches such as beam search (Lowerre, 1976; Reddy, 1977; Sutskever et al., 2014; Bahdanau et al., 2015).

Information theory offers a principled account of why naive LM sampling fails. The negative log-probability of an event corresponds to its *surprisal*; under the asymptotic equipartition property

[1] A limitation of softmax is that the resulting probability distribution always has full support.

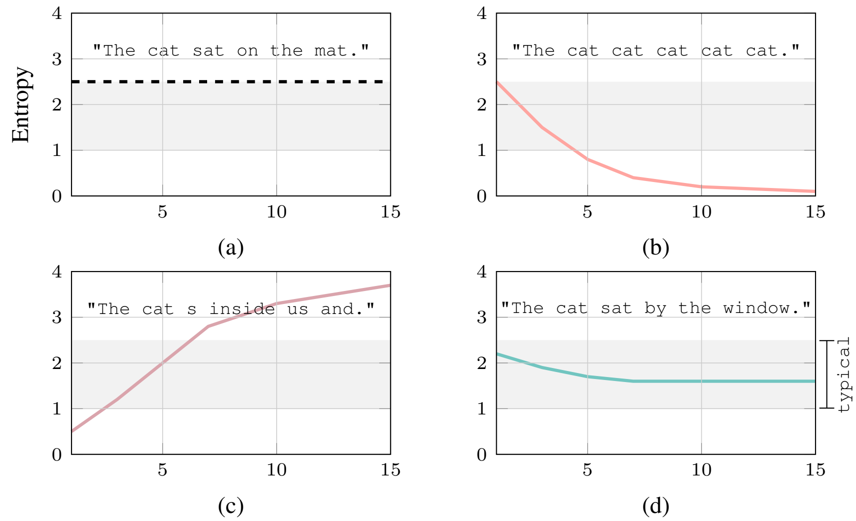

Figure 1: **Entropy trajectories during text generation conditioned on "The cat" prompt.** Figure (a) shows a reference sample with entropy values staying in the "typical" band (gray region). Figure (b) illustrates degeneration into repetition, where entropy collapses below the typical range. Figure (c) shows gibberish, where entropy rises well above the typical range. Figure (d) demonstrates EPIC decoding, which maintains entropy close to the typical band and yields coherent text.

(AEP) (Shannon, 1948), samples from a distribution are overwhelmingly likely to fall in its typical set, with probability mass concentrated near the entropy. In language, typicality is intuitive: highly predictable strings convey little information, while excessively surprising ones tend toward incoherence. Stochastic autoregressive sampling from an LM is unbiased (Koller & Friedman, 2009)[2], and so one might expect it to recover the true typical set. However, even a small divergence between the LM and the true distribution can cause an unbounded mismatch in entropy, and hence in typicality (Braverman et al., 2020). This distortion is further aggravated by mode collapse effects observed in finetuning (O'Mahony et al., 2024), compounding the departure from natural language behavior.

In this work, we introduce EPIC, a hyperparameter-free decoding method that incorporates the entropy of future trajectories into language model decoding. By correcting the entropy mismatch between the learned model and the true distribution, EPIC overcomes the distortions of greedy stepwise heuristics. Our approach combines lazy Gumbel-Max sampling (Hazan et al., 2013; Maddison et al., 2014; Mussmann et al., 2017) with admissible entropy upper bounds to efficiently prune candidate tokens, yielding an exact yet tractable algorithm for sampling from the entropy-aligned next-token distribution. Since entropy is generally intractable to compute (Ahmed et al., 2022; Valiant, 1979a;b), and naïve Monte Carlo estimates suffer from high variance, we further introduce a Rao–Blackwellized estimator that achieves substantially lower variance and improved sample efficiency. We evaluate EPIC on AlpacaEval Creative Writing (Li et al.), GSM8K (Cobbe et al., 2021a), and CNN/DailyMail (Nallapati et al., 2016), spanning creative writing, reasoning, and summarization. Across benchmarks, EPIC consistently outperforms top-p (Holtzman et al., 2020b), top-k (Fan et al., 2018b), and min-p (Minh et al., 2025), producing diverse yet coherent generations.

Contributions In summary, we introduce EPIC, a decoding approach that calibrates, or *aligns*, the entropy of the LM with that of the true language distribution. This is achieved by reweighting the model's next-token distribution with a quantity proportional to the entropy of the distribution over future trajectories. Leveraging lazy Gumbel-Max sampling together with admissible entropy upper bounds, EPIC efficiently prunes candidate tokens while retaining exactness. Monte Carlo estimation of entropy is prohibitively high-variance, so we develop a low-variance, sample-efficient, Rao–Blackwellized entropy estimator. Our experiments across creative writing, reasoning, and summarization benchmarks show EPIC consistently improves the coherence and diversity of generations.

[2]Each step samples exactly from the model's conditional distribution.

2 Related Works

Sampling methods are crucial in controlling the quality and diversity of text generated by LLMs. The choice of sampling strategy directly affects the balance between creativity and coherence, which is critical in many generative tasks. In this section, we review existing approaches and their limitations.

Greedy Decoding and Beam Search Greedy decoding and beam search (Lowerre, 1976; Reddy, 1977; Sutskever et al., 2014; Bahdanau et al., 2015) are deterministic decoding strategies that select the token with the highest probability at each step (Freitag & Al-Onaizan, 2017b). While these methods ensure high-probability token selection, they often lead to repetitive and generic text due to their lack of diversity. Beam search also incurs a significant runtime performance penalty.

Stochastic Sampling Methods Stochastic sampling methods aim to inject diversity into the generated text by introducing randomness in token selection. Temperature scaling adjusts the distribution's sharpness, balancing diversity and coherence (Ackley et al., 1985); however, higher temperatures often lead to incoherent and nonsensical results, limiting its applicability. Top-k sampling selects from the top k most probable tokens, ensuring that only high-probability tokens are considered (Fan et al., 2018a). While it offers a simple way to prevent unlikely tokens from being sampled, it does not adapt dynamically to varying confidence levels across different contexts. Top-p sampling, also known as nucleus sampling, restricts the token pool to those whose cumulative probability exceeds a predefined threshold p (Holtzman et al., 2020a). This method effectively balances quality and diversity by focusing on the "nucleus" of high-probability tokens and dynamically adapts to different contexts. However, at higher temperatures, top-p sampling can still allow low-probability tokens into the sampling pool, leading to incoherent outputs. This trade-off between creativity and coherence at high temperatures is a key limitation that min-p sampling has aimed to address.

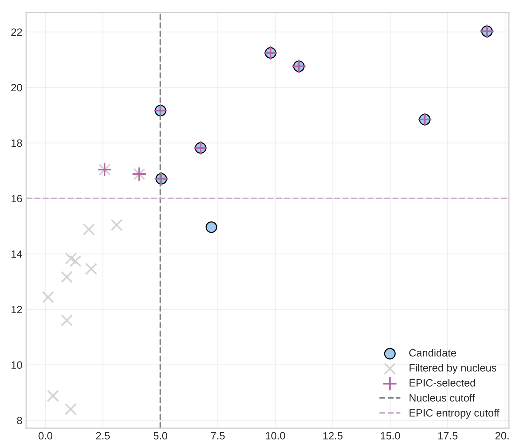

Figure 2: **Interaction of Epic and truncation.** Points denote candidate tokens, represented by probability (→) and lookahead entropy (↑). Gray crosses denote truncated tokens. Tokens considered by truncation algorithms are denoted using blue circles, while Epic-candidate tokens are denoted by violet pluses. The vertical dashed line denotes the probability cutoff, and the horizontal dashed line denotes the entropy cutoff. Epic favors tokens that balance both probability and future uncertainty, avoiding degenerate low-entropy repetitions or incoherent high-entropy expansions.

Entropy-Based Methods Recent work has introduced methods such as entropy-dependent truncation (η-sampling) and mirostat sampling, which attempt to dynamically adjust the sampling pool based on the entropy of the token distribution (Hewitt et al., 2022b; Basu et al., 2021). While entropy- and uncertainty-based approaches show promise in improving text quality, the local heuristics they rely on are often ad hoc, and their parameters tend to be unintuitive and difficult to tune.

