Fine-Tuned Language Models Exhibit Improved Information Conveyance

Yuwei Cheng* **Weiyi Tian*** **Haifeng Xu**
The University of Chicago
{yuweicheng,weiyitian,haifengxu}@uchicago.edu

Abstract

Fine-tuning is often believed to reduce uncertainty and diversity in large language models, but existing analyses overlook output length, a key confounder, and therefore fail to capture how uncertainty is distributed across an entire generation rollout. To address this, we propose *Canopy Entropy* ($\mathrm{CE}^\star$), a measure that views language generation from a tree perspective, where "canopy" represents the space of all possible rollouts, making $\mathrm{CE}^\star$ naturally quantify the *effective size of the generation space*. $\mathrm{CE}^\star$ jointly captures uncertainty in both the output length N and the generated sequence $Y_{1:N}$ – indeed, we show that it equals total Shannon entropy $H(N, Y_{1:N} \mid X)$, where X denotes the prompt. This formulation yields interpretable metrics, including a length–entropy correlation term $\rho(N, r_N)$, where r_N is the entropy rate, quantifying *information conveyance efficiency* by indicating whether longer outputs are more or less informative per token. Empirically, across tasks and model families, we find that fine-tuned models predominantly exhibit more positive correlation $\rho(N, r_N)$, even when total entropy decreases. Furthermore, after controlling for model family, task, prompt, and output-length effects, we find that fine-tuning nearly *triples* the correlation strength between entropy rate and semantic diversity, suggesting that aligned models convert token uncertainty into semantic diversity more efficiently. Overall, these results demonstrate that fine-tuning does not simply reduce uncertainty, but fundamentally reorganizes it into more informative and semantically meaningful generations. Our code is available at `https://github.com/WeiyiTian/canopy-entropy`.

1 Introduction

Large language models (LLMs) have achieved strong performance across tasks such as natural language understanding [4, 16], code generation [5, 30], and mathematical reasoning [24, 25]. A key driver of this success is fine-tuning, including instruction tuning [44, 6] and alignment with human preferences [28, 29]. While fine-tuning substantially improves helpfulness and task performance, a growing body of work suggests that it may also reduce output diversity and compress the effective generation space of language models [42, 47, 22, 45]. Prior studies report that base models exhibit greater randomness and creativity, whereas aligned models produce more concentrated and similar responses across samples [45]. Similar conclusions have been drawn using branching factor [47], lexical diversity [22], and token-level uncertainty measures [1, 41], reinforcing the view that fine-tuning makes language models more deterministic and less diverse.

However, existing analyses suffer from an important limitation – they do not control for output length, despite length being a major confounder in diversity measurement [32]. Classical metrics such as type-token ratio are biased by text length, often making shorter responses appear more diverse

*Equal contribution.

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

[32]. Such metrics that ignore generation length cannot distinguish genuinely diverse outputs from statistical artifacts induced by shorter outputs. This limitation is especially problematic in aligned language models because fine-tuning changes not only what models generate, but also how long they generate. Base models often produce shorter responses, while instruction-tuned models generate longer and more structured outputs [35]. As illustrated in Figure 2, base models exhibit declining entropy rates (average uncertainty per generated token, see Definition 3), suggesting increasingly predictable and redundant outputs. In contrast, fine-tuned models maintain more stable entropy rates, preserving information diversity over longer generations.

More importantly, prior work largely treats diversity as a static property of outputs, without studying how uncertainty evolves along generation trajectories. Previous work on entropy-rate constancy examines how information density changes across positions within human-written text [12, 39]. However, to the best of our knowledge, no existing work analyzes the relationship between LLM generation length and entropy rate—that is, whether longer generations become more informative or increasingly redundant. Yet this interaction is crucial, since fine-tuning learns not only *what* to say, but also *when* to elaborate and *when* to stop. As a result, output length itself becomes part of the learned generation strategy, making diversity comparisons incomplete without accounting for how uncertainty is allocated across trajectories.

Meanwhile, recent studies suggest a more nuanced picture than simple diversity reduction. Fine-tuning can increase certain forms of diversity, such as semantic diversity in code generation, even while reducing lexical variation [34]. These findings suggest that fine-tuning may reorganize uncertainty and generation structure, rather than uniformly suppress diversity, leaving the relationship between uncertainty allocation, semantic diversity, and generation trajectories still poorly understood.

Our contribution. In this work, we revisit diversity in language models through a principled information-theoretic lens. We propose *Canopy Entropy* ($\mathrm{CE}^\star$), an information-theoretic measure that views language generation from a tree perspective. Just as a biological canopy represents the total spread and reach of a tree's branches, our metric views the "canopy" as the set of all potential leaves—or finished rollouts—available to the model. This allows $\mathrm{CE}^\star$ to naturally quantify the effective volume of the generation space by accounting for both how *widely* the model branches and how *deeply* it explores. $\mathrm{CE}^\star$ is introduced to jointly capture output length and content uncertainty, and we show that it admits an exact characterization as the total Shannon entropy $H(N, Y_{1:N} \mid X)$, where X denotes the prompt, N is the generated length, and $Y_{1:N}$ is the generated sequence. Building on this formulation, we derive interpretable metrics and introduce a length–entropy correlation term that measures how uncertainty is allocated across generation trajectories (see section 3). Since $\mathrm{CE}^\star$ exhausts the entire space of rollouts, a key challenge is to estimate it efficiently. Towards that end, we develop Monte Carlo estimators and establish their consistency (see section 4).

Empirically, we evaluate multiple model families across sentence completion, mathematical reasoning, coding, and story generation, while explicitly controlling for output length. Our results reveal three consistent findings: **(i)** fine-tuning generally reduces token-level and trajectory-level uncertainty, although the magnitude of reduction is highly task-dependent; **(ii)** fine-tuning predominantly shifts $\rho(N, r_N)$ from negative toward positive, indicating that longer generations become more informative rather than being redundant as often seen in base models; and **(iii)** entropy rate becomes a substantially stronger predictor of semantic diversity after fine-tuning, suggesting that aligned models utilize uncertainty more efficiently and translate it more effectively into meaningful semantic diversity (see section 5). Overall, our results suggest that fine-tuning does not simply reduce uncertainty or shrink the generation space. Rather, it restructures how uncertainty is allocated across generation trajectories, producing outputs that are more organized, semantically meaningful, and information-efficient.

2 Related work

Length as a fundamental confounder in diversity measurement. Recent work [35] shows that RLHF and instruction tuning systematically increase LLM output length, since longer responses are often positively correlated with human preference scores. This makes output length a critical confounder in diversity evaluation. Metrics that favor shorter generations may therefore unfairly penalize instruct models simply because they produce longer responses. For example, n-gram diversity metrics [23] naturally decrease with sequence length, as longer texts inevitably reuse a finite vocabulary, reducing the ratio of unique n-grams even when the underlying content remains

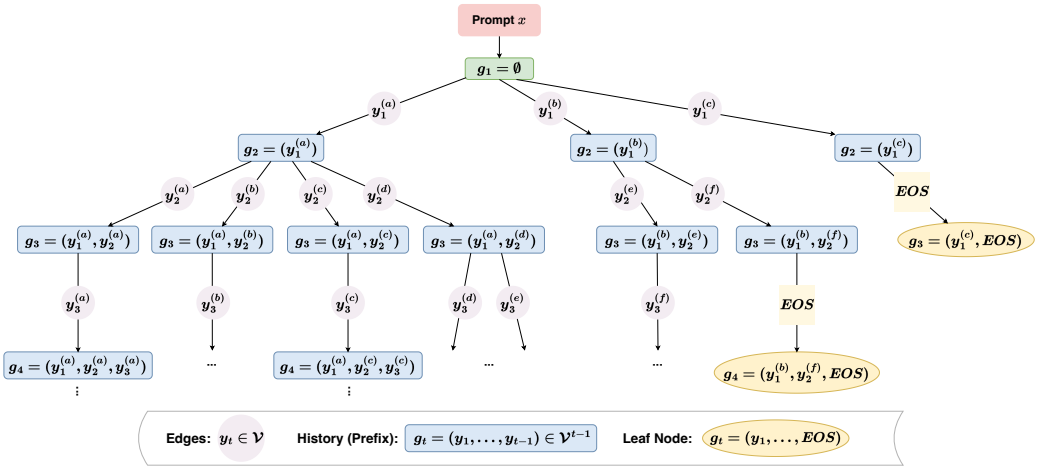

Figure 1: Path-dependent generation tree induced by autoregressive LLMs.

semantically diverse. Embedding-based semantic diversity measures [38] suffer from a related issue: longer generation trajectories must be compressed into fixed-dimensional representations, often through mean pooling, which can dilute semantic diversity. Moreover, many existing diversity metrics rely on heuristic sampling procedures and lack a unified probabilistic foundation [45]. Consequently, current approaches cannot rigorously distinguish whether the increased verbosity of instruct models reflects genuine information gain or merely redundant continuation, motivating the need for a principled length-aware and model-intrinsic measure of generation diversity.

From local branching to global canopy. Conceptualizing autoregressive generation as a stochastic, path-dependent branching tree, Branching Factor (BF) [47] was introduced to quantify the probability concentration induced by model alignment. However, their framework lacks an intuitive heuristic for interpreting the measure and focuses primarily on the *width* of the generation tree (branching) while ignoring its *depth* (length) and the coupling between the two. $\mathrm{CE}^\star$ addresses this gap by providing a model-intrinsic characterization of the generation space by jointly modeling content uncertainty and stochastic stopping behavior. By deriving the BF as a normalized version of $\mathrm{CE}^\star$, we provide a principled theoretical foundation for its use. Crucially, our length–entropy correlation demonstrates that while fine-tuning reduces the absolute BF, it restructures the generation tree to sustain information density across longer paths while translating uncertainty into semantic diversity more efficiently. To our knowledge, this is the first work to provide a unified measure of the global LLM generation space that accounts for both the local branching uncertainty and the stochasticity of trajectory length, while also studying the redistribution of uncertainty across generation trajectories.

3 Methodology

Notations. The Shannon entropy [33] of any discrete random variable (r.v.) Z, with probability $q(z)$ for $z \in \mathcal{Z}$, is $H(Z) = -\sum_{z\in\mathcal{Z}} q(z)\log q(z)$. Let X be another r.v. and X, Z have joint probability $q(z, x)$, then $H(Z|X = x) = -\sum_{z\in\mathcal{Z}} q(z|x)\log q(z|x)$ is the conditional entropy of Z given realized x, whereas $H(Z|X) = \sum_x H(Z|x)q(x)$ is the *expected* conditional entropy of Z.

3.1 Measuring the size of LLM generation space from a tree perspective

Autoregressive generation as a stochastic tree. Autoregressive language models induce a stochastic generation process that can naturally be represented as a rooted, path-dependent tree. Let $X \sim p_X$ denote a prompt sampled from a prompt distribution over a space $\mathcal{X}$, and let $\mathcal{V}$ denote the vocabulary. Conditioned on $X = x$ and a decoding policy (e.g., temperature, top-k, or nucleus sampling), generation proceeds sequentially by sampling tokens from a conditional distribution. Formally, let $Y_1, Y_2, \ldots,$ denote the autoregressive token process generated by the model. For each step $t \geq 1$, define the generation history as $G_t := (Y_1, \ldots, Y_{t-1})$, with $G_1 := \emptyset$. The next-step random variable is denoted by $Z_t \in \mathcal{Z} := \mathcal{V} \cup \{\texttt{EOS}\}$, where EOS denotes termination. The conditional distribution

$q(Z_t \mid G_t, X)$ governs the next-step generation process.[2] Under this construction, each node of the generation tree (see Figure 1) corresponds to a history G_t. Each outgoing edge corresponds to a possible realization of Z_t. A complete generated sequence therefore corresponds to a root-to-leaf trajectory. A key feature of this construction is its path dependence. The next-step distribution q depends on the entire generation history G_t, not merely the most recent token. Thus, two trajectories ending with the same token may evolve very differently if they arise from different histories.

Width and depth of the generation tree. The size of the generation tree is determined by two complementary factors. First, at each history G_t, the spread of $q(\cdot \mid G_t, X)$ determines the number of plausible continuations, capturing the local branching behavior (*width*) of the tree. Second, generation proceeds for a random number of steps before termination, inducing variability in trajectory depth. We define the stopping time $N := \inf\{t \geq 1 : Z_t = \texttt{EOS}\}$, which represents the generation length (*depth*). Under this formulation, the stochastic process jointly models both continuation uncertainty and stopping behavior, together determining the overall size of the generation space.

A unified measure of generation space. To quantify local uncertainty, we use Shannon entropy [33]. For a given prompt $X = x$ and a realized generation history $G_t = g_t$, define

$$\widetilde{H}(g_t \mid x) := H(Z_t \mid G_t = g_t, X = x) = -\sum_{z \in \mathcal{Z}} q(z \mid g_t, x) \log q(z \mid g_t, x). \tag{1}$$

This quantity measures uncertainty via entropy in the immediate next generation step, including both continuation-token selection and termination. Note that this quantity summarizes only the uncertainty of the next-step generation conditioned on a realized history g_t and prompt x. In the following, we slightly overload the notation by also using uppercase G_t and X to denote realized histories and prompts whenever the distinction between random variables and their realizations is clear from context. To jointly account for local branching diversity and random trajectory length, we introduce the following uncertainty measure.