Controllable Generation Approaches Our work bears resemblance to prior works that bias LM samples towards satisfying syntactic (Ahmed et al., 2023; 2025b; Zhang et al., 2024; Geh et al., 2024; Lundberg et al., 2024; Willard & Louf, 2023; van Krieken et al., 2025; Koo et al., 2024, *inter alia*), or semantic constraints (Ahmed et al., 2025a; Yidou-Weng et al., 2025; Zhao et al., 2024; Loula et al., 2025; Yang & Klein, 2021; Dathathri et al., 2020; Liu et al., 2021; Beurer-Kellner et al., 2024; Kumar et al., 2022; Qin et al., 2022; Du et al., 2024; Pynadath & Zhang, 2025, *inter alia*), steering LM generations through adjusting next-token probabilities. Conversely, our setting differs in that the constraint is distributional rather than token-level, or even sequence-level: we require that samples be drawn from a restricted subset of the model's support that satisfies certain distributional properties, rather than merely biasing next-token probabilities toward syntactic or semantic criteria.

3 The Typical Set Paradox in LMs

Autoregressive sampling from a language model produces sequences by drawing each token in turn from the model's conditional distribution. This ensures an unbiased generation process whereby the sentences are sampled according to the model's joint distribution. Importantly, such samples overwhelmingly come from *the typical set* of the distribution, rather than from the single most likely sequence. Roughly speaking, although there is a staggering amount of samples that might be generated by a random process, the one actually produced is almost surely from a loosely defined set of samples that all have approximately the same chance of being the one actually realized. And despite individual samples whose probability dominates that of any single outcome in this set, the vast number of realizations in the set all but guarantees the the outcome will come from such set.

To make this more intuitive, consider flipping a biased coin ($p = 0.7$), n times. The most likely sequence is the one where all flips turn up heads, and has probability 0.7^n. By contrast, the typical set consists of sequences with roughly $0.7n$ heads and $0.3n$ tails. Each of these sequences has probability roughly $(0.54)^n$, which is smaller than 0.7^n. However, there are about $\binom{n}{0.7n} \approx 2^{nH(0.7)}$ such sequences. As a result, nearly all of the probability mass lies in this exponentially large typical set, not on the single mode sequence of all heads, even though the latter is individually more likely. Concretely, for $n = 20$, the probability of the all heads sequences is a meager 0.08 compared to the collective probability of flipping 13, 14, or 15 heads, which accounts for over 53% of the mass.

This raises a paradox: *how can we assert that autoregressive sampling draws from the typical set, yet simultaneously argue for a decoding method to bias generation toward that very same set?*

3.1 Typical Set Misalignment under Model Miscalibration

Concretely, let us denote an LM generation of arbitrary length T by $\boldsymbol{y}_{1:T} \coloneqq [y_1 y_2 \dots y_T]$, where y_i is the instantiation of random variable Y_i and takes values from a fixed vocabulary $\mathbb{V} = \{1, \dots, V\}$.

For a distribution p, we denote by $H(p)$ its *entropy*, defined as $H(p) = \mathbb{E}_{\boldsymbol{y}_{1:T} \sim p}\left[-\log p(\boldsymbol{y}_{1:T})\right]$. Additionally, for a distribution q, we denote by $H(p, q)$ the *cross-entropy* between the distributions p and q, defined as $H(p, q) = -\mathbb{E}_{\boldsymbol{y}_{1:T} \sim p}[\log q(\boldsymbol{y}_{1:T})]$, quantifying the average number of bits required to encode samples from p using a code optimized for q. We can also re-write the *cross-entropy* as $H(p, q) = H(p) + D_{\mathrm{KL}}(p \parallel q)$, where $D_{\mathrm{KL}}(p \parallel q)$ denotes the *KL-divergence* between p and q, defined as $D_{\mathrm{KL}}(p \parallel q) = \mathbb{E}_{\boldsymbol{y}_{1:T} \sim p}\left[\log p(\boldsymbol{y}_{1:T}) - \log q(\boldsymbol{y}_{1:T})\right]$, and quantifies the expected extra number of bits needed to encode samples from distribution p when using a code optimized for q.

Note that for a *calibrated* language model, we would hope that $H(p, q) \approx H(q)$ i.e., it's uncertainty regarding its own generations matches the uncertainty it exhibits on data drawn from the true distribution. Furthermore, throughout the paper we will make the assumption that the model is *accurate*

$$D_{\mathrm{KL}}(p \parallel q) = H(p, q) - H(p) < T \cdot \varepsilon, \tag{1}$$

i.e., we assume *uncertainty* due to imperfect knowledge of p can be made arbitrarily small, so that the model's uncertainty primarily reflects the (irreducible) *aleatoric uncertainty* intrinsic to the data.

Prologue. Next, we will show that even under the aforementioned assumption that the epistemic uncertainty due to imperfect knowledge of the distribution can be made arbitrarily small, the mismatch in the entropy between the model and the true underlying distribution can grow unboundedly.

We start by redefining our learned model q as $q \coloneqq (1 - \varepsilon) \cdot q + \varepsilon \cdot \mathcal{U}$, where $\mathcal{U}$ denotes the uniform distribution over $\boldsymbol{y}_{1:T}$. This is to ensure that $q(\boldsymbol{y}_{1:T}) > 0$ for all $\boldsymbol{y}_{1:T}$, as is standard in softmax-based LMs. Now consider the *bounded* function $f = -\log q$. If the learned model q is accurate, in the sense of Equation (1), one might hope that the expected value of f under the true distribution p would be close to the expected value under q, i.e., $\mu_p(f) \approx \mu_q(f)$. However, as we will show next, the model q might be ε-*accurate* and still suffer from entropy miscalibration under long generations.

Lemma 3.1. *[Entropy Miscalibration (Braverman et al., 2020)] Suppose the bound in Equation (1) holds. The calibration error between the true distribution p and the learned model q is bounded as*

$$|H(p, q) - H(q)| \leq \sqrt{2\varepsilon(T + 1)}(T \log M + \log(1/\varepsilon)). \tag{2}$$

The last inequality is the *entropy miscalibration bound*, and it clearly shows that, in the worst case, even a small cross entropy may provide little control over the generations under the learned model. In fact, for $\varepsilon = O(\frac{1}{T})$, which we may hope is an accurate model, the bound turns out to be vacuous.

3.2 Entropy Calibration via Entropy-Aligned Decoding

To address the aforementioned *entropy miscalibration*, we propose explicitly *calibrating the model distribution to the entropy functional*, the idea being to define a family of reweighted distributions

$$q_{t,\alpha}(y_t \mid \boldsymbol{y}_{<t}) \propto q_t(y_t \mid \boldsymbol{y}_{<t}) \exp\left(-\alpha\, H_{t:t+k}(y_t)\right), [3] \tag{3}$$

where $H_{t:t+k}(y_t) := \mathbb{E}_{\mathbf{Y}_{t+1:t+k}\sim q(\cdot|\boldsymbol{y}_{\le t}))}[H(\mathbf{Y}_{t+1:t+k} \mid \boldsymbol{y}_{\le t})]$ denotes the k*-step lookahead entropy* associated with prefix $\boldsymbol{y}_{<t}$, and α is a learnable calibration parameter. Intuitively, the *entropy-aligned* distribution in Equation (3) biases the sampling process according to the estimated lookahead entropy of a prefix: sequences expected to devolve into low-entropy repetitions or high-entropy gibberish are down weighted, whereas those expected to maintain a calibrated entropy are amplified.

We now show that the *entropy-aligned* distribution in Equation (3) is such that the model's k-step lookahead entropy is calibrated to the true distribution without degrading the model's accuracy. Furthermore, α^* is the unique minimizer of $H(p, q_\alpha)$, and can be efficiently computed using bisection.

Lemma 3.2. *[Entropy-Aligned Decoding Lowers CE and Calibrates Entropy] Let p be the true distribution, and q be the learned model. For horizon k, prefix $\boldsymbol{y}_{<t}$, and candidate token y_t, let*

$$H_{t:t+k}(y_t) := \mathbb{E}_{\mathbf{Y}_{t+1:t+k}\sim q(\cdot|\boldsymbol{y}_{\le t})}[H(\mathbf{Y}_{t+1:t+k} \mid \boldsymbol{y}_{\le t})]. \tag{4}$$

Define the entropy-aligned *distribution $q_{t,\alpha}(y_t \mid \boldsymbol{y}_{<t})$ as*

$$q_{t,\alpha}(y_t \mid \boldsymbol{y}_{<t}) \propto q_t(y_t \mid \boldsymbol{y}_{<t}) \exp\left(-\alpha\, H_{t:t+k}(y_t)\right). \tag{5}$$

Let

$$\mu_{\mathcal{D}} := \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{y}<t\sim p, y_t\sim\mathcal{D}(\cdot|\boldsymbol{y}_{<t})}\left[H_{t:t+k}(y_t)\right]. \tag{6}$$

Then there exists $\alpha^* \in \mathbb{R}$ such that

1. $\alpha^* = \operatorname{argmin}_\alpha H(p, q_\alpha)$ *[α^* minimizes CE of the true and entropy-aligned distributions]*
2. $\mu_{q_{\alpha^*}} = \mu_p$ *[The model's entropy is calibrated to the true entropy]*
3. $H(p, q_{\alpha^*}) \le H(p, q)$ *[Entropy Aligned decoding does not worsen accuracy]*

Lemma 3.2 suggests a simple algorithm. Find the α^* that minimizes the cross entropy on a held-out set, or a larger model's generations prompted on the target task. Then, at every timestep, for every token, estimate the lookahead entropy, reweigh the logits by the estimated lookahead entropies, renormalize and sample. The reader might observe that such a simple algorithm is not a very practical one: *it requires that we perform $O(|\mathcal{V}|)$ forward passes of the LM at every time step.* In what follows, *we show how we can compute the cross-entropy and draw exact samples from the entropy-aligned distribution while evaluating the lookahead entropy for only a handful of candidate tokens.*

4 Entropy-Aware Lazy Gumbel-Max for Sublinear Sampling

Background: the Gumbel-Max trick. For any given timestep, let $\{\tilde{y}_i\}_{i=1}^{|\mathcal{V}|}$ denote the *log unnormalized* scores, or *logits*, parameterizing the LM's next-token distribution. One means of sampling from the above distribution is to add i.i.d *Gumbel noise* to each of the logits and take the argmax:

$$y^\star = \operatorname*{argmax}_i\{\tilde{y}_i + G_i\} \sim \text{Categorical}\left(\frac{e^{y_i}}{\sum_j e^{y_j}}\right), \text{where } G_i \sim \text{Gumbel}(0,1). \tag{7}$$

The resulting sample y is precisely distributed according to the *normalized* target distribution (Hazan et al., 2013; Maddison et al., 2014). We will call $\tilde{y}_i + G_i$ the *perturbed scores* or *perturbed logits*.

[3]This recovers *global entropy calibration*, i.e., $q_\alpha(\boldsymbol{y}) \propto \mathrm{e}^{\alpha - \log p(\boldsymbol{y})} \cdot p(\boldsymbol{y}) = p(\boldsymbol{y})^{1+\alpha}$ reducing to *global temperature scaling*, decomposing token-wise as $p_{t,\alpha}(y_t \mid \boldsymbol{y}_{<t}) \propto p_t(y_t \mid \boldsymbol{y}_{<t})^{1+\alpha} \cdot \mathbb{E}_{\boldsymbol{y}_{>t}}[p(\boldsymbol{y}_{>t} \mid \boldsymbol{y}_{<t})^{1+\alpha}]$.

Lazy (sublinear) Gumbel-Max. A naive application of Gumbel-Max samples *all* $|\mathcal{V}|$ Gumbels. A key observation of *lazy Gumbel-Max* (Mussmann et al., 2017) is that *a maximizer of the perturbed logits either posses a large $\tilde{y}_i$, or realizes an unusually large G_i.* Consequently, we can avoid instantiating all Gumbels by (a) selecting the top-k scores $\mathcal{S} = \mathtt{TopK}(\{\tilde{y}_i\}, k)$ and instantiating their Gumbels, and (b) *lazily* drawing *only* Gumbels exceeding the gap to the front-runner in the tail.[4]

Entropy-Aware Lazy Gumbel-Max. Applying Gumbel-Max to Equation (3) reduces to taking

$$y^\star = \operatorname*{argmax}_{i \in \mathcal{V}} \{\tilde{y}_i + G_i - \alpha \cdot H_{t:t+k}(y_i)\}, \tag{8}$$

requiring we compute the lookahead entropy $H_{t:t+k}(y_i)$ for each token $y_i \in \mathcal{V}$, which is infeasible.

Bounding $H_{t:t+k}(\cdot)$ for lazy selection. Taking inspiration from *lazy Gumbel-Max*, we can *bound* $H_{t:t+k}(\cdot)$, and therefore the Entropy-Aligned Perturbed Logits (EARL) in Equation (8). We can then consider *only* the candidates likely to overtake the front-runner. More precisely, for a token y_i,

$$\tilde{y}_i + G_i - \alpha \cdot \mathtt{H}^{\mathtt{U}}_{\mathtt{t}}(\mathtt{k}) \leq s_i \leq \tilde{y}_i + G_i - \alpha \cdot \mathtt{H}^{\mathtt{L}}_{\mathtt{t}}(\mathtt{k}), \tag{9}$$

we can *loosely* bound the entropy from below by 0 and from above by $T \cdot \log|\mathcal{V}|$, the theoretical minimum and maximum, respectively, such that the EARL score of every token y_i is *admissibly* bounded. We can, however, further tighten the above bounds by making use of the result in Lemma 3.1 in tandem with the model error to arrive at much tighter bounds (see Figure 3). We will now show that in light of these bounds, we only need a handful of new entropy evaluations to ascertain the actual winner.

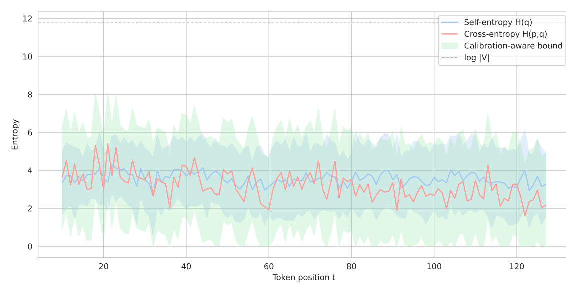

Figure 3: **Tighter Entropy Bounds.** Making use of Lemma 3.1 in tandem with the empirical model error lends to much tighter entropy bounds than theoretical.

Lemma 4.1. *[Expected Entropy Evaluations under Bounded Correction] Let $\{\tilde{y}_i\}_{i=1}^{|\mathcal{V}|}$ denote the* log unnormalized *scores, or* logits*, parameterizing the LM's next-token distribution. Let $\{G_i\}$ denote i.i.d standard Gumbel random variables. let z_i denote the centered perturbed-logits, $z_i = (l_i + g_i) - \log \sum_j \mathrm{e}^{l_j}$. Assume we have an $\hat{H}_i$ such that $|H_{t:t+k}(y_i) - \hat{H}_i| \leq C_t$. Let $w \coloneqq |\alpha| \cdot C_t$. Define plug-in scores $u_i \coloneqq z_i - \alpha\hat{H}_i$, then it is the case that every true score s_i lies in the interval $s_i \in [u_i - w, u_i + w]$. and we have that the expected number of entropy evaluations is $\mathbb{E}[N_{eval}] = \mathrm{e}^{2w}$.*

Instantiating the above lemma with an average per-step error of $\varepsilon = 10^{-3}$, and $\alpha = 0.2$, the maximum we have encountered across domains in our experiments, yields only 4 expected evaluations.

Entropy-Aligned Decoding (EPIC) We are now ready to give a full treatment of our EPIC procedure. To reiterate, our goal is to sample the next token using Gumbel-max from the *entropy-tilted* next-token distribution given in Equation (8), while *avoiding an expensive lookahead over the full vocabulary*. To that end, we start by perturbing the next-token logits with Gumbel noise on Line 3. Next, on Line 4-Line 6 candidates that are $|\alpha| \cdot (\mathtt{H}^{\mathtt{U}}_{\mathtt{t}}(\mathtt{k}) - \mathtt{H}^{\mathtt{L}}_{\mathtt{t}}(\mathtt{k}))$ apart from `frontrunner` are eliminated from the race: such candidates can't overtake the `frontrunner` even if the `frontrunner` only attains the lower bound while the candidates attain the upper bound. If a single candidate remains, it is declared a winner, terminating our procedure (Line 8). Otherwise, we proceed by tightening our bounds (Line 9-Line 12), computing the one-step lookahead entropy and upper-bounding the remaining EARL scores by adding the most favorable value consistent with the bounds, given by $\max\{-\alpha \cdot \mathtt{H}^{\mathtt{U}}_{\mathtt{t}}(\mathtt{k-1}), -\alpha \cdot \mathtt{H}^{\mathtt{L}}_{\mathtt{t}}(\mathtt{k-1})\}$ on Line 11. This retains an admissible upper bound that accounts for the sign of α.[5] Finally, candidates are examined block-wise (Line 13), in descending order of their upperbounded EARL scores, computing the lookahead entropies only as needed on

[4]Sample the expected number of Gumbels m exceeding the gap. Then, uniformly sample a subset of size m from the set $|\mathcal{V}| \setminus \mathcal{S}$. Assign each element in the subset a Gumbel drawn conditionally greater than the bound.