Definition 1 (Canopy Entropy)**.** *We define* Canopy Entropy *as* $\mathrm{CE}^\star := \mathbb{E}\left[\sum_{t=1}^{N(x)} \widetilde{H}(g_t \mid x)\right]$. *More formally,* $\mathrm{CE}^\star = \mathbb{E}_{x, \{g_t\}_{t=1}^{N(x)}}\left[\sum_{t=1}^{N(x)} H(Z_t \mid G_t = g_t, X = x)\right]$, *where the* $\mathbb{E}$ *is over the randomness of prompt* x, $N(x)$ *(the length), and* $\{g_t\}_{t=1}^{N(x)}$ *(all possible partial histories of rollouts).*

That is, the quantity $\mathrm{CE}^\star$ aggregates entropy-based uncertainty along generation trajectories and across prompts. A single generated sequence corresponds to a specific traversal from the tree root (prompt X) to a leaf (EOS termination). The summation $\sum_{t=1}^{N(X)}$ thus represents the total uncertainty accumulated along that entire path. By taking the expectation over all trajectories, $\mathrm{CE}^\star$ captures the aggregate "canopy" of the model's output distribution across all possible rollouts. As such, it provides a unified measure of generation space by accounting for both *local branching width* (the diversity of choices at each node) and *trajectory depth* (the variability in generation length).

$\mathrm{CE}^\star$ as the joint entropy of length and content. Besides being an intuitive measure, the above construction turns out to also admit a clean information-theoretic characterization. In particular, the cumulative uncertainty along a trajectory can be interpreted as the total entropy of the generation process. For reference, we call $H(N, Y_{1:N}|X)$ the *total uncertainty*, $H(N|X)$ the *length uncertainty*, and $H(Y_{1:N} \mid N, X)$ the *content uncertainty*. The following theorem formalizes the connection.

Theorem 1 (Equivalence of $\mathrm{CE}^\star$ and total uncertainty)**.** $\mathrm{CE}^\star = H(N, Y_{1:N} \mid X)$.

Theorem 1 specializes classical information-theoretic results on randomly stopped stochastic processes [3, 8] to autoregressive language generation, showing that the Canopy Entropy coincides with the total uncertainty of the generation process (see subsection A.2 for a detailed proof). By the entropy chain rule, $H(N, Y_{1:N} \mid X) = H(N \mid X) + H(Y_{1:N} \mid N, X)$, decomposing total uncertainty into two components. The *length uncertainty* $H(N \mid X)$ captures how unpredictable the stopping time is, while the *content uncertainty* $H(Y_{1:N} \mid N, X)$ captures how diverse the generated sequences are once the length is fixed. This decomposition aligns naturally with the tree perspective: length uncertainty corresponds to variability in the depth of trajectories, while content uncertainty corresponds to the diversity of paths of a given length. We can interpret $\exp(\mathrm{CE}^\star)$ as the effective number of plausible full sequences, analogous to how perplexity represents the effective branching at the token level.

[2] This formulation is equivalent to a two-stage stochastic decision process. Specifically, the model first makes a binary decision between STOP (terminate) and CONT (continue). Conditional on CONT, it then samples the next token from the vocabulary $\mathcal{V}$. We provide a formal proof of this equivalence in subsection A.1.

3.2 Two normalized variants of Canopy Entropy

While $\mathrm{CE}^\star = H(N, Y_{1:N} \mid X)$ provides a clean characterization of the generation space, it aggregates uncertainty across both sequence lengths and content diversity, making its direct interpretation difficult. To obtain more interpretable metrics, we introduce two normalized variants of $\mathrm{CE}^\star$, respectively coined *Generation Perplexity* and *Branching Factor* (as introduced in [47]). Intuitively, the former corresponds to the token-level normalization of $\mathrm{CE}^\star$, whereas the latter is trajectory-level normalization. In both definitions, the $\exp$ function converts entropy uncertainty to effective diversity.

Definition 2 (GenPPL)**.** *Generation Perplexity is defined as* $\mathrm{GenPPL} := \exp\left(\frac{\mathrm{CE}^\star}{\mathbb{E}[N]}\right)$*, which is equivalent to* $\exp\left(\frac{\sum_{X\sim p_X} H(N, Y_{1:N}|X=x)}{\sum_{X\sim p_X} \mathbb{E}[N|X=x]}\right)$.

Definition 3 (BF, [47])**.** *Branching Factor is defined as* $\mathrm{BF} := \exp\left(\mathbb{E}_X \mathbb{E}_N[r_N \mid X = x]\right)$*, where* $r_N := \frac{H(Y_{1:N}|N=n, X=x)}{n}$ *is the entropy rate (i.e., per-token content uncertainty).*[3]

Invoking Theorem 1, we observe that the only difference between Canopy Entropy $\mathrm{CE}^\star$ and GenPPL or BF is that – excluding the "exp" function – $\mathrm{CE}^\star$ does not have the normalization term $1/(\sum_{X\sim p_X} \mathbb{E}[N \mid X = x])$ or $1/n$. It is worthwhile to think more deeply about the difference between GenPPL and BF, due to their different normalization factors, which will be prominent when the generated sequence length N can differ a lot. In such a case, GenPPL *treats each token equally*, regardless of whether it comes from a short or long sequence, through a "global" normalization factor $1/(\sum_{X\sim p_X} \mathbb{E}[N \mid X = x])$. However, BF as the expectation of r_N *treats each trajectory equally* and consequently will discount the influence of tokens in long sequences much more than those in short sequences. Therefore, at a high level, GenPPL reflects the diversity of a *typical token* in the generation whereas BF reflects the diversity of a *typical trajectory*. For further intuition, we refer readers to subsection B.1 for a concrete example illustrating the distinction between GenPPL and BF.

GenPPL and BF together provide a more complete view of generation behavior across tasks. In particular, we observe that structured tasks such as mathematical reasoning or coding are better characterized by token-level uncertainty (GenPPL), whereas open-ended tasks like story generation are better captured by trajectory-level diversity (BF). Our framework provides a principled formulation of GenPPL as a generative analogue of perplexity [19] for variable-length processes, and offers a unified information-theoretic interpretation of BF, initially proposed by Yang et al. [47], showing that both arise naturally as normalized variants of Canopy Entropy. See Appendix B.2 for guidance on when to use BF or GenPPL, and Appendix B.3 for a comparison with perplexity.

3.3 The length–entropy correlation: how entropy evolves with generation length

Aggregate measures such as Canopy Entropy, GenPPL, and BF quantify the overall magnitude of uncertainty. However, they fail to capture how uncertainty evolves across generation rollouts or how effectively uncertainty is translated into meaningful variation as generation length increases. Whether longer generations remain information-rich or become progressively redundant [35, 32] remains poorly understood despite rich previous studies on the diversity of language generation [38, 23, 11]. Understanding this interaction is crucial for characterizing the *information-conveying efficiency* of language models—namely, how much uncertainty is maintained per token as generation progresses.

To study this phenomenon, we introduce the prompt-controlled correlation $\rho(N, r_N)$ between the length N and entropy rate r_N, defined below.

Definition 4 (The length–entropy correlation)**.** *For each prompt, the length–entropy correlation is defined as*

$$\rho(N, r_N) = \mathbb{E}_{X\sim p_X}\left\{\texttt{Corr}(N, r_N \mid X = x)\right\},\ \textit{where}\ r_N = \frac{H(Y_{1:N} \mid N = n, X = x)}{n}$$

is the entropy rate and `Corr` *is any standard correlation notion from the set {Pearson correlation, Spearman correlation, Kendall correlation}.*

A positive correlation $\rho(N, r_N)$ indicates that longer sequences remain informative per token, while a negative correlation indicates that longer outputs become increasingly predictable. This quantity

[3] In information theory, the entropy rate quantifies the average amount of information produced per generated token by a stochastic process, and can be interpreted as a measure of information density or non-redundancy.

characterizes uncertainty allocation along trajectories and is later used in section 5 to compare base and fine-tuned models across tasks and model families. In our evaluation, we observe broadly consistent trends across the three standard correlation notions as mentioned in the definition above, which illustrates the robustness of the discovered correlations among generation length and entropy rate. Notably, the correlation $\rho(N, r_N)$ is evaluated over the randomness of rollouts drawn from an exponentially large space; we discuss efficient estimation methods in the section below, with full details deferred to subsection A.6.

4 Efficient and consistent estimation of $\mathrm{CE}^\star$

The previous section introduced $\mathrm{CE}^\star = H(N, Y_{1:N} \mid X)$ and its derived metrics such as GenPPL, BF and length–entropy correlation. However, efficiently computing these metrics in practice is challenging since they are expectations over exponentially many trajectories [17, 47]. This section develops computationally tractable approaches to estimate $\mathrm{CE}^\star$ which are also generalizable to estimating the derived metrics. In particular, we introduce a Monte Carlo estimator under a maximum length constraint $T_{\max}$, and analyze the consistency of this estimator.

For a random prompt X, let $N(X)$ be the generation's (random) stopping time. Given a maximum length $T_{\max}$, define the truncated stopping time $N_{\max}(X) := \min\{N(X), T_{\max}\}$. The corresponding truncated functional is $\mathrm{CE}^\star_{\max} := \mathbb{E}\left[\sum_{t=1}^{N_{\max}(X)} \widetilde{H}(G_t \mid X)\right]$. Now suppose $x_1, \ldots, x_P$ are sampled i.i.d. from p_X, the prompt distribution. For each prompt x_p, we generate M independent rollouts. For the i-th rollout, define the truncated cumulative local entropy by $S^{(p,i)}_{\max} := \sum_{t=1}^{N^{(p,i)}_{\max}} \widetilde{H}(G^{(p,i)}_t \mid x_p)$, which measures the total branching uncertainty along the generated trajectory up to truncation. We then define the Monte Carlo estimator (see algorithm 1 for more detail)

$$\widehat{\mathrm{CE}}^\star_{\max,P,M} := \frac{1}{PM} \sum_{p=1}^{P} \sum_{i=1}^{M} S^{(p,i)}_{\max}.$$

The estimator's convergence and consistency require minor technical assumptions. First, let $g(T_{\max}) = \mathrm{CE}^\star - \mathrm{CE}^\star_{\max}$ denote the truncation bias. Naturally, it always satisfies $g(T_{\max}) \geq 0$. Convergence requires **diminishing truncation bias** — i.e., $g(T_{\max}) \to 0$ as $T_{\max} \to \infty$. Second, in our estimation, $T_{\max}$ increases as parameter P, M increases. Convergence requires **modest truncation length** — i.e., the length cap $T_{\max}(P, M)$ is picked such that $\nu^2_{\max}(\frac{1}{P} + \frac{1}{PM}) \to 0$ as $T_{\max} = T_{\max}(P, M) \to \infty$, where $\nu^2_{\max} := \mathbb{E}\left[N_{\max}(X)^2\right]$ is the second raw moment of the truncated stopping time. Both conditions are easily satisfied when the tail of N is not too heavy, which we observe in our empirical evaluation (see Figure 3). The following theorem characterizes the convergence of $\widehat{\mathrm{CE}}^\star_{\max,P,M}$ under these technical conditions.

Theorem 2 (Consistency and non-asymptotic error bounds of the Monte Carlo estimator)**.** *Assume diminishing truncation bias $g(T_{\max})$ and modest truncation length. Then $\widehat{\mathrm{CE}}^\star_{\max,P,M}$ converges to $\mathrm{CE}^\star$ in probability when $P, M \to \infty$. Moreover, $\left|\widehat{\mathrm{CE}}^\star_{\max,P,M} - \mathrm{CE}^\star\right| = O_{\mathbb{P}}\left(\nu_{\max}\sqrt{\frac{1}{P} + \frac{1}{PM}} + g(T_{\max})\right)$, where $\nu^2_{\max} := \mathbb{E}\left[N_{\max}(X)^2\right]$.*

Theorem 2 shows that the estimator $\widehat{\mathrm{CE}}^\star_{\max,P,M}$ is consistent, with estimation error naturally decomposing into a Monte Carlo error term and a truncation bias term. The Monte Carlo error is of order $O_{\mathbb{P}}\left(\frac{1}{\sqrt{P}} + \frac{1}{\sqrt{PM}}\right)$, arising from sampling P prompts and M rollouts per prompt.

The second component is a truncation bias $g(T_{\max})$, which captures the bias introduced by truncating trajectories at length $T_{\max}$ instead of allowing full generations. Empirically, we observe that the generation length distribution of LLMs typically exhibits exponential decay across tasks and model architectures (see Figure 3). This implies that long trajectories are increasingly rare, so the truncation bias $g(T_{\max})$ decays rapidly with $T_{\max}$. We defer the proof of Theorem 2 to Appendix A.5.

Notably, the same Monte Carlo framework extends to the estimation of GenPPL, BF, and prompt-controlled correlation. Their consistency follows by the continuous mapping theorem [26] together with mild integrability and truncation-bias conditions, which are empirically supported by the

observed low truncation rates and rapidly decaying active-rollout distributions (see Appendix A.6 for detailed proofs and discussions).

Remark 4.1 (Connecting truncation bias to the distribution tail of N)**.** *The truncation bias can be expressed as*

$$g(T_{\max}) = \mathrm{CE}^{\star} - \mathrm{CE}^{\star}_{\max} = \mathbb{E}\left[\sum_{t=T_{\max}+1}^{N(X)} \widetilde{H}(G_t \mid X)\mathbf{1}\{N(X) > T_{\max}\}\right]. \tag{2}$$

Let $H_{\max}$ denote the entropy upper bound, i.e., $\widetilde{H}(G_t \mid X) \le H_{\max}$ for any G_t. Then we have $g(T_{\max}) \le H_{\max}\mathbb{E}_{X\sim p_X}\left[\mathbb{E}\left[(N(X)-T_{\max})_+|X\right]\right]$ by the tail-sum formula $\mathbb{E}\left[(N(x)-T_{\max})_+|X=x\right] = \sum_{t=T_{\max}+1}^{\infty}\mathbb{P}(N(x) \ge t \mid X = x)$. Therefore, $H_{\max}\mathbb{E}_{X\sim p_X}\left[\sum_{t=T_{\max}+1}^{\infty}\mathbb{P}(N(X)\ge t \mid X)\right]$, with $H_{\max} := \log|\mathcal{Z}|$, is a natural upper bound of $g(T_{\max})$, showing that the truncation bias is governed by the prompt-averaged tail behavior of the stopping time. We refer readers to subsection A.4 for examples of prompt-averaged bias decay rates.