[5]Depending on the domain, α can assume either positive or negative values. Positive α biases decoding towards low-entropy, predictable continuations, while negative α favors high-entropy, diverse continuations.

Algorithm 1: Entropy-Aligned Decoding Logits Processor

Input: Sequence `input_ids`, next-token `scores`, calibration parameter α, and lookahead k
Output: A token y_i sampled from the entropy-aligned $p_\alpha(Y_i|\boldsymbol{y}_{<i}) \propto p(Y_i|\boldsymbol{y}_{<i}) \cdot \mathrm{e}^{-\alpha \cdot H(\mathbf{Y}_{i:i+k})}$

```
/* Get non-zero candidate tokens and corresponding scores */
1  candidates = scores.isfinite().nonzero(as_tuple = True);
2  scores = scores[scores.isfinite()];
/* Perturb the logits using Gumbel noise */
3  gumbel_logits = − log(− log(rand_like(scores))) + scores;
/* Eliminate candidates separated by α × H^U_t from frontrunner as they can't win even if the
   frontrunner only attains the lowerbound entropy H^L_t while they attain the upperbound H^U_t */
4  frontrunner = max(gumbel_logits);
5  mask = gumbel_logits > frontrunner − |α| · (H^U_t(k) − H^L_t(k));
6  gumbel_logits, candidates = gumbel_logits[mask], candidates[mask];
/* Only a single candidate left after pruning */
7  if len(candidates) == 0 then
8      return candidates[0]
/* Tighten bounds by computing step_entropy corresponding to 1-step lookahead entropy */
9  lookaheads = concat(input_ids, candidates);
10 step_entropy = Categorical(model(lookaheads)).entropy();
11 upb_scores = gumbel_logits − α · step_entropy + max{−α · H^U_t(k − 1), −α · H^L_t(k − 1)};
12 sorted_scores, sorted_candidates = sort(upb_scores, candidates);
/* Uncover entropy-weighted scores for the top candidates */
13 for each block B in sorted_candidates do
14     lookaheads = concat(input_ids, B);
15     threshold = sorted_scores[B + 1]  if B + 1 < len(sorted_scores)  else  − ∞;
16     horizn = GetRolloutHorizn(sorted_scores[B], step_entropy[B], α, k, threshold);
17     lookahead_entropies = rb_lookahead_entropy(model, lookaheads, horizn);
18     candidate_scores = sorted_scores[B] + α · lookahead_entropies;
19     if max(candidate_scores) > threshold then
20         return sorted_candidates[argmax(candidate_scores)]
```

a per-block basis. Evaluation stops once a candidate's EARL score exceeds the next-best upper-bounded EARL score, at which point it is guaranteed to be optimal. We adaptively compute the lookahead horizon[6] (Line 16) by estimating the minimum number of rollout steps required for a candidate's optimistic score to exceed the current threshold. The rollout budget is set to the smallest horizon for which the remaining entropy term could close the observed score gap, capped by a global maximum. This strategy allocates lookahead computations only where it can affect the decision, yielding sizable efficiency gains in practice. Our full EPIC procedure is given in Algorithm 1.

Correctness Note that our EPIC procedure detailed above maintains for each candidate index i bounds $\mathrm{LB}_i \leq s_i \leq \mathrm{UB}_i$ with LB_i non-decreasing and UB_i non-increasing as bounds are refined. Furthermore, the stopping condition guarantees the returned $y^\star$ satisfies $s_{y^\star} \geq \mathrm{UB}_j$ for all $j \neq y^\star$, and that no unvisited tail index j can have $H_{t:t+k}(y_i)$ large enough to make $s_j \geq s_{y^\star}$. Therefore the algorithm returns $\arg\max_i s_i$, i.e., an exact sample from Equation (8), and therefore, Equation (3).

5 Experiments

In this section, we explore the efficacy of our EPIC decoding strategy. First off, we qualitatively ascertain that, using a humble lookahead horizon of merely $k = 4$, EPIC is able to track the target entropy achieving almost perfect calibration. Next, we show that such qualitative merits are

[6] We give the definition for the `GetRolloutHorizn` function in the appendix.

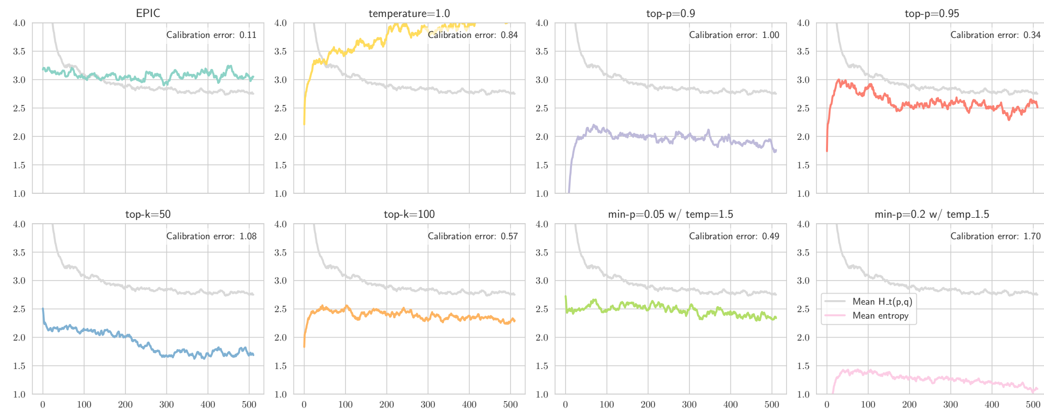

Figure 4: **Calibration of Decoding Methods to Model Cross-Entropy.** We plot the mean conditional entropy of generated sequences as a function of the reference cross-entropy, along with the resulting calibration error for each decoding strategy. EPIC closely tracks the target cross-entropy across the full range, yielding substantially lower calibration error and a markedly smoother trajectory than competing methods. In contrast, standard heuristics such as top-p, top-k, and min-p exhibit systematic bias and high variance, leading to persistent miscalibration despite aggressive tuning.

coupled with meaningful improvements in generation quality, as measured both using an LM-AS-JUDGE (Zheng et al., 2023) as well as automatic quality and diversity metrics. More concretely, we evaluate EPIC on three language generation tasks: abstractive summarization, create story generation, and mathematical reasoning. Our hope in curating such a diverse set of benchmarks is to establish EPIC's ability to preserve the fidelity of the articles to be summarized, generate diverse yet coherent stores in response to creative prompts, and improve the reasoning abilities of LMs. We assess EPIC's performance with respect to several other stochastic decoding strategies: nucleus sampling, top-k sampling, typical decoding, and temperature sampling.[7] For writing tasks, we report the win-rate (WR), the length-controlled win-rate (LC-WR). Additionally, for abstractive summarization we report BERTSCORE-F1 (Zhang et al., 2020) to measure semantic coverage, and for creative writing we report SELF-BLEU (Montahaei et al., 2019) to measure the generation lexical diversity. Lastly, for mathematical reasoning we report the accuracy of predicting the correct answer.

5.1 CALIBRATION TO CROSS-ENTROPY

We evaluate how well different decoding approaches preserve the entropy profile of the base model by measuring their calibration to cross-entropy. For each method, we generate sequences across a range of operating points and compute the mean entropy rate of the resulting samples. We then compare this quantity to the corresponding reference cross-entropy, reporting both the full trajectory and an aggregate calibration error. As shown in Figure 4, EPIC closely tracks the target cross-entropy across the entire range. Its entropy trajectory is smooth and stable, resulting in substantially lower calibration error than all competing approaches. In contrast, standard decoding heuristics such as top-p, top-k, and min-p exhibit systematic bias away from the target entropy, leading to persistent miscalibration even when their hyperparameters are tuned. This behavior highlights a key distinction between EPIC and prior decoding methods: While conventional approaches adjust next-token probabilities locally, EPIC constrains the induced sequence-level distribution. As a result, EPIC maintains global entropy alignment, yielding smoother entropy dynamics and improved calibration.

5.2 QUANTITATIVE EXPERIMENTS

Implementation and Data We use the HuggingFace framework (Wolf et al., 2020), employing their implementations of top-p, top-k, min-p, temperature sampling, and typical decoding. For story generation, we evaluate on the WRITINGPROMPTS dataset (Fan et al., 2018a). For abstractive summarization, we evalaute on the CNN/DAILYMAIL dataset (Nallapati et al., 2016). In a

[7]Temperature sampling is defined as ancestral sampling after renormalization with an annealing term τ

Table 1: **Comparison of the Different Decoding Approaches on the WRITINGPROMPTS and CNN/DAILYMAIL Datasets.** The win rate (WR) and length-controlled win rate are computed using ChatGPT-5, against min-p, and are averaged across 10 seeds. BLEU denotes the SELF-BLEU score, measuring the diversity of each generation against the remaining samples. F1 denotes the BERTSCORE-F1 SCORE, capturing how well the generated output preserves the source content.

	WRITINGPROMPTS			CNN/DAILYMAIL		
Decoding Approach	WR (↑)	LC-WR (↑)	BLEU (↓)	WR (↑)	LC-WR (↑)	F1 (↑)
Top-k ($k = 50$)	54%	54%	1.84	49%	49%	0.16
Top-p ($p = 0.9$)	51%	51%	1.92	52%	52%	0.18
Temperature ($\tau = 1.5$)	0%	0%	**0.69**	2%	2%	−0.24
Typical ($\tau = 0.95$)	43%	43%	1.80	49%	49%	0.17
Min-p ($p = 0.2, \tau = 1.5$)	–	–	2.12	–	–	0.17
EPIC (ours)	**58%**	**58%**	1.55	**56%**	**55%**	**0.19**

preliminary hyperparameter sweep we determine the optimal hyperparameter values on a small validation set using the LM-AS-JUDGE metric. For mathematical reasoning, we evaluate on the GSM8K dataset (Cobbe et al., 2021b). The value of α in EPIC is determined using the procedure outlined in Appendix B.1. All reported metrics are computed on the respective test sets, or subsets thereof.

EVALUATION METRICS

For writing tasks, we evaluate generation quality using LM-AS-JUDGE (Zheng et al., 2023). We fix a *reference* decoding strategy—the best-performing variant of min-p—and compare all other methods against it in pairwise evaluations. For each prompt, the LM judge is presented with a pair of generations, one from the reference method and the other from the candidate method. The LM is then asked to ascertain whether the reference wins, the candidate wins, or there was a tie. All evaluations are conducted anonymously, with generations presented in randomized order to avoid positional or method-specific bias. We repeat this procedure across 10 random seeds using the same underlying set of prompts and generations, reporting the win rate (WR) and the length-controlled win rate (LC-WR), thereby controlling for generation length, in an effort to eliminate length bias.

Furthermore, to assess lexical diversity in creative writing, we report SELF-BLEU (Montahaei et al., 2019) scores, where lower values indicate greater diversity among generated outputs. SELF-BLEU measures the average BLEU score of each generation against the remaining samples, and thus captures mode collapse or excessive similarity across generations. We treat SELF-BLEU as a complementary diagnostic rather than a standalone quality metric seeing as diversity taken alone is not indicative of generation quality. Furthermore, we report BERTSCORE-F1 SCORE (Zhang et al., 2020) on abstractive summarization to measure semantic coverage with respect to the source article.

RESULTS

We first evaluate EPIC on open-ended text generation tasks, including creative writing on the WRITINGPROMPTS dataset and abstractive summarization on CNN/DAILYMAIL. Our results are shown in Table 1. Across both tasks, EPIC consistently outperforms competing decoding strategies when evaluated using an LM-AS-JUDGE. On WRITINGPROMPTS, EPIC achieves the highest preference win-rate among all methods, substantially exceeding standard decoding heuristics such as top-p, top-k, and min-p. Importantly, these gains persist under length-controlled evaluation, indicating that improvements are not driven by verbosity but reflect higher perceived quality of the generated stories. These preference-based results are further corrobo-

Table 2: Decoding approaches evaluated on a random subset of the GSM8k dataset.

Decoding Method	**Accuracy** (↑)
Greedy	79.27 %
Top-k ($k = 50$)	71.95 %
Top-k ($k = 100$)	68.29 %
Top-p ($p = 0.9$)	74.39 %
Top-p ($p = 0.95$)	74.39 %
Typical ($\tau = 0.2$)	73.17 %
Typical ($\tau = 0.95$)	74.39%
min-p ($p = 0.05, \tau = 1.5$)	60.98%
min-p ($p = 0.2, \tau = 1.5$)	75.61%
EPIC (ours)	**85.37%**

rated by diversity metrics. EPIC attains a markedly lower Self-BLEU score than almost all baseline methods, in fact matching the SELF-BLEU achieved by the human reference generations. This suggests that EPIC produces a more diverse set of outputs, avoiding the mode collapse and repetitive patterns commonly observed in entropy-reducing decoding strategies. Note that while the temperature baseline achieves a much lower SELF-BLEU, indicative of higher diversity, it achieves a 0% win-rate compared to Min-p. In fact, upon inspecting the outputs of the temperature baselines, generations almost always devolved into incoherent gibberish. This goes to the point we made earlier: measures of diversity are only meaningful when considered in tandem with measures of generation quality. On CNN/DAILYMAIL, EPIC again achieves the highest preference win-rate under the LM-AS-JUDGE evaluation. In addition, EPIC attains the best BERTSCORE-F1 SCORE among the evaluated methods, indicating improved semantic coverage of the source articles. While the absolute differences in F1 are modest, they are consistent across runs and align with the preference-based judgments, suggesting that EPIC better preserves salient content without over-constraining generation. We note that we make use of the official BERTSCORE implementation[8], with `rescale_with_baseline` = `True`. Consequently, the BERTScore is normalized relative to the expected similarity between unrelated sentences for the underlying encoder. As a result, scores are centered around zero, with negative values indicating worse-than-chance semantic alignment. While this leads to lower absolute values, it yields a calibrated metric that better reflects semantic fidelity and avoids inflated scores due to embedding bias. We also evaluate EPIC on GSM8K, shown in Table 2, to assess its performance on structured mathematical reasoning tasks. EPIC outperforms the strongest baseline approach by an almost 5% absolute accuracy, demonstrating that its advantages extend beyond open-ended generation to settings requiring precise multi-step reasoning.

CONCLUSION

We introduced EPIC, an entropy-aware decoding approach for LMs. EPIC steers language model generations by explicitly aligning the sampling distribution to a target entropy profile, capturing the irreducible (aleatoric) uncertainty of plausible continuations. We show that, unlike existing decoding baselines, EPIC yields sampling distributions that are empirically well-aligned with the entropy of the underlying data distribution. Across creative writing and summarization tasks, EPIC consistently improves LM-AS-JUDGE preference win-rates over widely used decoding strategies: top-k, top-p, min-p typical decoding, and temperature scaling. Importantly, these gains persist under length-controlled evaluation, indicating that EPIC 's improvements are not driven by superficial verbosity effects but by genuinely higher-quality generations. These preference gains are complemented by automatic metrics, which show that EPIC produces more diverse generations in creative settings and more faithful summaries in summarization tasks. We further evaluate EPIC on mathematical reasoning, where it outperforms considered baselines. Through ENTROPY-AWARE LAZY GUMBEL-MAX, EPIC manages to be exact for any given horizon k, while also being efficient and lightweight.

ACKNOWLEDGMENTS

This work is supported in part by the DARPA ANSR program FA8750-23-2-0004 as well as an NSF CAREER award number IIS-2046873. The conclusions presented within this research paper are of the authors and do not reflect the official policy or position of DARPA or the U.S. Government.

REFERENCES

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985. doi: 10.1016/S0364-0213(85)80012-4.

Kareem Ahmed, Eric Wang, Kai-Wei Chang, and Guy Van den Broeck. Neuro-symbolic entropy regularization. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.

Kareem Ahmed, Kai-Wei Chang, and Guy Van den Broeck. A pseudo-semantic loss for autoregressive models with logical constraints. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Kareem Ahmed, Catarina G. Belém, Padhraic Smyth, and Sameer Singh. Semantic probabilistic control of language models. *CoRR*, abs/2505.01954, 2025a.