5 Empirical studies

In this section, we present our experimental results, organized into two complementary parts. First, we examine the global uncertainty structure of the LLM generation space using the metrics introduced in section 3, including $\mathrm{CE}^{\star}$, GenPPL, BF, and length–entropy correlation. Second, building on this analysis, we investigate how uncertainty is translated into semantic diversity through a regression framework that links entropy rate to semantic diversity. We evaluate three model families—LLaMA-3.1-8B [13], Qwen3-8B [46], and Gemma-3-12B [37]—testing both their pre-trained (base) and instruction-tuned (instruct) versions. We evaluate these models on four tasks spanning a spectrum from highly structured settings, including mathematical reasoning [7], coding [49], and sentence completion [48], to more open-ended creative story generation [18]. For each task, we sample $P = 100$ prompts, and for each sampled prompt X_p, we generate $M = 100$ independent rollouts using stochastic decoding with temperature equal to 1 and top-$k = 100$ (renormalized next-token probabilities),[4] where each rollout produces a sequence $Y_{1:N}$ with stopping time N up to a maximum of 4000 new tokens, i.e. $T_{\max} = 4000$ (see subsection C.1 for detailed description). To ensure consistent evaluation across model families, we disable the thinking mode in Qwen3-8B Instruct.

5.1 Global effects of fine-tuning on uncertainty

Across all model families and tasks, we observe three consistent patterns.

Fine-tuning reduces both token-level and trajectory-level normalized uncertainty. Instruction fine-tuning consistently reduces both token-level (measured by GenPPL) and trajectory-level (measured by BF) normalized uncertainty, as evidenced by systematic declines in GenPPL and BF across nearly all models and tasks (see Table 1). Nevertheless, the magnitude of this reduction varies substantially across settings. For example, Qwen3-8B on mathematical reasoning exhibits only a mild decrease of roughly 5%, whereas many other model–task combinations show much larger reductions, often in the range of 60%–80%. These results suggest that fine-tuning contracts the effective generation space at both the token and trajectory levels, producing outputs that are more concentrated and deterministic. At the same time, the heterogeneous magnitude of the reduction suggests that alignment does not impose a uniform compression of uncertainty. Instead, the extent of contraction depends on the underlying task structure and model behavior, a phenomenon we discuss in the following paragraph. Overall, our findings are consistent with prior observations that alignment reduces diversity and increases probability concentration [42, 47, 22, 45], with reductions in branching factor broadly comparable in magnitude to those reported by Yang et al. [47].

The effect of fine-tuning on canopy entropy is *task-dependent* and could exhibit opposite trends. In structured domains such as math, coding, and sentence completion, $\mathrm{CE}^{\star}$ decreases significantly after fine-tuning. In contrast, story generation exhibits a different pattern: Qwen3-8B shows a slight increase of roughly 2%, while Gemma-3-12B shows a much larger increase of about 29%

[4] Specifically, the one step local entropy $\tilde{H}(G_t|X)$ is computed as $-\sum_{z\in\mathcal{Z}_{100}} q(z|G_t,X)\log q(z|G_t,X)$, where $\mathcal{Z}_{100}$ is the set of the top-100 tokens (including `EOS` if present) and q is the renormalized distribution over the sampled subset.

Table 1: Task-dependent changes in generation-space uncertainty after instruction tuning across different model families. We report three complementary uncertainty measures: GenPPL, BF, and $\mathrm{CE}^\star$ (see subsection A.6). For each task and model family, we compare the base and instruct variants and report the relative percentage change after instruction tuning. Standard errors (SE) are estimated via prompt-cluster bootstrapping (2K iterations, resampled with replacement) as in subsection C.4. Blue cells indicate reductions in uncertainty after instruction tuning, while red cells indicate increases. Darker color intensity corresponds to larger magnitude changes.

Task	Type	GenPPL			BF			$\mathrm{CE}^\star$		
		Qwen3-8B	Llama-3.1-8B	Gemma-3-12B	Qwen3-8B	Llama-3.1-8B	Gemma-3-12B	Qwen3-8B	Llama-3.1-8B	Gemma-3-12B
Math	Base	1.23 ± 0.0052	3.57 ± 0.041	4.56 ± 0.052	1.23 ± 0.0064	5.31 ± 0.12	4.98 ± 0.055	55.7 ± 2.0	458 ± 18	487 ± 11
	Instruct	1.17 ± 0.0067	1.76 ± 0.076	1.16 ± 0.0091	1.16 ± 0.0057	1.51 ± 0.025	1.15 ± 0.0052	40.5 ± 2.5	108 ± 12	56.7 ± 4.7
	% Change	$-4.44 \pm 0.47\%$	$-50.8 \pm 2.3\%$	$-74.5 \pm 0.33\%$	$-5.76 \pm 0.38\%$	$-71.5 \pm 0.71\%$	$-76.9 \pm 0.24\%$	$-27.3 \pm 3.3\%$	$-76.3 \pm 2.8\%$	$-88.4 \pm 1.0\%$
Coding	Base	1.68 ± 0.017	3.16 ± 0.055	2.60 ± 0.041	2.08 ± 0.034	4.96 ± 0.057	3.24 ± 0.038	207 ± 4.8	326 ± 13	492 ± 25
	Instruct	1.23 ± 0.0053	1.27 ± 0.0075	1.04 ± 0.0023	1.21 ± 0.0061	1.27 ± 0.0070	1.04 ± 0.0018	81.2 ± 4.0	126 ± 3.9	16.2 ± 1.0
	% Change	$-27.1 \pm 0.79\%$	$-59.6 \pm 0.71\%$	$-59.9 \pm 0.64\%$	$-41.9 \pm 0.99\%$	$-74.4 \pm 0.33\%$	$-67.8 \pm 0.38\%$	$-60.7 \pm 1.6\%$	$-61.4 \pm 1.7\%$	$-96.7 \pm 0.28\%$
Sentence Completion	Base	4.77 ± 0.16	6.73 ± 0.12	8.81 ± 0.11	4.20 ± 0.11	7.80 ± 0.15	9.49 ± 0.11	826 ± 55	969 ± 40	1084 ± 36
	Instruct	1.68 ± 0.025	2.32 ± 0.060	1.78 ± 0.019	1.55 ± 0.018	2.53 ± 0.042	1.71 ± 0.024	112 ± 11	188 ± 9.4	513 ± 29
	% Change	$-64.9 \pm 1.1\%$	$-65.6 \pm 1.1\%$	$-79.8 \pm 0.30\%$	$-63.0 \pm 0.88\%$	$-67.6 \pm 0.81\%$	$-82.0 \pm 0.30\%$	$-86.5 \pm 1.0\%$	$-80.6 \pm 1.2\%$	$-52.6 \pm 2.9\%$
Story Generation	Base	3.57 ± 0.11	7.40 ± 0.071	7.98 ± 0.062	4.03 ± 0.092	8.24 ± 0.073	8.68 ± 0.070	445 ± 20	1320 ± 41	692 ± 23
	Instruct	2.28 ± 0.037	2.93 ± 0.083	2.22 ± 0.023	2.11 ± 0.039	3.07 ± 0.077	2.16 ± 0.027	454 ± 28	471 ± 19	892 ± 39
	% Change	$-36.0 \pm 1.3\%$	$-60.4 \pm 1.2\%$	$-72.2 \pm 0.28\%$	$-47.6 \pm 1.3\%$	$-62.7 \pm 1.1\%$	$-75.1 \pm 0.30\%$	$1.92 \pm 3.5\%$	$-64.3 \pm 1.3\%$	$29.0 \pm 8.0\%$

(see Table 1). This suggests that fine-tuning does not uniformly reduce canopy entropy, but instead behaves differently in constrained versus open-ended generation settings.

For deterministic tasks such as math and coding, the valid generation space is naturally constrained, so fine-tuning acts as a pruning force that collapses the model's generation space into a small set of correct trajectories. Consequently, both stopping-time variance and local token-level entropy are significantly reduced. Meanwhile, fine-tuned models generally exhibit a rightward shift in sequence length distributions across tasks (see Figure 3), reflecting a tendency toward longer and more elaborative generations. In story generation, this contrast is further accentuated by the fact that base models frequently fall into "completion traps", treating prompts as short text continuation tasks rather than directly answering the underlying questions (see subsection C.2).

These differences in generation behavior help explain why the effect of instruction tuning on $\mathrm{CE}^\star$ is task-dependent. Although fine-tuning reduces local branching diversity, it can also increase stopping-time uncertainty and sustain generation over much deeper trajectories. As shown in Figure 2 and Figure 3, while base models have higher local entropy rates (*width*), they may lack sufficient trajectory length (*depth*) to accumulate a large Canopy Entropy ($\mathrm{CE}^\star$). Since $\mathrm{CE}^\star$ acts as a measure of the total *volume* of the generation tree by accumulating uncertainty across the entire rollout – including stopping behavior, the increase in trajectory depth can outweigh the reduction in local branching.

These findings suggest that fine-tuning changes not only the magnitude of uncertainty, but also how it is allocated across generation trajectories, particularly through its interaction with generation length and stopping behavior.

Fine-tuning predominantly increases length–entropy correlation. To better understand how uncertainty is distributed across trajectories, we analyze the relationship between output length N and entropy rate r_N. Empirically, fine-tuning consistently shifts the correlation (see Definition 4) between N and r_N from negative toward positive values in structured tasks, while generally making the correlation less negative in open-ended tasks such as story generation. Table 2 shows the concrete results under Pearson correlation, with positive shifts in 11 of the 12 model–task comparisons. Similar results based on Spearman correlation and Kendall's τ exhibit the same trends in Table 3.

This reveals a fundamental change in generation behavior. In base models, longer outputs are often associated with lower entropy rates, meaning later tokens become increasingly redundant and contribute little new information. In contrast, fine-tuned models generally exhibit much higher length–entropy correlation: longer generations tend to sustain higher entropy rates and remain informative throughout the trajectory (see Figure 2).

The distinction is especially important in the presence of length confounding. As shown in Figure 3, fine-tuned models generally produce longer outputs than their base counterparts, consistent with observations in prior work [35]. Under conventional diversity metrics that do not explicitly account for length [38, 23], these longer generations can artificially appear less diverse due to length-dependent

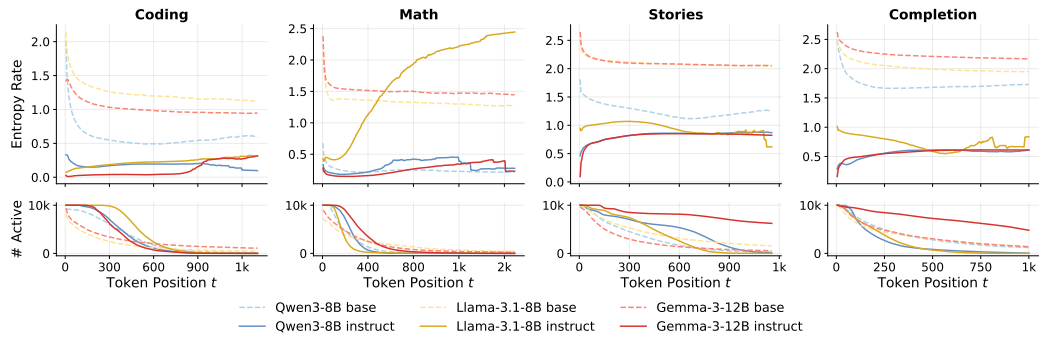

Figure 2: Running entropy rate vs. token position. We plot mean running entropy rate $\bar{R}_{\leq t} = \frac{1}{t}\sum_{s=1}^{t} H(Z_s \mid G_s, X)$ averaged across active rollouts at position t, with the corresponding active rollout count beneath. A rollout is considered active at position t if its generated sequence has length $N \geq t$ (i.e., it has not terminated before position t). Dashed curves denote base models and solid curves denote fine-tuned models.

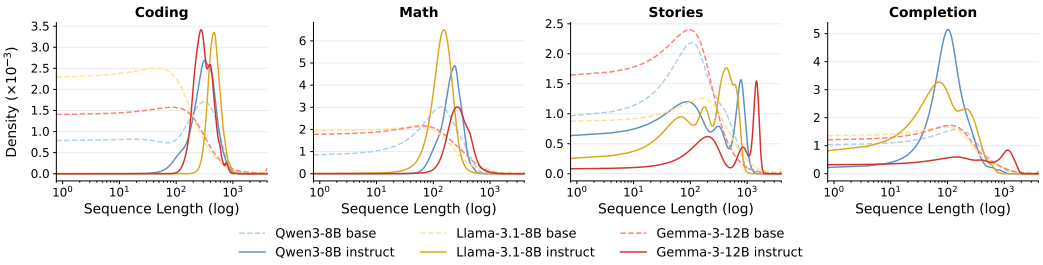

Figure 3: Kernel density estimates of generated sequence lengths across task domains with sequence length shown on a logarithmic scale.

normalization effects, potentially biasing diversity comparisons [11, 32]. The positive length–entropy correlation observed here demonstrates that increased length in fine-tuned models reflects sustained information content rather than simple redundancy. Under five-shot prompting, positive shifts persist in the majority of model–task comparisons (see subsection D.2).

Overall, these results suggest that fine-tuning does not merely reduce the quantity of uncertainty, but also improves its organization and efficiency. Rather than uniformly suppressing diversity, fine-tuning restructures uncertainty to better align with task requirements – constraining the generation space while preserving informative variation across longer trajectories. Our findings on generation space uncertainty and length–entropy coupling remain robust under top-$k = 5000$ entropy estimation (see subsection D.1).

5.2 Fine-tuning more effectively converts token uncertainty into semantic diversity

After establishing how fine-tuning reshapes entropy allocation, we next investigate how these changes in entropy rate translate into semantic diversity [14]. Specifically, we study how generation uncertainty is converted into semantic diversity across sampled trajectories.