[8] https://github.com/Tiiiger/bert_score

Kareem Ahmed, Kai-Wei Chang, and Guy Van den Broeck. Controllable generation via locally constrained resampling. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

Rohan Basu, Rishav Saha, and Lav R. Varshney. Mirostat: A neural text decoding algorithm that directly controls perplexity. *arXiv preprint arXiv:2007.14966*, 2021.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. Guiding llms the right way: fast, non-invasive constrained generation. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Mark Braverman, Xinyi Chen, Sham Kakade, Karthik Narasimhan, Cyril Zhang, and Yi Zhang. Calibration, entropy rates, and memory in language models. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168, 2021a.

Kristian Cobbe, Vinay U. Kosaraju, Mohammad Bavarian, Jacky Shi, Dawn Song, Serge Egelman, and John Schulman. Training verifiers to solve math word problems. In *Proceedings of the 9th International Conference on Learning Representations*, 2021b.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2020.

Li Du, Afra Amini, Lucas Torroba Hennigen, Xinyan Velocity Yu, Holden Lee, Jason Eisner, and Ryan Cotterell. Principled gradient-based MCMC for conditional sampling of text. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 889–898. Association for Computational Linguistics, 2018a. doi: 10.18653/v1/P18-1082.

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

Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, August 2017a.

Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 56–60. Association for Computational Linguistics, 2017b. doi: 10.18653/v1/W17-3207.

Renato Lui Geh, Honghua Zhang, Kareem Ahmed, Benjie Wang, and Guy Van den Broeck. Where is the signal in tokenization space? In *Empirical Methods in Natural Language Processing 2024*, 2024.

Tamir Hazan, Subhransu Maji, and Tommi Jaakkola. On sampling from the gibbs distribution with random maximum a-posteriori perturbations. In *NIPS*, 2013.

John Hewitt, Christopher Manning, and Percy Liang. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022a.

John Hewitt, Danqi Zhao, and Christopher D. Manning. Truncation sampling as language model desmoothing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 10839–10853. Association for Computational Linguistics, 2022b. doi: 10.18653/v1/2022.emnlp-main.746.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020a.

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

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Terry Koo, Frederick Liu, and Luheng He. Automata-based constraints for language model decoding, 2024.

Sachin Kumar, Biswajit Paria, and Yulia Tsvetkov. Gradient-based constrained sampling from language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 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)*, 2021.

João Loula, Benjamin LeBrun, Li Du, Ben Lipkin, Clemente Pasti, Gabriel Grand, Tianyu Liu, Yahya Emara, Marjorie Freedman, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Alexander K. Lew, Tim Vieira, and Timothy J. O'Donnell. Syntactic and semantic control of large language models via sequential monte carlo. In *The Thirteenth International Conference on Learning Representations*, 2025.

Bruce T. Lowerre. *The HARPY Speech Recognition System*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, April 1976.

Scott Lundberg, Marco Ribeiro, Richard Edgar, and Harsha-Nori. Guidance: a guidance language for controlling large language models., 2024.

Chris J. Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In *NIPS*, 2014.

Clara Meister, Gian Wiher, Tiago Pimentel, and Ryan Cotterell. 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)*, 2022.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 2023.

Nguyen Nhat Minh, Andrew Baker, Clement Neo, Allen G Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning up the heat: Min-p sampling for creative and coherent LLM outputs. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. Jointly measuring diversity and quality in text generation models. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pp. 90–98. Association for Computational Linguistics, 2019.

Stephen Mussmann, Daniel Lévy, and Stefano Ermon. Fast amortized inference and learning in log-linear models with randomly perturbed nearest neighbor search. 2017.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016.

Laura O'Mahony, Leo Grinsztajn, Hailey Schoelkopf, and Stella Biderman. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024.

Patrick Pynadath and Ruqi Zhang. Controlled LLM decoding via discrete auto-regressive biasing. In *The Thirteenth International Conference on Learning Representations*, 2025.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. COLD decoding: Energy-based constrained text generation with langevin dynamics. In *Advances in Neural Information Processing Systems*, 2022.

D. Raj Reddy. Speech recognition by machine: A review. *Proceedings of the IEEE*, 65(5):501–531, 1977. doi: 10.1109/PROC.1977.10529.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27, 1948.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pp. 3104–3112, 2014.

Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 1979a.

L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 1979b.

Emile van Krieken, Pasquale Minervini, Edoardo Ponti, and Antonio Vergari. Neurosymbolic diffusion models. In *Advances in Neural Information Processing Systems*, 2025.

Brandon T. Willard and Rémi Louf. Efficient guided generation for large language models. *ArXiv*, abs/2307.09702, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2020.

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

Gwen Yidou-Weng, Benjie Wang, and Guy Van den Broeck. Trace back from the future: A probabilistic reasoning approach to controllable language generation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Honghua Zhang, Po-Nien Kung, , Masahiro Yoshida, Nanyun Peng, and Guy Van den Broeck. Adaptable logical control for large language models. In *NeurIPS*, 2024.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, 2021.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations (ICLR)*, 2020.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic inference in language models via twisted sequential monte carlo. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Soheil Feizi Zhuang, Yuxin Wu, Yu Zhuang, Zi Lin, Zhuohan Li, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. 2023.

A Proofs

Lemma 3.1. *[Entropy Miscalibration (Braverman et al., 2020)] Suppose the bound in Equation (1) holds. The calibration error between the true distribution p and the learned model q is bounded as*

$$|H(p, q) - H(q)| \leq \sqrt{2\varepsilon(T+1)}(T \log M + \log(1/\varepsilon)). \tag{2}$$

Proof. Pinsker's Inequality tells us that for any bounded function f with $\|f\|_\infty < B$, we have that

$$|\mathbb{E}_p[f] - \mathbb{E}_q[f]| \leq B\sqrt{2D_{\text{KL}}(p \parallel q)}. \tag{10}$$

We will start by deriving a bound on $D_{\text{KL}}(p \parallel q)$, followed by deriving a bound on B.

Recall our re-definition of q in Section 3.1 as a mixture, where, explicating the dependence on ε

$$q^{(\varepsilon)}(\boldsymbol{y}_{1:T}) = (1-\varepsilon) \cdot q(\boldsymbol{y}_{1:T}) + \varepsilon \cdot M^{-T} \geq (1-\varepsilon) \cdot q(\boldsymbol{y}_{1:T}), \tag{11}$$

and therefore,

$$\frac{1}{q^{(\varepsilon)}(\boldsymbol{y}_{1:T})} \leq \frac{1}{(1-\varepsilon) \cdot q(\boldsymbol{y}_{1:T})}. \tag{12}$$

Taking the logarithm, which is a monotonically increasing function, we get

$$\log \frac{1}{q^{(\varepsilon)}(\boldsymbol{y}_{1:T})} \leq \log \frac{1}{(1-\varepsilon) \cdot q(\boldsymbol{y}_{1:T})} \tag{13}$$

$$= \log \frac{1}{q(\boldsymbol{y}_{1:T})} + \log \frac{1}{1-\varepsilon}. \tag{14}$$

Relaxing $\log \frac{1}{1-\varepsilon}$ using a simple bound valid for $\varepsilon \in (0, \frac{1}{2}]$, we have that

$$\log \frac{1}{1-\varepsilon} = -\log(1-\varepsilon) \leq 2\varepsilon, \tag{15}$$

and therefore,

$$\boxed{\log \frac{1}{q^{(\varepsilon)}(\boldsymbol{y}_{1:T})} \leq \log \frac{1}{q(\boldsymbol{y}_{1:T})} + 2\varepsilon.} \tag{16}$$

Expanding the expression for the KL-divergence and substituting the bound in Equation (16)

$$D_{\text{KL}}(p \parallel q^{(\varepsilon)}) = \mathbb{E}_p[\log \frac{p}{q^{(\varepsilon)}}] = \mathbb{E}_p[\log \frac{1}{q^{(\varepsilon)}}] - \mathbb{E}_p[\log \frac{1}{p}] = \mathbb{E}_p[\log \frac{1}{q}] - \mathbb{E}_p[\log \frac{1}{p}] + 2\varepsilon. \tag{17}$$

Substituting the bound in Equation (1), we get our bound on the KL-divergence term,

$$\boxed{D_{\text{KL}}(p \parallel q^{(\varepsilon)}) \leq T\varepsilon + 2\varepsilon.} \tag{18}$$