For each task–prompt–model tuple (t, p, m), we measure semantic diversity by computing the average pairwise cosine distance between generated outputs: $D_{tpm} = \frac{2}{M(M-1)} \sum_{i<j} d(e_{t,p,m,i}, e_{t,p,m,j})$ [38], where $e_{t,p,m,i}$ denotes the embedding of the i-th Monte Carlo rollout, prepended with the retrieval prefix (`"search_document:"`) to ensure alignment with the encoder's training objective for ModernBERT embedding model (modernbert-embed-large) [43]. Here $d(\cdot, \cdot)$ denotes cosine distance. Since cosine distance is bounded in $(0, 1)$, we employ the Beta mixed-effects regression, which is a standard method to accommodate bounded responses while allowing flexible modeling of dispersion and hierarchical variation [9].[5]

[5] In contrast, Gaussian linear models are inappropriate for bounded responses and exhibit clear heteroscedasticity and residual misspecification in diagnostic analyses.

Table 2: Base–instruct comparison of the length–entropy Pearson correlation, measured by $\rho(N, r_N)$ (see Definition 4), for the Qwen3-8B, Llama-3.1-8B, and Gemma-3-12B model families. Each entry reports the estimate $\pm$ standard error, where the absolute change is computed as the instruct value minus the corresponding base value.

Metric	Model	MATH			CODING			SENTENCE COMPLETION			STORY GENERATION		
		Base	Instruct	Abs. Change	Base	Instruct	Abs. Change	Base	Instruct	Abs. Change	Base	Instruct	Abs. Change
$\rho(N, r_N)$	Qwen	-0.018 ± 0.016	0.087 ± 0.037	0.11 ± 0.042	-0.41 ± 0.014	0.61 ± 0.034	1.0 ± 0.037	0.062 ± 0.016	0.52 ± 0.023	0.46 ± 0.027	-0.27 ± 0.017	0.16 ± 0.021	0.43 ± 0.027
	Llama	-0.27 ± 0.016	0.38 ± 0.041	0.65 ± 0.045	-0.38 ± 0.011	0.26 ± 0.018	0.64 ± 0.019	-0.26 ± 0.015	-0.29 ± 0.021	-0.028 ± 0.027	-0.24 ± 0.012	-0.13 ± 0.016	0.11 ± 0.022
	Gemma	-0.13 ± 0.014	0.23 ± 0.039	0.36 ± 0.041	-0.27 ± 0.012	0.40 ± 0.036	0.68 ± 0.038	-0.23 ± 0.015	0.15 ± 0.038	0.39 ± 0.040	-0.17 ± 0.012	-0.015 ± 0.026	0.16 ± 0.031

Formally, we model $D_{tpm} \sim \mathrm{Beta}(\mu_{tpm}\phi_{tpm}, (1-\mu_{tpm})\phi_{tpm})$, where $\mu_{tpm} \in (0,1)$ denotes the conditional mean and $\phi_{tpm} > 0$ is the precision parameter. The conditional mean model is specified as $\mathrm{logit}(\mu_{tpm}) = \alpha + \beta_1 R_{tpm} + \beta_2 \mathbf{1}\{m = \mathrm{FT}\} + f_0(\mathrm{invN}_{tpm}) + \sum_k \theta_k \mathbf{1}\{t = k\} + \beta_3 (R_{tpm} \cdot \mathbf{1}\{m = \mathrm{FT}\}) + f_1(\mathrm{invN}_{tpm})\,\mathbf{1}\{m = \mathrm{FT}\} + u_p + v_m$, where R_{tpm} is the entropy rate, and invN_{tpm} is the inverse sequence length, both standardized to zero mean and unit variance, so the entropy-rate coefficients are reported per standard deviation. $f_0(\cdot)$, $f_1(\cdot)$ are natural spline functions corresponding to nonlinear length effects and their interaction with instruction tuning. For consistency with the measurement of semantic diversity, the reported sequence length and entropy rate are likewise computed by averaging over the same set of M rollouts for each task–prompt–model combination. The coefficients θ_k represent fixed task effects. To account for hierarchical dependence, we include random intercepts $u_p \sim \mathcal{N}(0, \sigma_p^2), v_m \sim \mathcal{N}(0, \sigma_m^2)$, for prompt identity and model family, respectively. To capture heteroscedasticity, we also model the precision parameter ϕ_{tpm} as a function of task, model variant, entropy rate, sequence length, and model family. Residual diagnostics (see Figure 6) indicate good calibration of the mean structure (Kolmogorov-Smirnov Test [27]: $p = 0.47$) and no significant dispersion issues ($p = 0.11$) or outliers ($p = 0.084$).

Main findings. First, entropy rate exhibits a strong and highly significant positive association with semantic diversity. The coefficient on entropy rate is large and significant ($\beta_1 = 0.688$, $p < 2 \times 10^{-16}$), indicating that higher per-token uncertainty leads to substantially greater semantic diversity across generated outputs. More importantly, the interaction between entropy rate and fine-tuning is strongly positive ($\beta_3 = 1.323, p < 2\times10^{-16}$). Under the logit link, this implies a substantial amplification of the slope: $\partial\,\mathrm{logit}(\mu)/\partial R = \beta_1 = 0.688$ (base), $\beta_1 + \beta_3 = 0.688 + 1.323 = 2.011$ (fine-tuned). Thus, fine-tuning nearly triples the sensitivity of semantic diversity to entropy rate. This indicates that uncertainty in fine-tuned models is significantly more *semantically productive*: small increases in token-level uncertainty translate into much larger increases in trajectory-level diversity. These findings remain robust across various alternative embedding models, semantic diversity metrics, and regression specifications (see subsection D.3).

At the same time, the main effect of fine-tuning is negative ($\beta_2 = -0.325, p < 2 \times 10^{-4}$), indicating that at fixed entropy rate and length, fine-tuned models produce less baseline semantic diversity. This is consistent with alignment compressing the generation space and enforcing more structured outputs. However, the strong positive interaction shows that the remaining uncertainty is used more efficiently. We observe a nonlinear and generally negative relationship between semantic diversity and sequence length, making length control essential in diversity analysis (see Table 6). Full regression results are reported in subsection C.5, along with additional model specifications and regression details.

6 Conclusion and limitations

In this work, we revisit diversity in large language models through a trajectory-level information-theoretic perspective. By introducing Canopy Entropy and length–entropy coupling, we show that fine-tuning does not simply reduce uncertainty or shrink the generation space. Instead, aligned models reorganize uncertainty more efficiently across generation trajectories, producing longer generations that remain more semantically informative. Our results highlight the importance of modeling how uncertainty evolves throughout the rollout process, beyond aggregate diversity metrics alone.

Our work also has several limitations. Entropy-based measures characterize uncertainty allocation, but do not directly capture factuality or human preference quality. Our framework is descriptive rather than causal: although we observe systematic changes in length–entropy coupling after fine-tuning, the underlying mechanisms causing this effect remain an important direction for future work.

Acknowledgments

We thank Chenghao Yang for helpful discussions during the early stage of this project.

References

[1] Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*, 2025.

[2] Eltayeb Ahmed, Uljad Berdica, Martha Elliott, Danijela Horak, and Jakob N. Foerster. Intent factored generation: Unleashing the diversity in your language model. *CoRR*, abs/2506.09659, 2025.

[3] Fabrizio Biondi, Axel Legay, Bo Friis Nielsen, and Andrzej Wąsowski. Maximizing entropy over markov processes. *J. Log. Algebraic Methods Program.*, 83(5-6):384–399, 2014.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. *NeurIPS*, 2020.

[5] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[6] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

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

[8] Laura Ekroot and Thomas M. Cover. The entropy of a randomly stopped sequence. *IEEE Trans. Inf. Theory*, 37(6):1641–1644, 1991.

[9] Jorge I Figueroa-Zúñiga, Reinaldo B Arellano-Valle, and Silvia LP Ferrari. Mixed beta regression: A bayesian perspective. *Computational Statistics & Data Analysis*, 61:137–147, 2013.

[10] Gerald B Folland. *Real analysis: modern techniques and their applications*. John Wiley & Sons, 1999.

[11] Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

[12] Dmitriy Genzel and Eugene Charniak. Entropy rate constancy in text. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 199–206, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[13] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[14] Yanzhu Guo, Guokan Shang, and Chloé Clavel. Benchmarking linguistic diversity of large language models. *Transactions of the Association for Computational Linguistics*, 13:1507–1526, 2025.

[15] Florian Hartig. Dharma: residual diagnostics for hierarchical (multi-level/mixed) regression models. *CRAN: contributed packages*, 2016.

[16] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[17] Edward J Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing intractable inference in large language models. *arXiv preprint arXiv:2310.04363*, 2023.

[18] Mete Ismayilzada, Antonio Laverghetta Jr., Simone A. Luchini, Reet Patel, Antoine Bosselut, Lonneke Van Der Plas, and Roger E. Beaty. Creative preference optimization. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9580–9609, Suzhou, China, November 2025. Association for Computational Linguistics.

[19] Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.

[20] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[21] Robert Kubinec. Ordered beta regression: a parsimonious, well-fitting model for continuous data with lower and upper bounds. *Political analysis*, 31(4):519–536, 2023.

[22] Thom Lake, Eunsol Choi, and Greg Durrett. From distributional to overton pluralism: Investigating large language model alignment. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6794–6814, 2025.

[23] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 110–119, 2016.

[24] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601, 2024.

[25] Wei Lu, Rachel K Luu, and Markus J Buehler. Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities. *npj Computational Materials*, 11(1):84, 2025.

[26] Henry B Mann and Abraham Wald. On stochastic limit and order relationships. *The Annals of Mathematical Statistics*, 14(3):217–226, 1943.

[27] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[28] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[29] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[30] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[31] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 3rd edition, 1976.

[32] Chantal Shaib, Venkata S Govindarajan, Joe Barrow, Jiuding Sun, Alexa F Siu, Byron C Wallace, and Ani Nenkova. Standardizing the measurement of text diversity: A tool and a comparative analysis of scores. *arXiv preprint arXiv:2403.00553*, 2024.

[33] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[34] Alexander Shypula, Shuo Li, Botong Zhang, Vishakh Padmakumar, Kayo Yin, and Osbert Bastani. Does instruction tuning reduce diversity? a case study using code generation, 2025.

[35] Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*, 2023.

[36] C Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[37] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, et al. Gemma 3 technical report, 2025.

[38] Guy Tevet and Jonathan Berant. Evaluating the evaluation of diversity in natural language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 326–346, 2021.

[39] Vivek Verma, Nicholas Tomlin, and Dan Klein. Revisiting entropy rate constancy in text. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15537–15549, Singapore, December 2023. Association for Computational Linguistics.

[40] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11897–11916. Association for Computational Linguistics, 2024.

[41] Shumin Wang, Yuexiang Xie, Wenhao Zhang, Yuchang Sun, Yanxi Chen, Yaliang Li, and Yanyong Zhang. On the entropy dynamics in reinforcement fine-tuning of large language models. *arXiv preprint arXiv:2602.03392*, 2026.

[42] Yichen Wang, Chenghao Yang, Tenghao Huang, Muhao Chen, Jonathan May, and Mina Lee. Optimizing diversity and quality through base-aligned model collaboration. *arXiv preprint arXiv:2511.05650*, 2025.

[43] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024.

[44] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[45] Peter West and Christopher Potts. Base models beat aligned models at randomness and creativity. *arXiv preprint arXiv:2505.00047*, 2025.

[46] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[47] Chenghao Yang, Sida Li, and Ari Holtzman. Llm probability concentration: How alignment shrinks the generative horizon. *arXiv preprint arXiv:2506.17871*, 2025.

[48] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[49] Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. *arXiv preprint arXiv:2406.15877*, 2024.

A Missing proofs and algorithms

A.1 Equivalence to a two-stage stochastic decision process

Theorem 3 (Equivalence between autoregressive generation and a two-stage decision process)**.** *Fix a prompt $X = x$ and a generation history $G_t = g_t = (y_1, \dots, y_{t-1})$. Suppose the autoregressive model defines a probability distribution over the extended vocabulary $Z := \mathcal{V} \cup \{\texttt{EOS}\}$, denoted by $q(z \mid g_t, x), z \in Z$. Then this one-step generation rule is equivalent to the following two-stage stochastic decision process: $A_t \in \{\textsf{STOP}, \textsf{CONT}\}$, where $\mathbb{P}(A_t = \textsf{STOP} \mid g_t, x) = q(\texttt{EOS} \mid g_t, x)$, and $\mathbb{P}(A_t = \textsf{CONT} \mid g_t, x) = 1 - q(\texttt{EOS} \mid g_t, x)$. Conditional on $A_t = \textsf{CONT}$, the next token $Y_t \in \mathcal{V}$ is sampled from $\mathbb{P}(Y_t = v \mid g_t, x, A_t = \textsf{CONT}) = \frac{q(v \mid g_t, x)}{1 - q(\texttt{EOS} \mid g_t, x)}, v \in \mathcal{V}$, whenever $1 - q(\texttt{EOS} \mid g_t, x) > 0$. This two-stage process induces the same distribution over generated sequences as the original autoregressive model.*

Proof. Fix a prompt x and history g_t. Under the original autoregressive model, the next outcome is sampled directly from $q(z \mid g_t, x), z \in \mathcal{V} \cup \{\texttt{EOS}\}$. In particular, generation terminates at step t with probability $q(\texttt{EOS} \mid g_t, x)$. Now consider the two-stage process. The probability of termination is $\mathbb{P}(A_t = \textsf{STOP} \mid g_t, x) = q(\texttt{EOS} \mid g_t, x)$, which matches the probability assigned to EOS by the original model. Next, for any token $v \in \mathcal{V}$, the probability that the two-stage process generates v is

$$\begin{aligned}\mathbb{P}(Y_t = v \mid g_t, x) &= \mathbb{P}(A_t = \textsf{CONT} \mid g_t, x)\, \mathbb{P}(Y_t = v \mid g_t, x, A_t = \textsf{CONT}) \\ &= (1 - q(\texttt{EOS} \mid g_t, x))\, \frac{q(v \mid g_t, x)}{1 - q(\texttt{EOS} \mid g_t, x)} \\ &= q(v \mid g_t, x).\end{aligned}$$