Lastly, deriving a value for B, we bound the maximum value of $-\log q^{(\varepsilon)}(\boldsymbol{y}_{1:T})$ as

$$\boxed{-\log q^{(\varepsilon)}(\boldsymbol{y}_{1:T}) \leq -\log(\varepsilon M^{-T}) = \log \frac{M^T}{\varepsilon} = T \log M + \log \frac{1}{\varepsilon}.} \tag{19}$$

Substituting Equation (18) and Equation (19) into Equation (2), we get our desired result. □

Lemma 3.2. *[Entropy-Aligned Decoding Lowers CE and Calibrates Entropy] Let p be the true distribution, and q be the learned model. For horizon k, prefix $\mathbf{y}_{<t}$, and candidate token y_t, let*

$$H_{t:t+k}(y_t) \coloneqq \mathbb{E}_{\mathbf{Y}_{t+1:t+k} \sim q(\cdot|\boldsymbol{y}_{\leq t})}[H(\mathbf{Y}_{t+1:t+k} \mid \boldsymbol{y}_{\leq t})]. \tag{4}$$

Define the entropy-aligned *distribution $q_{t,\alpha}(y_t \mid \boldsymbol{y}_{<t})$ as*

$$q_{t,\alpha}(y_t \mid \boldsymbol{y}_{<t}) \propto q_t(y_t \mid \boldsymbol{y}_{<t}) \exp\left(-\alpha\, H_{t:t+k}(y_t)\right). \tag{5}$$

Let

$$\mu_{\mathcal{D}} \coloneqq \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{y}<t \sim p, y_t \sim \mathcal{D}(\cdot|\boldsymbol{y}_{<t})}\left[H_{t:t+k}(y_t)\right]. \tag{6}$$

Then there exists $\alpha^* \in \mathbb{R}$ such that

1. $\alpha^* = \operatorname{argmin}_\alpha H(p, q_\alpha)$ *[α^* minimizes CE of the true and entropy-aligned distributions]*
2. $\mu_{q_{\alpha^*}} = \mu_p$ *[The model's entropy is calibrated to the true entropy]*
3. $H(p, q_{\alpha^*}) \le H(p, q)$ *[Entropy Aligned decoding does not worsen accuracy]*

Proof. We will write $S_t(y_t) := H_{t:t+k}(y_t)$ for brevity. We will also write the token-wise entropy-aligned distribution, making explicit the dependence on the partition function $Z_t(\alpha; \boldsymbol{y}_{<t})$, as

$$q_{t,\alpha}(y_t \mid \boldsymbol{y}_{<t}) = \frac{q_t(y_t \mid \boldsymbol{y}_{<t}) \exp\left(-\alpha S_t(y_t)\right)}{Z_t(\alpha; \boldsymbol{y}_{<t})}, \text{ where } Z_t(\alpha; \boldsymbol{y}_{<t}) = \sum_{y_t} q_t(y_t \mid \boldsymbol{y}_{<t}) \exp\left(-\alpha S_t(y_t)\right). \tag{20}$$

Taking the derivative of the log-partition function $Z_t(\alpha; \boldsymbol{y}_{<t})$ w.r.t. α, we get

$$\frac{\partial}{\partial \alpha} \log Z_t(\alpha; \boldsymbol{y}_{<t}) = -\mathbb{E}_{y_t \sim q_{t,\alpha}(\cdot \mid \boldsymbol{y}_{<t})}[S_t(y_t)]. \tag{21}$$

Moreover, the second derivative

$$\frac{\partial^2}{\partial^2 \alpha} \log Z_t(\alpha; \boldsymbol{y}_{<t}) = \mathrm{Var}_{y_t \sim q_{t,\alpha}(\cdot \mid \boldsymbol{y}_{<t})}[S_t(y_t)] \ge 0, \tag{22}$$

and therefore, $\log Z_t(\alpha; \boldsymbol{y}_{<t})$ is convex in α. We will also write the sequence-level cross-entropy

$$F(\alpha) := H(p, q_\alpha) = \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{y}_{<t} \sim p} \mathbb{E}_{y_t \sim p(\cdot \mid \boldsymbol{y}_{<t})}[-\log q_{t,\alpha}(y_t \mid \boldsymbol{y}_{<t})] \tag{23}$$

and differentiating w.r.t. α

$$F'(\alpha) = \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{y}_{\le t} \sim p}[S_t(y_t)] + \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{y}_{<t} \sim p}[\frac{\partial}{\partial \alpha} \log Z_t(\alpha; \boldsymbol{y}_{<t})]. \tag{24}$$

Substituting Equation (21) into Equation (24), we get

$$F'(\alpha) = \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{y}_{<t} \sim p}[\mathbb{E}_{y_t \sim p(\cdot \mid \boldsymbol{y}_{<t})} \left[S_t(y_t)\right] - \mathbb{E}_{y_t \sim q_{t,\alpha}(\cdot \mid \boldsymbol{y}_{<t})}[S_t(y_t)]\right] = \mu_p - \mu_{q_\alpha} \tag{25}$$

Since $F(\alpha)$ is a sum of a constant, an affine function, as well as the convex log-partition function, it follows that $F(\alpha)$ is convex. Since F is convex and continuous in α, α^* is a minimizer of F.

Next, having showed $F'(\alpha) = \mu_p - \mu_{q_\alpha}$ Equation (25), and since $F'(\alpha) = 0$, we have $\mu_p - \mu_{q_{\alpha^*}}$, which establishes calibratio. Lastly, since α^* minimizes $F(\alpha)$, then $F(\alpha^*) = H(p, q_{\alpha^*}) \le F(0) = H(p, q_0) = H(p, q)$, which proves that entropy-aligned decoding does not degrade accuracy. □

Lemma 4.1. *[Expected Entropy Evaluations under Bounded Correction] Let $\{\tilde{y}_i\}_{i=1}^{|\mathcal{V}|}$ denote the* log unnormalized *scores, or* logits*, parameterizing the LM's next-token distribution. Let $\{G_i\}$ denote i.i.d standard Gumbel random variables. let z_i denote the centered perturbed-logits, $z_i = (l_i + g_i) - \log \sum_j \mathrm{e}^{l_j}$. Assume we have an $\hat{H}_i$ such that $|H_{t:t+k}(y_i) - \hat{H}_i| \le C_t$. Let $w := |\alpha| \cdot C_t$. Define plug-in scores $u_i := z_i - \alpha \hat{H}_i$, then it is the case that every true score s_i lies in the interval $s_i \in [u_i - w, u_i + w]$. and we have that the expected number of entropy evaluations is $\mathbb{E}\left[N_{\mathit{eval}}\right] = \mathrm{e}^{2w}$.*

Proof. Let $M = \max_i z_i$. Then for any $y \in \mathbb{R}$, the probability of the maximum $M < y$ is given by

$$p(M \le y) = \prod_{i=1}^{n} p(z_i \le y) = \prod_{i=1}^{n} \exp(-\mathrm{e}^{-(y - \log a_i)}) = \exp(-\mathrm{e}^{-y} \sum_i a_i) = \exp(-\mathrm{e}^{-y}), \tag{26}$$

with $a_i = \frac{\mathrm{e}^{l_i}}{\sum_j \mathrm{e}^{l_j}}$ the probability of token y_i. Then, M is Gumbel with density $f_M(y) = \mathrm{e}^{-y} \mathrm{e}^{-\mathrm{e}^{-y}}$.