Thus, the two-stage process assigns exactly the same probability to every possible next outcome:

$$\mathbb{P}(\textsf{STOP} \mid g_t, x) = q(\texttt{EOS} \mid g_t, x),\ \mathbb{P}(Y_t = v \mid g_t, x) = q(v \mid g_t, x), \quad v \in \mathcal{V}.$$

Therefore, the one-step transition distribution is identical under the two formulations. Since this equality holds for every prompt x, every generation history g_t, and every generation step t, the induced probability of any finite sequence $y_{1:N}$ terminating at length N is the same under both processes. Thus, the two-stage stochastic decision process is equivalent to the original autoregressive generation model. □

A.2 Proof of Theorem 1

Proof. Let $X \sim p_X$ denote the prompt distribution. For each realization $X = x$, consider the sequence of random variables $(Z_t)_{t \geq 1}$ with $Z_t \in \mathcal{Z} := \mathcal{V} \cup \{\texttt{EOS}\}$, where selecting EOS indicates termination. Define the stopping time $N := \min\{t : Z_t = \texttt{EOS}\}$, and let $Y_t = Z_t$ for $t < N$. Adopt the convention that $Z_t = \texttt{EOS}$ for all $t > N$, which induces a one-to-one correspondence between the infinite sequence $(Z_1, Z_2, \dots)$ and the pair $(N, Y_{1:N})$, conditional on X. Therefore,

$$H(N, Y_{1:N} \mid X) = H(Z_1, Z_2, \dots \mid X).$$

By the chain rule of entropy, $H(Z_1, Z_2, \dots \mid X) = \sum_{t=1}^{\infty} H(Z_t \mid Z_{1:t-1}, X)$. Since the history G_t is completely determined by $(Z_{1:t-1}, X)$, we have

$$H(Z_t \mid Z_{1:t-1}, X) = H(Z_t \mid G_t, X) = \mathbb{E}_{G_t, X}\big[\widetilde{H}(G_t \mid X)\big].$$

Therefore, $H(Z_1, Z_2, \dots \mid X) = \sum_{t=1}^{\infty} \mathbb{E}_{G_t, X}\big[\widetilde{H}(G_t \mid X)\big]$.

Since $\widetilde{H}(G_t \mid X) \geq 0$, we can apply Tonelli's theorem [10] to exchange the sum and expectation.

$$\sum_{t=1}^{\infty} \mathbb{E}_{G_t, X}\big[\widetilde{H}(G_t \mid X)\big] = \mathbb{E}_{G_t, X}\left[\sum_{t=1}^{\infty} \widetilde{H}(G_t \mid X)\right].$$

Moreover, for all $t > N$, $Z_t = \texttt{EOS}$ deterministically, so $\widetilde{H}(G_t \mid X) = 0$. Therefore, $\sum_{t=1}^{\infty} \widetilde{H}(G_t \mid X) = \sum_{t=1}^{N(X)} \widetilde{H}(G_t \mid X)$, almost surely. Combining the above, we have $H(Z_1, Z_2, \dots \mid X) = \mathbb{E}_{G_t, X}\left[\sum_{t=1}^{N(X)} \widetilde{H}(G_t \mid X)\right]$. Finally, since $(Z_1, Z_2, \dots)$ is in one-to-one correspondence with $(N, Y_{1:N})$ given X, we obtain $H(N, Y_{1:N} \mid X) = \mathbb{E}_{G_t, X}\left[\sum_{t=1}^{N(X)} \widetilde{H}(G_t \mid X)\right] = \mathrm{CE}^{\star}$. □

A.3 Algorithms

Input: Prompt distribution p_X; number of prompts P; rollout count per prompt M; maximum length $T_{\max}$; sampling policy π (e.g. temperature/top-p/top-k).
for $p = 1$ **to** P **do**
  Sample prompt $x_p \sim p_X$;
  for $i = 1$ **to** M **do**
    Initialize history $G_1^{(p,i)} \leftarrow x_p$; set $S_{\max}^{(p,i)} \leftarrow 0$;
    for $t = 1$ **to** $T_{\max}$ **do**
      Compute model distribution $q_t^{(p,i)}(\cdot \mid G_t^{(p,i)}, x_p)$ under sampling policy π, including `EOS` as an outcome;
      Compute one-step local entropy $\tilde{H}(G_t^{(p,i)} \mid X_p) \leftarrow -\sum_{z \in \mathcal{Z}} q_t^{(p,i)}(z \mid G_t^{(p,i)}, x_p) \log q_t^{(p,i)}(z \mid G_t^{(p,i)}, x_p)$;
      Update $S_{\max}^{(p,i)} \leftarrow S_{\max}^{(p,i)} + \tilde{H}(G_t^{(p,i)} \mid X_p)$;
      Sample next outcome $z_t^{(p,i)} \sim q_t^{(p,i)}(\cdot \mid G_t^{(p,i)}, x_p)$;
      If $z_t^{(p,i)} = \texttt{EOS}$ then set $N_{\max}^{(p,i)} \leftarrow t$ and break;
      Else update history $G_{t+1}^{(p,i)} \leftarrow G_t^{(p,i)} \oplus z_t^{(p,i)}$.
    end
    If no `EOS` is sampled by $T_{\max}$: set $N_{\max}^{(p,i)} \leftarrow T_{\max}$.
  end
end
return $\widehat{\mathrm{CE}}^{\star}_{\max,P,M} = \frac{1}{PM}\sum_{p=1}^{P}\sum_{i=1}^{M} S_{\max}^{(p,i)}$, $\widehat{\mathbb{E}}[N_{\max}] = \frac{1}{PM}\sum_{p=1}^{P}\sum_{i=1}^{M} N_{\max}^{(p,i)}$.

Algorithm 1: Estimating $\mathrm{CE}^{\star}_{\max}$ over a prompt distribution

A.4 Examples of prompt-averaged bias decay rates

Corollary 1. *By Remark 4.1, $g(T_{\max})$ admits the following rates.*

Uniform exponential tail. *If there exist constants $c, \lambda > 0$ such that, for all prompts $x \in \mathcal{X}$ and all $t \geq 1$, $\mathbb{P}(N(x) \geq t \mid X = x) \leq ce^{-\lambda t}$, then $g(T_{\max}) \leq c \log|\mathcal{Z}| \sum_{t=T_{\max}+1}^{\infty} e^{-\lambda t} = c \log|\mathcal{Z}| \frac{e^{-\lambda(T_{\max}+1)}}{1-e^{-\lambda}} = O(e^{-\lambda T_{\max}})$.*

Uniform polynomial tail. *If there exist constants $c > 0$ and $\alpha > 1$ such that, for all prompts $x \in \mathcal{X}$ and all $t \geq 1$, $\mathbb{P}(N(x) \geq t \mid X = x) \leq ct^{-\alpha}$, then $g(T_{\max}) \leq c \log|\mathcal{Z}| \sum_{t=T_{\max}+1}^{\infty} t^{-\alpha} \leq \frac{c \log|\mathcal{Z}|}{\alpha - 1} T_{\max}^{1-\alpha} = O(T_{\max}^{1-\alpha})$.*

A.5 Proof of Theorem 2 and MSE analysis

Proof. Given any fixed prompt $x \in \mathcal{X}$, denote $\mu_{\max}(x) := \mathbb{E}[S_{\max} \mid X = x]$, and $S_{\max} = \sum_{t=1}^{N_{\max}(x)} \widetilde{H}(G_t \mid x)$. Then $\mathrm{CE}^{\star}_{\max} = \mathbb{E}_{X \sim p_X}[\mu_{\max}(X)]$. Since $|\mathcal{Z}| < \infty$, for all g_t, we have $0 \leq \widetilde{H}(g_t \mid x) \leq H_{\max} := \log|\mathcal{Z}|$. Therefore $0 \leq S_{\max} \leq H_{\max} N_{\max}(x)$. Therefore, $\mathbb{E}[S_{\max}^2] \leq H_{\max}^2 \mathbb{E}[N_{\max}(X)^2] = H_{\max}^2 \nu_{\max}^2$, where $\nu_{\max}^2 := E[N_{\max}(X)^2]$.

Now decompose the variance of $\widehat{\mathrm{CE}}^{\star}_{\max,P,M}$. Conditional on $X_1, \ldots, X_P$, the M rollouts for each prompt are independent, by law of total variance, we have

$$\mathrm{Var}\Big(\widehat{\mathrm{CE}}^{\star}_{\max,P,M}\Big) = \mathrm{Var}\left(\frac{1}{P}\sum_{p=1}^{P} \mu_{\max}(X_p)\right) + \mathbb{E}\Big[\mathrm{Var}\Big(\widehat{\mathrm{CE}}^{\star}_{\max,P,M} \mid X_1, \ldots, X_P\Big)\Big].$$

By Jensen's inequality, $\mu_{\max}^2(X) \leq H_{\max}^2 \, \mathbb{E}\big[N_{\max}(X)^2 \mid X\big]$, and taking expectations over X gives $\mathbb{E}[\mu_{\max}^2(X)] \leq H_{\max}^2 \nu_{\max}^2$. Therefore

$$\mathrm{Var}\left(\frac{1}{P}\sum_{p=1}^{P} \mu_{\max}(X_p)\right) = \frac{1}{P}\mathrm{Var}(\mu_{\max}(X)) \leq \frac{1}{P}\mathbb{E}[\mu_{\max}(X)^2] \leq \frac{H_{\max}^2 \nu_{\max}^2}{P}.$$

From $\mathbb{E}\left[\mathrm{Var}\left(\widehat{\mathrm{CE}}^{\star}_{\max,P,M} \mid X_1,\ldots,X_P\right)\right] \le \frac{H_{\max}^2\nu_{\max}^2}{PM}$, we have

$$\mathrm{Var}\left(\widehat{\mathrm{CE}}^{\star}_{\max,P,M}\right) \le H_{\max}^2\nu_{\max}^2\left(\frac{1}{P}+\frac{1}{PM}\right).$$

Since $\mathbb{E}\left[\widehat{\mathrm{CE}}^{\star}_{\max,P,M}\right] = \mathrm{CE}^{\star}_{\max}$, Chebyshev's inequality gives that, for any $\delta \in (0,1)$, with probability at least $1-\delta$, $\left|\widehat{\mathrm{CE}}^{\star}_{\max,P,M} - \mathrm{CE}^{\star}_{\max}\right| \le H_{\max}\nu_{\max}\sqrt{\frac{1}{\delta}\left(\frac{1}{P}+\frac{1}{PM}\right)}$. By the truncation bias assumption, $0 \le \mathrm{CE}^{\star} - \mathrm{CE}^{\star}_{\max} \le g(T_{\max})$. Thus,

$$\left|\widehat{\mathrm{CE}}^{\star}_{\max,P,M} - \mathrm{CE}^{\star}\right| \le H_{\max}\nu_{\max}\sqrt{\frac{1}{\delta}\left(\frac{1}{P}+\frac{1}{PM}\right)} + g(T_{\max}).$$

Equivalently, $\left|\widehat{\mathrm{CE}}^{\star}_{\max,P,M} - \mathrm{CE}^{\star}\right| = O_{\mathbb{P}}\left(\nu_{\max}\sqrt{\frac{1}{P}+\frac{1}{PM}} + g(T_{\max})\right)$. Thus, if $T_{\max} = T_{\max}(P,M) \to \infty$, $g(T_{\max}) \to 0$, and if $\nu_{\max}(P,M)\sqrt{\frac{1}{P}+\frac{1}{PM}} \to 0$, then we have $\widehat{\mathrm{CE}}^{\star}_{\max,P,M} \xrightarrow{\mathbb{P}} \mathrm{CE}^{\star}$. □

Theorem 4 (MSE of the prompt-averaged truncated CE estimator)**.**

$$\mathbb{E}\left[\left(\widehat{\mathrm{CE}}^{\star}_{\max,P,M} - \mathrm{CE}^{\star}\right)^2\right] \le \frac{H_{\max}^2}{P}\mathrm{Var}_{X\sim p_X}(\mathbb{E}[N_{\max} \mid X]) + \frac{H_{\max}^2}{PM}\mathbb{E}[N_{\max}^2] + g^2(T_{\max}).$$

In Theorem 4, we further characterize the mean squared error (MSE) of $\widehat{\mathrm{CE}}^{\star}_{\max,P,M}$ as the sum of the squared truncation bias and the variance of the estimator. The variance decreases with both the number of prompts P and the total number of rollouts PM, highlighting two distinct sources of statistical efficiency: diversity across prompts and repeated sampling within each prompt.

The variance admits a hierarchical decomposition with two distinct sources of randomness: a $1/P$ term due to variability across prompts, and a $1/PM$ term arising from rollout-level stochasticity within each prompt. Importantly, the prompt-level variance cannot be reduced by increasing the number of rollouts per prompt, reflecting the intrinsic hierarchical structure of the estimator. Moreover, the Monte Carlo error is scaled by the effective trajectory length (captured by $T_{\max}$ or, more precisely, $N_{\max}$), since longer generations accumulate more uncertainty along the trajectory.

These results reveal a fundamental trade-off. Increasing $T_{\max}$ reduces the truncation bias by capturing more of the generation process, but also increases computational cost and variance through longer trajectories.

Proof. Define the per-prompt rollout average $\bar{S}^{(p)}_{\max} := \frac{1}{M}\sum_{i=1}^{M} S^{(p,i)}_{\max}$. Then $\widehat{\mathrm{CE}}^{\star}_{\max,P,M} = \frac{1}{P}\sum_{p=1}^{P}\bar{S}^{(p)}_{\max}$. Conditional on X_p, $\mathbb{E}[\bar{S}^{(p)}_{\max} \mid X_p] = \mu_{\max}(X_p)$, and, since the M rollouts are conditionally independent, $\mathrm{Var}(\bar{S}^{(p)}_{\max} \mid X_p) = \frac{1}{M}\sigma^2_{\max}(X_p)$. By the law of total variance, we have

$$\mathrm{Var}(\bar{S}^{(p)}_{\max}) = \mathrm{Var}_{X_p}\left(\mathbb{E}[\bar{S}^{(p)}_{\max} \mid X_p]\right) + \mathbb{E}_{X_p}\left[\mathrm{Var}(\bar{S}^{(p)}_{\max} \mid X_p)\right].$$

Substituting the conditional mean and variance gives $\mathrm{Var}(\bar{S}^{(p)}_{\max}) = \mathrm{Var}_{X\sim p_X}(\mu_{\max}(X)) + \frac{1}{M}\mathbb{E}_{X\sim p_X}[\sigma^2_{\max}(X)]$. Since the prompts $X_1,\ldots,X_P$ are i.i.d., the variables $\bar{S}^{(1)}_{\max},\ldots,\bar{S}^{(P)}_{\max}$ are i.i.d. Therefore, $\mathrm{Var}\left(\widehat{\mathrm{CE}}^{\star}_{\max,P,M}\right) = \frac{1}{P}\mathrm{Var}(\bar{S}^{(p)}_{\max}) = \frac{1}{P}\mathrm{Var}_{X\sim p_X}(\mu_{\max}(X)) + \frac{1}{PM}\mathbb{E}_{X\sim p_X}[\sigma^2_{\max}(X)]$.

For the MSE, write $\widehat{\mathrm{CE}}^{\star}_{\max,P,M} - \mathrm{CE}^{\star} = \left(\widehat{\mathrm{CE}}^{\star}_{\max,P,M} - \mathrm{CE}^{\star}_{\max}\right) - g(T_{\max})$. Taking squares and expectations yields $\mathbb{E}\left[\left(\widehat{\mathrm{CE}}^{\star}_{\max,P,M} - \mathrm{CE}^{\star}\right)^2\right] = \mathbb{E}\left[\left(\widehat{\mathrm{CE}}^{\star}_{\max,P,M} - \mathrm{CE}^{\star}_{\max}\right)^2\right] + (g(T_{\max}))^2$, because $\mathbb{E}[\widehat{\mathrm{CE}}^{\star}_{\max,P,M} - \mathrm{CE}^{\star}_{\max}] = 0$.

Since pathwise, for any given history g_t and prompt x, we have $0 \le S_{\max}(x) = \sum_{t=1}^{N_{\max}(x)} \widetilde{H}(g_t \mid x) \le H_{\max} N_{\max}(x)$. Thus, conditional on $X = x$, $\mathrm{Var}(S_{\max}(x) \mid X = x) \le \mathbb{E}[S_{\max}^2(x) \mid X = x] \le H_{\max}^2 \mathbb{E}[N_{\max}^2(x) \mid X = x]$. Averaging over prompts gives $\mathbb{E}_X[\sigma_{\max}^2(X)] \le H_{\max}^2 \mathbb{E}_X \mathbb{E}[N_{\max}^2(X) \mid X] = H_{\max}^2 \mathbb{E}[N_{\max}^2]$.

Similarly, $\mu_{\max}(X) = \mathbb{E}[S_{\max}(X) \mid X] \le H_{\max} \mathbb{E}[N_{\max} \mid X]$. Since both quantities are non-negative, this gives the prompt-level bound $0 \le \mu_{\max}(X) \le H_{\max} \mathbb{E}[N_{\max} \mid X]$. Therefore, $\mathrm{Var}_X(\mu_{\max}(X)) \le H_{\max}^2 \mathrm{Var}_X(\mathbb{E}[N_{\max} \mid X])$. Combining these two refined bounds with the exact variance decomposition gives

$$\mathrm{Var}\left(\widehat{\mathrm{CE}}^{\star}_{\max,P,M}\right) \le \frac{H_{\max}^2}{P} \mathrm{Var}_{X \sim p_X}(\mathbb{E}[N_{\max} \mid X]) + \frac{H_{\max}^2}{PM} \mathbb{E}[N_{\max}^2].$$

Adding the squared truncation bias $g^2(T_{\max})$ gives the stated refined MSE upper bound. □

A.6 Estimation and consistency of GenPPL, BF, and length–entropy correlation

Lemma 1 (Consistency of the truncated length estimator)**.** *Let* $\widehat{N}_{\max,P,M} := \frac{1}{PM} \sum_{p=1}^{P} \sum_{i=1}^{M} N_{\max}^{(p,i)}$, *where* $N_{\max}^{(p,i)} := \min\{N^{(p,i)}, T_{\max}\}$. *Assume* $\mathbb{E}[N] < \infty$. *If* $T_{\max} = T_{\max}(P, M) \to \infty$ *and* $\frac{\sqrt{\mathbb{E}[N_{\max}^2]}}{\sqrt{P}} \to 0$, *then* $\widehat{N}_{\max,P,M} \xrightarrow{\mathbb{P}} \mathbb{E}[N]$.

Proof. Let $\eta_{\max}(x) := \mathbb{E}[N_{\max} \mid X = x]$. By the law of total variance, $\mathrm{Var}(\widehat{N}_{\max,P,M}) = \frac{1}{P} \mathrm{Var}_{X \sim p_X}(\eta_{\max}(X)) + \frac{1}{PM} \mathbb{E}_{X \sim p_X}[\mathrm{Var}(N_{\max} \mid X)]$. Using $\mathrm{Var}(\eta_{\max}(X)) \le \mathbb{E}[\eta_{\max}(X)^2] \le \mathbb{E}[N_{\max}^2]$, and $\mathbb{E}[\mathrm{Var}(N_{\max} \mid X)] \le \mathbb{E}[N_{\max}^2]$, we obtain $\mathrm{Var}(\widehat{N}_{\max,P,M}) \le \mathbb{E}[N_{\max}^2]\left(\frac{1}{P} + \frac{1}{PM}\right)$. Therefore, under the stated condition, $\mathrm{Var}(\widehat{N}_{\max,P,M}) \to 0$.

In addition, $\mathbb{E}[\widehat{N}_{\max,P,M}] = \mathbb{E}[N_{\max}]$. Since $N_{\max} = \min\{N, T_{\max}\} \uparrow N$ as $T_{\max} \to \infty$, the monotone convergence theorem [31] gives $\mathbb{E}[N_{\max}] \to \mathbb{E}[N]$. Hence,

$$\widehat{N}_{\max,P,M} - \mathbb{E}[N] = \left(\widehat{N}_{\max,P,M} - \mathbb{E}[N_{\max}]\right) + \left(\mathbb{E}[N_{\max}] - \mathbb{E}[N]\right).$$

The first term converges to 0 in probability by Chebyshev's inequality, and the second term converges to 0 deterministically. Therefore, $\widehat{N}_{\max,P,M} \xrightarrow{\mathbb{P}} \mathbb{E}[N]$. □

Corollary 2 (Consistency of GenPPL)**.** *Let* $\widehat{\mathrm{CE}}^{\star}_{\max,P,M} = \frac{1}{PM} \sum_{p=1}^{P} \sum_{i=1}^{M} S_{\max}^{(p,i)}$ *be the Monte Carlo estimator of* $\mathrm{CE}^{\star}$, *and let* $\widehat{N}_{\max,P,M} := \frac{1}{PM} \sum_{p=1}^{P} \sum_{i=1}^{M} N_{\max}^{(p,i)}$ *be the corresponding estimator of* $\mathbb{E}[N]$. *Recall that* $\widehat{\mathrm{CE}}^{\star}_{\max,P,M} \xrightarrow{\mathbb{P}} \mathrm{CE}^{\star}$ *(by Theorem 2) and* $\widehat{N}_{\max,P,M} \xrightarrow{\mathbb{P}} \mathbb{E}[N]$ *(by Lemma 1). Define* $\widehat{\mathrm{GenPPL}}_{\max,P,M} := \exp\left(\frac{\widehat{\mathrm{CE}}^{\star}_{\max,P,M}}{\widehat{N}_{\max,P,M}}\right)$, $\mathrm{GenPPL} := \exp\left(\frac{\mathrm{CE}^{\star}}{\mathbb{E}[N]}\right)$. *Then* $\widehat{\mathrm{GenPPL}}_{\max,P,M} \xrightarrow{\mathbb{P}} \mathrm{GenPPL}$.

Proof. Since we have $\widehat{\mathrm{CE}}^{\star}_{\max,P,M} \xrightarrow{\mathbb{P}} \mathrm{CE}^{\star}$, $\widehat{N}_{\max,P,M} \xrightarrow{\mathbb{P}} \mathbb{E}[N]$, and $\mathbb{E}[N] > 0$, the function $f(a, b) = \exp(a/b)$ is continuous at $(\mathrm{CE}^{\star}, \mathbb{E}[N])$. Therefore, by the continuous mapping theorem [26], $\exp\left(\frac{\widehat{\mathrm{CE}}^{\star}_{\max,P,M}}{\widehat{N}_{\max,P,M}}\right) \xrightarrow{\mathbb{P}} \exp\left(\frac{\mathrm{CE}^{\star}}{\mathbb{E}[N]}\right)$. Hence, $\widehat{\mathrm{GenPPL}}_{\max,P,M} \xrightarrow{\mathbb{P}} \mathrm{GenPPL}$, which completes the proof. □

Corollary 3 (Consistency of the BF estimator)**.** *Let* $\widehat{L}_{\max,P,M} := \frac{1}{PM} \sum_{p=1}^{P} \sum_{i=1}^{M} \frac{S_{\max}^{(p,i)}}{N_{\max}^{(p,i)}}$, $\widehat{\mathrm{BF}}_{\max,P,M} := \exp\left(\widehat{L}_{\max,P,M}\right)$, *where* $S_{\max}^{(p,i)} = \sum_{t=1}^{N_{\max}^{(p,i)}} \widetilde{H}(G_t^{(p,i)} \mid x_p)$, $N_{\max}^{(p,i)} = \min\{N^{(p,i)}, T_{\max}\}$. *Assume that* $N \ge 1$ *almost surely and that the normalized truncated entropy variables* $r_{\max} := \frac{S_{\max}}{N_{\max}}$ *are integrable. Then, for fixed* $T_{\max}$, *as* $P, M \to \infty$, $\widehat{L}_{\max,P,M} \xrightarrow{\mathbb{P}} L_{\max} := \mathbb{E}\left[\frac{S_{\max}}{N_{\max}}\right]$, *and, therefore, by the continuous mapping theorem,* $\widehat{\mathrm{BF}}_{\max,P,M} \xrightarrow{\mathbb{P}} \mathrm{BF}_{\max} := \exp(L_{\max})$. *Moreover, if* $\frac{S_{\max}}{N_{\max}} \xrightarrow[T_{\max} \to \infty]{a.s.} \frac{S}{N}$ *and the sequence*

$\{S_{\max}/N_{\max}\}_{T_{\max}\geq 1}$ *is uniformly integrable, then* $L_{\max} \to L := \mathbb{E}\Big[\frac{1}{N}\sum_{t=1}^{N}\widetilde{H}(G_t \mid X)\Big]$, *and consequently* $\mathrm{BF}_{\max} \to \mathrm{BF} := \exp(L)$. *Thus, under these conditions,* $\widehat{\mathrm{BF}}_{\max,P,M} \xrightarrow{\mathbb{P}} \mathrm{BF}$ *as* $P, M \to \infty$ *and* $T_{\max} \to \infty$.

Proof. For fixed $T_{\max}$, the variables $\hat{r}_{\max}^{(p,i)} = \frac{S_{\max}^{(p,i)}}{N_{\max}^{(p,i)}}$ are integrable rollout-level observations. Since $N \geq 1$ almost surely, the denominator is bounded away from zero. Applying the law of large numbers to the empirical average over prompts and rollouts gives $\widehat{L}_{\max,P,M} = \frac{1}{PM}\sum_{p=1}^{P}\sum_{i=1}^{M}\hat{r}_{\max}^{(p,i)} \xrightarrow{\mathbb{P}} \mathbb{E}[r_{\max}] = L_{\max}$. Since the exponential map is continuous, $\widehat{\mathrm{BF}}_{\max,P,M} = \exp(\widehat{L}_{\max,P,M}) \xrightarrow{\mathbb{P}} \exp(L_{\max}) = \mathrm{BF}_{\max}$. Finally, if $S_{\max}/N_{\max} \to S/N$ almost surely and the normalized entropy ratios are uniformly integrable, then convergence of expectations follows, yielding $L_{\max} \to L$. Another application of continuity of the exponential function gives $\mathrm{BF}_{\max} \to \mathrm{BF}$. □

Empirical support for the truncation condition. The consistency result above requires the truncation bias induced by $T_{\max}$ to vanish, or at least to be negligible in finite-sample estimation. While this is a population-level condition and cannot be proven from data alone, our diagnostics provide empirical support for it in our experimental setting. With $T_{\max} = 4000$, the observed truncation rates are uniformly small across tasks, model families, and variants: most are below 1%, and the largest observed rate is 4.32% (see subsection C.1). In addition, the empirical length distributions concentrate well below the truncation threshold (see Figure 3), and the number of active rollouts decays rapidly with token position (see Figure 2). Together with the bounded observed entropy-rate trajectories (see Figure 2), these results suggest that the normalized entropy ratios are well behaved and that the finite-$T_{\max}$ truncation bias is small in practice.