We denote by T the *count of* z_i such that each z_i is within a w-window of the maximum M, i.e.,

$$T = \#\{i : z_i > M - w\} = \sum_i \mathbb{1}\{z_i > M - w\}. \tag{27}$$

where we're interested in computing the value of T on average. One observation is that, conditional on $M = y$, each non-max indicator $\mathbb{1}\{z_i > M - w\}$ is an independent Bernoulli with probability

$$p_i(y) := p(z_i > y - w) = 1 - \exp(-a_i \mathrm{e}^w \mathrm{e}^{-y}), \tag{28}$$

and consequently,

$$\mathbb{E}[T|M = y] = \sum_i \cdot p_i(y). \tag{29}$$

Averaging out all values of M using the law of total expectation, we get

$$\mathbb{E}[T] = \mathbb{E}_M[\mathbb{E}[T|M]] = \sum_i \mathbb{E}[p_i(M)] = \sum_i \int p(y) f_M(y) dy \tag{30}$$

Using the fact that if $M \sim \text{Gumbel}(0, 1)$ then $X := \mathrm{e}^{-M} \sim \text{Exp}(1)$, we have that for any given z_i

$$\mathbb{E}[p(M)] = \mathbb{E}[1 - \mathrm{e}^{-a_i \mathrm{e}^w \mathrm{e}^{-M}}] = \mathbb{E}[1 - \mathrm{e}^{-a_i \mathrm{e}^w X}] = \int_0^\infty 1 - \mathrm{e}^{-a_i \mathrm{e}^w x} e^{-x} dx = \frac{a_i \mathrm{e}^w}{1 + a_i \mathrm{e}^w}. \tag{31}$$

Therefore,

$$\mathbb{E}[T] = \sum_i \frac{a_i \mathrm{e}^w}{1 + a_i \mathrm{e}^w}. \tag{32}$$

For any given token probability a, define

$$f(a) = \frac{a \mathrm{e}^w}{1 + a \mathrm{e}^w}, \qquad a \in [0, 1] \tag{33}$$

which is concave on the interval $(0, 1]$. Therefore, by Jensen's inequality, we have that

$$\frac{1}{n} \sum_i f(a_i) \leq f(\frac{1}{n} \sum_i a_i) = f(\frac{1}{n}), \tag{34}$$

and,

$$\mathbb{E}[T] = \sum_i f(a_i) \leq n \cdot f(\frac{1}{n}) = \frac{n \mathrm{e}^w}{n + \mathrm{e}^w} < \mathrm{e}^w. \tag{35}$$

That is, for any set of next-token probabilities $\{a_i\}_{i=1}^n$ it holds that $\mathbb{E}[T] < \mathrm{e}^w$.

Moving from a one-sided uncertainty band of width w, where only challengers may gain, to a two-sided band of width $2w$, where the incumbent may decrease and challengers may increase, enlarges the feasible window from $[-w, 0]$ to $[-2w, 0]$, and since the expected number of near- maximal points scales as e^w, the candidate count increases from $\mathrm{e}^w - 1$ to $me^{2w} - 1$, concluding our proof. $\square$

B Implementation Details

B.1 Efficiently Estimating α

Recall from Lemma 3.2 that $\alpha^* = \operatorname{argmin}_\alpha H(p, q_\alpha) = \operatorname{argmin}_\alpha \mathbb{E}_p[-\log q_\alpha(Y)]$. Unfortunately, as written, computing the cross-entropy, and by extension the minimizer of the cross-entropy is computationally intractable due to the dependence on q_α. In what follows, we will show how we can efficiently estimate α by leveraging importance sampling while side stepping the above intractability.

We start by rewriting the cross entropy of the distribution p w.r.t. the entropy-aligned distribution q_α

$$\mathcal{C}(\alpha) := \mathbb{E}_p[-\log q_\alpha(Y)] = \mathbb{E}_p[-\log q(Y)] + \alpha \mathbb{E}_p[H(Y)] + \log Z(\alpha). \tag{36}$$

Since only the last two terms depend on α, by differentiating, we get

$$\mathcal{C}'(\alpha) = \mathbb{E}_p[H(Y)] + \frac{d}{d\alpha} \log Z(\alpha), \tag{37}$$

where

$$\frac{d}{d\alpha} \log Z(\alpha) = -\mathbb{E}_{q_\alpha}[H(Y)], \tag{38}$$

and therefore,

$$\mathcal{C}'(\alpha) = \mathbb{E}_p[H(Y)] - \mathbb{E}_{q_\alpha}[H(Y)]. \tag{39}$$

We therefore have that any minimizer $\alpha^\star$ of $\mathcal{C}(\alpha)$ satisfies the moment-matching condition

$$\mathbb{E}_p[H(Y)] = \mathbb{E}_{q_{\alpha^\star}}[H(Y)]. \tag{40}$$

That is, the expected entropy under the entropy-aligned model matches the true expected entropy.

Moreover,

$$\frac{d}{d\alpha}\mathbb{E}_{q_\alpha}[H(Y)] = -\mathrm{Var}_{q_\alpha}\big(H(Y)\big) \;\leq\; 0, \tag{41}$$

so $\mathbb{E}_{p_\alpha}[H]$ is nonincreasing in α. Furthermore, letting

$$g(\alpha) \;\coloneqq\; \mathbb{E}_p[H(Y)] - \mathbb{E}_{p_\alpha}[H(Y)], \tag{42}$$

we have that $g'(\alpha) \geq 0$. Therefore, the function g is monotone nondecreasing in α. Consequently, the solution to the minimization problem α^* is unique, and can be efficiently found using bisection.

What remains is to show how to efficiently estimate the two expectations in Equation (42):

- $\mu_p \equiv \mathbb{E}_p[H(Y)]$, estimated from a held-out validation dataset $\{y^{(m)}\}_{m=1}^M$ via

$$\widehat{\mu}_p = \frac{1}{M}\sum_{m=1}^{M} H(y^{(m)}), \text{ and;} \tag{43}$$

- $\mu_\alpha \equiv \mathbb{E}_{q_\alpha}[H(Y)]$, which admits the importance-sampling identity

$$\mu_\alpha = \frac{\mathbb{E}_q\big[H(Y)\, e^{-\alpha H(Y)}\big]}{\mathbb{E}_q\big[e^{-\alpha H(Y)}\big]}, \tag{44}$$

estimated using samples $\tilde{y}^{(i)} \sim q$ and importance weights $w_\alpha(\tilde{y}) = e^{-\alpha H(\tilde{y})}$.

B.2 Efficient Entropy Estimation via Rao-Blackwellization

Naïve Monte Carlo estimator for $H_{t:t+k}(\cdot)$ One means of estimating the entropy $H_{t:t+k}(\cdot)$ is to simulate K rollouts $\{\boldsymbol{y}_{t:t+k}^{(i)}\}_{i=1}^K$ of length k from p, and averaging their negative log-probabilities

$$\hat{H}^{\mathrm{MC}}_{t:t+k}(y_t) = -\frac{1}{K}\sum_{i=1}^{K} p(\boldsymbol{y}_{t+1:t+k} \mid \boldsymbol{y}_{<t}). \tag{45}$$

While the above estimator is unbiased for large values of K, it suffers from very high variance.

Rao-Blackwellized estimator By the chain rule of entropy, we can write the lookahead entropy as

$$H\left(\mathbf{Y}_{t+1:t+k} \mid \boldsymbol{y}_{\leq t}\right) = \sum_{j=1}^{k} \mathbb{E}\left[H\left(Y_{t+j} \mid \mathbf{Y}_{\leq t+j-1}\right)\right], \tag{46}$$

where the inner term $H(Y_{t+j} \mid Y_{\leq t+j-1})$ is the *entropy of a one-step predictive distribution* that the model exposes directly from its logits at the corresponding prefix. Equation (46) therefore admits a low-variance *Rao-Blackwellized* estimator that replaces the empirical entropy of joint rollouts by the *exact* conditional entropies, averaged only over the (random) prefixes, which can be written as

$$\hat{H}^{\mathrm{RB}}_{t:t+k}(y_t) = \sum_{j=1}^{k}\frac{1}{K}\sum_{i=1}^{K} H\left(Y_{t+j} \mid \boldsymbol{y}^{(i)}_{\leq t+j-1}\right) \tag{47}$$

By the law of total variance,

$$\mathrm{Var}\left(\hat{H}^{\mathrm{RB}}_{t:t+k}\right) = \mathrm{Var}\big(\mathbb{E}[\hat{H}^{\mathrm{MC}}_{t:t+k} \mid \mathcal{F}]\big) \leq \mathrm{Var}\left(\hat{H}^{\mathrm{MC}}_{t:t+k}\right), \quad \mathcal{F} = \{p(Y_{t+j} \mid \mathbf{Y}^{(i)}_{\leq t+j-1})\}_{i,j}, \tag{48}$$

with strict inequality unless the conditional entropies are constant. Intuitively, Equation (47) collapses the Monte Carlo noise due to *which* token is drawn at step $t+j$ by conditioning on the prefix and using the *analytic* entropy of $p(Y_{t+j} \mid \boldsymbol{y}^{(i)}_{\leq t+j-1})$, for all j. In practice, we found $K \in \{2, 4\}$ to yielding very low variance estimates of the entropy, yielding a fast and stable entropy estimator.