Estimating entropy rate. For each rollout (p, i), Algorithm 1 returns the truncated stopping time $N_{\max}^{(p,i)} = \min\{N^{(p,i)}, T_{\max}\}$ and the accumulated truncated canopy entropy $S_{\max}^{(p,i)} = \sum_{t=1}^{N_{\max}^{(p,i)}}\widetilde{H}(G_t^{(p,i)} \mid x_p)$. We estimate the rollout-level entropy rate by $\widehat{r}_{\max}^{(p,i)} := \frac{S_{\max}^{(p,i)}}{N_{\max}^{(p,i)}}$. Thus, for each prompt x_p, the prompt-specific length–entropy relationship is estimated from the rollout-level pairs $\left\{\left(N_{\max}^{(p,i)}, \widehat{r}_{\max}^{(p,i)}\right)\right\}_{i=1}^{M}$.

Length–entropy correlation. Definition 4 introduced the length–entropy correlation $\rho(N, r_N)$ at a conceptual level. We now provide its statistically rigorous formulation together with the corresponding aggregation procedure.

Definition 5 (length–entropy correlation)**.** *For a fixed prompt* $X = x$, *define the prompt-conditional Pearson correlation* $\rho_X := \rho(N, r_N \mid X = x) = \frac{\mathrm{Cov}(N, r_N \mid X=x)}{\sqrt{\mathrm{Var}(N \mid X=x)\,\mathrm{Var}(r_N \mid X=x)}}$. *Since the prompt-conditional correlation* ρ_X *is bounded and nonlinear, directly averaging correlations across prompts may introduce bias. Therefore, we aggregate the prompt-conditional Pearson correlations on the Fisher-*z[6] *scale to obtain a prompt-averaged correlation measure. Then the (prompt-averaged) length–entropy correlation* $\rho(N, r_N) := \tanh\left(\frac{\mathbb{E}_{X\sim p_X}[w_X\,\mathrm{atanh}(\rho_X)]}{\mathbb{E}_{X\sim p_X}[w_X]}\right)$, *where* $w_X \geq 0$ *is a prompt-specific reliability weight.*

Rank-based length–entropy correlations. In addition to Pearson correlation, which measures linear association, we also report two rank-based measures. Spearman correlation [36] is Pearson correlation applied to the ranked variables and therefore captures monotone dependence. Kendall's τ_b [20] measures ordinal association by comparing concordant and discordant rollout pairs, with tie adjustment. Since Kendall's τ_b is defined through pairwise comparisons, we aggregate it using pair-count weights rather than Fisher-z averaging. We estimate and report rank-based length–entropy correlations to assess the robustness of our findings. As shown in Table 3, fine-tuning increases both Spearman's ρ and Kendall's τ in all twelve model–task settings, including Llama-3.1-8B on

[6]This transformation maps correlation coefficients from $[-1, 1]$ to $\mathbb{R}$ and stabilizes their variance, making them approximately normally distributed with variance $1/(n_p - 3)$. n_p is the number of rollouts associated with each prompt p. Directly averaging correlations would lead to biased estimates due to the nonlinear and bounded nature of correlation coefficients. See algorithm 2 for details.

Table 3: Base–instruct comparison of length–entropy correlation measured by Spearman's ρ and Kendall's τ. Each entry reports estimate $\pm$ standard error, and absolute change is computed as instruct minus base. All numbers are rounded to two significant digits. Warm colors indicate increases, while cold colors indicate decreases; darker colors correspond to larger magnitude changes.

Task	Type	Spearman's ρ correlation			Kendall's τ correlation		
		Qwen3-8B	Llama-3.1-8B	Gemma-3-12B	Qwen3-8B	Llama-3.1-8B	Gemma-3-12B
MATH	Base	-0.19 ± 0.023	-0.56 ± 0.031	-0.27 ± 0.023	-0.12 ± 0.015	-0.36 ± 0.024	-0.17 ± 0.015
	Instruct	0.039 ± 0.032	0.15 ± 0.031	0.12 ± 0.032	0.026 ± 0.020	0.094 ± 0.021	0.070 ± 0.020
	Abs. Change	0.23 ± 0.037	0.70 ± 0.048	0.39 ± 0.043	0.15 ± 0.024	0.45 ± 0.034	0.24 ± 0.027
CODING	Base	-0.28 ± 0.018	-0.66 ± 0.014	-0.40 ± 0.011	-0.18 ± 0.013	-0.47 ± 0.012	-0.26 ± 0.0079
	Instruct	0.49 ± 0.030	0.25 ± 0.016	0.38 ± 0.032	0.32 ± 0.021	0.17 ± 0.011	0.25 ± 0.022
	Abs. Change	0.76 ± 0.035	0.91 ± 0.019	0.78 ± 0.032	0.50 ± 0.024	0.64 ± 0.015	0.51 ± 0.022
SENTENCE COMPLETION	Base	-0.0049 ± 0.025	-0.39 ± 0.019	-0.32 ± 0.016	0.0026 ± 0.017	-0.26 ± 0.013	-0.22 ± 0.011
	Instruct	0.52 ± 0.024	-0.30 ± 0.024	0.14 ± 0.034	0.36 ± 0.018	-0.20 ± 0.016	0.092 ± 0.021
	Abs. Change	0.53 ± 0.033	0.091 ± 0.029	0.46 ± 0.038	0.35 ± 0.023	0.063 ± 0.020	0.31 ± 0.024
STORY	Base	-0.36 ± 0.015	-0.32 ± 0.015	-0.30 ± 0.014	-0.25 ± 0.011	-0.21 ± 0.011	-0.20 ± 0.0099
	Instruct	0.16 ± 0.021	-0.13 ± 0.016	-0.018 ± 0.025	0.10 ± 0.013	-0.089 ± 0.010	-0.013 ± 0.016
	Abs. Change	0.52 ± 0.025	0.18 ± 0.025	0.28 ± 0.030	0.35 ± 0.017	0.12 ± 0.017	0.18 ± 0.020

sentence completion, where the Pearson change is slightly negative but not statistically significant. The predominantly positive base–instruct shift across three different correlation measures suggests that the observed trend is not an artifact of any particular notion of correlation.

Proposition 1 (Consistency of prompt-averaged length–entropy correlation estimators)**.** *For each prompt $X = x$, define $r_{\max}(x) := \frac{S_{\max}(x)}{N_{\max}(x)}$, $S_{\max}(x) = \sum_{t=1}^{N_{\max}(x)} \widetilde{H}(G_t \mid x)$, $N_{\max}(x) = \min\{N(x), T_{\max}\}$. Note that in this case, $S_{\max}(x)$ and $N_{\max}(x)$ are random variables which depend on the generated history G_t. Let $\rho^{\mathrm{P}}_{\max}(x) := \mathrm{Corr}(N_{\max}(x), r_{\max}(x) \mid X = x)$ denote the Pearson correlation, let $\rho^{\mathrm{S}}_{\max}(x) := \mathrm{Corr}\left(F_{N_{\max}|x}(N_{\max}), F_{r_{\max}|x}(r_{\max}) \mid X = x\right)$ denote the population Spearman rank correlation, and let $\tau^{\mathrm{K}}_{\max}(x)$ denote Kendall's τ_b between $N_{\max}$ and $r_{\max}$ conditional on $X = x$. For sampled prompts $x_1, \ldots, x_P$, suppose that for each prompt x_p we observe n_p conditionally i.i.d. rollouts and compute the prompt-specific estimates $\widehat{\rho}^{\mathrm{P}}_p$, $\widehat{\rho}^{\mathrm{S}}_p$, $\widehat{\tau}^{\mathrm{K}}_p$, $\widehat{\mathrm{Cov}}_p$ as in Algorithm 2. Assume that $N_{\max} \geq 1$ almost surely. For Pearson and covariance, assume that the conditional fourth moments of $N_{\max}$ and $r_{\max}$ are uniformly bounded and that $0 < c \leq \mathrm{Var}(N_{\max} \mid X = x)$, $0 < c \leq \mathrm{Var}(r_{\max} \mid X = x)$ uniformly over x. For Spearman and Kendall, assume the corresponding rank-correlation functionals are continuous at the conditional law of $(N_{\max}, r_{\max}) \mid X = x$, uniformly over x; in particular, ties are either absent or handled by the mid-rank and τ_b conventions used in Algorithm 2. Finally, assume $|\rho^{\mathrm{P}}_{\max}(x)| \leq 1-\epsilon_0$, $|\rho^{\mathrm{S}}_{\max}(x)| \leq 1-\epsilon_0$ for some $\epsilon_0 > 0$, and that $\min_p n_p \to \infty$ as $P \to \infty$. Then, as $P \to \infty$ and $\min_p n_p \to \infty$, $\widehat{\rho}^{\mathrm{P}}_{\mathrm{pc,max}} \xrightarrow{\mathbb{P}} \rho^{\mathrm{P}}_{\mathrm{pc,max}}$, $\widehat{\rho}^{\mathrm{S}}_{\mathrm{pc,max}} \xrightarrow{\mathbb{P}} \rho^{\mathrm{S}}_{\mathrm{pc,max}}$, where $\rho^{\mathrm{P}}_{\mathrm{pc,max}} = \tanh\left(\frac{\mathbb{E}_X[w^{\mathrm{F}}(X)\,\mathrm{atanh}\{\rho^{\mathrm{P}}_{\max}(X)\}]}{\mathbb{E}_X[w^{\mathrm{F}}(X)]}\right)$, $\rho^{\mathrm{S}}_{\mathrm{pc,max}} = \tanh\left(\frac{\mathbb{E}_X[w^{\mathrm{F}}(X)\,\mathrm{atanh}\{\rho^{\mathrm{S}}_{\max}(X)\}]}{\mathbb{E}_X[w^{\mathrm{F}}(X)]}\right)$. Also, $\widehat{\tau}^{\mathrm{K}}_{\mathrm{pc,max}} \xrightarrow{\mathbb{P}} \tau^{\mathrm{K}}_{\mathrm{pc,max}} := \frac{\mathbb{E}_X[w^{\mathrm{K}}(X)\tau^{\mathrm{K}}_{\max}(X)]}{\mathbb{E}_X[w^{\mathrm{K}}(X)]}$, and $\widehat{\mathrm{Cov}}_{\mathrm{pc,max}} \xrightarrow{\mathbb{P}} \mathrm{Cov}_{\mathrm{pc,max}} := \frac{\mathbb{E}_X[w^{\mathrm{Cov}}(X)\,\mathrm{Cov}(N_{\max}, r_{\max} \mid X)]}{\mathbb{E}_X[w^{\mathrm{Cov}}(X)]}$. If additionally $T_{\max} \to \infty$ and the corresponding truncated targets converge to their untruncated counterparts, then the same estimators are consistent for the corresponding prompt-averaged untruncated Pearson, Spearman, Kendall, and covariance targets.*

Proof. We prove the result in two steps: first within each prompt, and then across prompts. For a fixed prompt x_p, the rollout-level pairs $\left(N^{(p,i)}_{\max}, r^{(p,i)}_{\max}\right)$, $i = 1, \ldots, n_p$, are conditionally i.i.d. By Algorithm 2, both coordinates are observed from the same rollout, with $r^{(p,i)}_{\max} = \frac{S^{(p,i)}_{\max}}{N^{(p,i)}_{\max}}$. The moment and nondegeneracy assumptions imply consistency of the sample covariance and variances. Hence, $\widehat{\rho}^{\mathrm{P}}_p \xrightarrow{\mathbb{P}} \rho^{\mathrm{P}}_{\max}(x_p)$, $\widehat{\mathrm{Cov}}_p \xrightarrow{\mathbb{P}} \mathrm{Cov}(N_{\max}, r_{\max} \mid x_p)$. Similarly, by consistency of empirical ranks and the assumed continuity of the rank-correlation functionals, $\widehat{\rho}^{\mathrm{S}}_p \xrightarrow{\mathbb{P}} \rho^{\mathrm{S}}_{\max}(x_p)$, $\widehat{\tau}^{\mathrm{K}}_p \xrightarrow{\mathbb{P}} \tau^{\mathrm{K}}_{\max}(x_p)$. Since $|\rho^{\mathrm{P}}_{\max}(x)|$ and $|\rho^{\mathrm{S}}_{\max}(x)|$ are bounded away from one, and the clipping constant is chosen smaller than this margin, the Fisher transform is continuous in a neigh-

Input: Prompt distribution p_X; number of prompts P; rollout counts $\{n_p\}_{p=1}^{P}$; maximum length $T_{\max}$; sampling policy π; clipping constant $\epsilon > 0$.

for $p = 1$ **to** P **do**

  Sample prompt $x_p \sim p_X$;

  for $i = 1$ **to** n_p **do**

    Generate one rollout using Algorithm 1;

    Record $N_{\max}^{(p,i)} = \min\{N^{(p,i)}, T_{\max}\}$, $S_{\max}^{(p,i)} = \sum_{t=1}^{N_{\max}^{(p,i)}} \widetilde{H}(G_t^{(p,i)} \mid x_p)$. Compute the rollout-level entropy rate $r_{\max}^{(p,i)} = \frac{S_{\max}^{(p,i)}}{N_{\max}^{(p,i)}}$.

  end

  Compute the prompt-specific Pearson correlation $\widehat{\rho}_p^{\mathrm{P}} = \mathrm{Corr}\left(\{N_{\max}^{(p,i)}\}_{i=1}^{n_p}, \{r_{\max}^{(p,i)}\}_{i=1}^{n_p}\right)$.

  Compute the prompt-specific Spearman correlation $\widehat{\rho}_p^{\mathrm{S}} = \mathrm{Corr}\left(\{\mathrm{rank}(N_{\max}^{(p,i)})\}_{i=1}^{n_p}, \{\mathrm{rank}(r_{\max}^{(p,i)})\}_{i=1}^{n_p}\right)$, using mid-ranks for ties.

  Compute the prompt-specific Kendall's τ_b $\widehat{\tau}_p^{\mathrm{K}} = \mathrm{KendallTauB}\left(\{N_{\max}^{(p,i)}\}_{i=1}^{n_p}, \{r_{\max}^{(p,i)}\}_{i=1}^{n_p}\right)$.

  Compute the prompt-specific covariance $\widehat{\mathrm{Cov}}_p = \widehat{\mathrm{Cov}}\left(\{N_{\max}^{(p,i)}\}_{i=1}^{n_p}, \{r_{\max}^{(p,i)}\}_{i=1}^{n_p}\right)$.

end

Aggregate Pearson and Spearman using Fisher-z weighted averages: $w_p^{\mathrm{F}} = \max\{n_p - 3, 0\}$,

$\widehat{\rho}_{\mathrm{pc}}^{\mathrm{P}} = \tanh\left(\frac{\sum_{p=1}^{P} w_p^{\mathrm{F}} \operatorname{atanh}\left(\mathrm{clip}(\widehat{\rho}_p^{\mathrm{P}}, -1+\epsilon, 1-\epsilon)\right)}{\sum_{p=1}^{P} w_p^{\mathrm{F}}}\right)$, and

$\widehat{\rho}_{\mathrm{pc}}^{\mathrm{S}} = \tanh\left(\frac{\sum_{p=1}^{P} w_p^{\mathrm{F}} \operatorname{atanh}\left(\mathrm{clip}(\widehat{\rho}_p^{\mathrm{S}}, -1+\epsilon, 1-\epsilon)\right)}{\sum_{p=1}^{P} w_p^{\mathrm{F}}}\right)$.

Aggregate Kendall's τ_b using pair-count weights:$w_p^{\mathrm{K}} = \max\left\{\frac{n_p(n_p-1)}{2}, 0\right\}$,$\widehat{\tau}_{\mathrm{pc}}^{\mathrm{K}} = \frac{\sum_{p=1}^{P} w_p^{\mathrm{K}} \widehat{\tau}_p^{\mathrm{K}}}{\sum_{p=1}^{P} w_p^{\mathrm{K}}}$.

Aggregate covariance using degrees-of-freedom weights: $w_p^{\mathrm{Cov}} = \max\{n_p - 1, 0\}$, and

$\widehat{\mathrm{Cov}}_{\mathrm{pc}} = \frac{\sum_{p=1}^{P} w_p^{\mathrm{Cov}} \widehat{\mathrm{Cov}}_p}{\sum_{p=1}^{P} w_p^{\mathrm{Cov}}}$.

return $\widehat{\rho}_{\mathrm{pc}}^{\mathrm{P}}$, $\widehat{\rho}_{\mathrm{pc}}^{\mathrm{S}}$, $\widehat{\tau}_{\mathrm{pc}}^{\mathrm{K}}$, $\widehat{\mathrm{Cov}}_{\mathrm{pc}}$.

Algorithm 2: Estimating prompt-averaged length–entropy correlations

borhood of the limiting values. Therefore, $\operatorname{atanh}\left(\mathrm{clip}(\widehat{\rho}_p^{\mathrm{P}}, -1+\epsilon, 1-\epsilon)\right) \xrightarrow{\mathbb{P}} \operatorname{atanh}\{\rho_{\max}^{\mathrm{P}}(x_p)\}$, and $\operatorname{atanh}\left(\mathrm{clip}(\widehat{\rho}_p^{\mathrm{S}}, -1+\epsilon, 1-\epsilon)\right) \xrightarrow{\mathbb{P}} \operatorname{atanh}\{\rho_{\max}^{\mathrm{S}}(x_p)\}$. Now consider the aggregation across prompts. For Pearson, define $Z_p^{\mathrm{P}} = \operatorname{atanh}\{\rho_{\max}^{\mathrm{P}}(x_p)\}$. By the previous step and the uniform assumptions, $\frac{\sum_{p=1}^{P} w_p^{\mathrm{F}}\left[\operatorname{atanh}\left(\mathrm{clip}(\widehat{\rho}_p^{\mathrm{P}}, -1+\epsilon, 1-\epsilon)\right) - Z_p^{\mathrm{P}}\right]}{\sum_{p=1}^{P} w_p^{\mathrm{F}}} \xrightarrow{\mathbb{P}} 0$. Since prompts are sampled i.i.d., the weighted law of large numbers gives $\frac{\sum_{p=1}^{P} w_p^{\mathrm{F}} Z_p^{\mathrm{P}}}{\sum_{p=1}^{P} w_p^{\mathrm{F}}} \xrightarrow{\mathbb{P}} \frac{\mathbb{E}_X[w^{\mathrm{F}}(X) \operatorname{atanh}\{\rho_{\max}^{\mathrm{P}}(X)\}]}{\mathbb{E}_X[w^{\mathrm{F}}(X)]}$. Applying the continuous mapping theorem with $\tanh(\cdot)$ yields $\widehat{\rho}_{\mathrm{pc,max}}^{\mathrm{P}} \xrightarrow{\mathbb{P}} \rho_{\mathrm{pc,max}}^{\mathrm{P}}$. The Spearman result is identical after replacing $\rho_{\max}^{\mathrm{P}}$ by $\rho_{\max}^{\mathrm{S}}$. For Kendall's τ_b, Algorithm 2 aggregates directly on the τ scale using pair-count weights $w_p^{\mathrm{K}} = \frac{n_p(n_p-1)}{2}$. Thus, by the prompt-level consistency of $\widehat{\tau}_p^{\mathrm{K}}$ and the weighted law of large numbers, $\widehat{\tau}_{\mathrm{pc,max}}^{\mathrm{K}} = \frac{\sum_{p=1}^{P} w_p^{\mathrm{K}} \widehat{\tau}_p^{\mathrm{K}}}{\sum_{p=1}^{P} w_p^{\mathrm{K}}} \xrightarrow{\mathbb{P}} \frac{\mathbb{E}_X[w^{\mathrm{K}}(X)\tau_{\max}^{\mathrm{K}}(X)]}{\mathbb{E}_X[w^{\mathrm{K}}(X)]}$. The covariance result follows in the same way, using weights $w_p^{\mathrm{Cov}} = (n_p - 1)_+$ and the prompt-level consistency of $\widehat{\mathrm{Cov}}_p$. Finally, if $T_{\max} \to \infty$ and the truncated conditional covariance, variance, and rank-association functionals converge to their untruncated counterparts, then the truncated prompt-averaged targets converge to the corresponding untruncated targets. Combining this approximation step with the fixed-$T_{\max}$ consistency above completes the proof. □

Empirical support for the consistency assumptions. The assumptions underlying Proposition 1 are supported by several empirical diagnostics observed in our experiments. First, the truncation rates under $T_{\max} = 4000$ are uniformly small across model families, tasks, and variants, with most

Table 4: Empirical Truncation Rates. We report the proportion of Monte Carlo rollouts that reached the maximum generation budget ($T_{\max} = 4000$ tokens) without sampling an `EOS` token. Rates are calculated over $P = 100$ prompts with $M = 100$ rollouts per prompt.

Model	MATH		STORY		CODING		COMPLETION	
	Base	Instruct	Base	Instruct	Base	Instruct	Base	Instruct
Qwen3-8B	0.0073	0.0000	0.0005	0.0000	0.0008	0.0004	0.0313	0.0000
Llama-3.1-8B	0.0105	0.0012	0.0325	0.0000	0.0095	0.0000	0.0174	0.0000
Gemma-3-12B	0.0036	0.0000	0.0066	0.0000	0.0432	0.0000	0.0119	0.0000

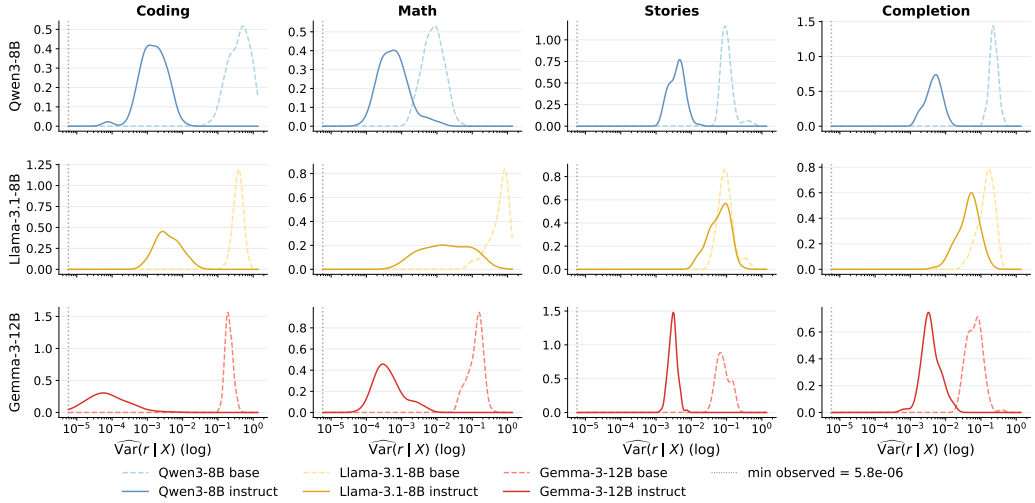

Figure 4: Gaussian KDEs of $\log \widehat{\mathrm{Var}}(r \mid x_p)$ over P=100 prompts. The dotted vertical marks $\min_p \widehat{\mathrm{Var}}(r \mid x_p) \approx 6 \times 10^{-6}$. All densities sit well to the right of zero, providing empirical support for the bounded-away-from-zero assumption.

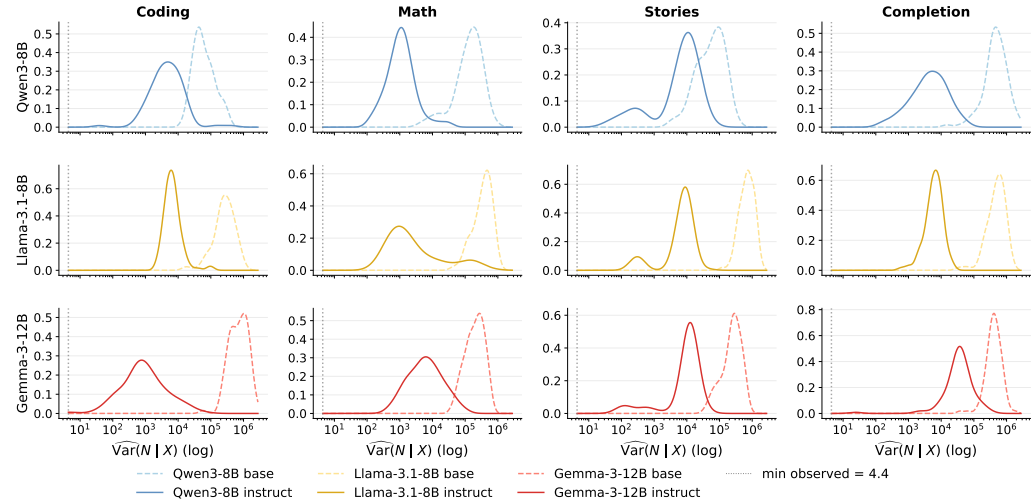

Figure 5: Gaussian KDEs of $\log \widehat{\mathrm{Var}}(N \mid x_p)$ over P=100 prompts. The dotted vertical line marks $\min_p \widehat{\mathrm{Var}}(N \mid x_p) \approx 4.4$, the smallest per-prompt variance observed across all model-dataset combinations. All densities concentrate at $\widehat{\mathrm{Var}}(N \mid x_p) \gg 0$, empirically supporting the bounded-away-from-zero assumption. Instruct variants consistently shift to smaller values, indicating that fine-tuning produces tighter length distributions.

rates below 1% and the maximum observed rate equal to 4.32% (see Table 4). This suggests that the truncated quantities $(N_{\max}, r_{\max})$ closely approximate their untruncated counterparts (N, r_N) in practice. Second, the empirical sequence-length distributions exhibit rapidly decaying tails, and the number of active rollouts decreases quickly as generation proceeds, providing evidence that the stopping-time distribution is sufficiently light-tailed and that truncation bias is negligible for the chosen $T_{\max}$ (see Figure 3).

Third, the rollout-level entropy rates remain bounded and stable throughout generation trajectories, supporting the bounded-moment and continuity assumptions required for consistency of the covariance and correlation estimators (see Figure 2). Finally, substantial variability is consistently observed in both generation length and entropy rate across prompts and rollouts, indicating that the conditional variances of $N_{\max}$ and $r_{\max}$ are bounded away from zero with high probability (see Figure 4 and Figure 5). Together, these empirical observations suggest that the regularity conditions required for the consistency of the prompt-averaged Pearson, Spearman, Kendall, and covariance estimators are well approximated in our experimental regime.

B Discussions

B.1 How GenPPL and BF capture different notions of diversity: an essay analogy case study

We illustrate the difference between GenPPL and BF through an essay–word analogy, where trajectories correspond to essays and tokens correspond to individual words. For a fixed prompt x, suppose a model generates two types of essays. With probability $1/2$, it produces a short essay of length $N = 2$, where each word is selected uniformly from 100 plausible words. With probability $1/2$, it produces a long essay of length $N = 10$, where each word is selected uniformly from only 2 plausible words. For the short essays, $H(Y_{1:2} \mid N = 2, x) = 2\log 100, r_2 = \log 100$. For the long essays, $H(Y_{1:10} \mid N = 10, x) = 10\log 2, r_{10} = \log 2$.

BF treats each *trajectory* equally: $\log \mathrm{BF} = \mathbb{E}_N[r_N] = \sum_n P(N = n)\frac{H(Y_{1:n}|N=n,x)}{n} = \frac{1}{2}\log 100 + \frac{1}{2}\log 2$, which yields $\mathrm{BF} \approx 14$. This means that a typical generated trajectory behaves as if each token has roughly 14 plausible continuations. Because BF weights trajectories equally, short but highly diverse generations contribute substantially even if they contain relatively few tokens overall. In contrast, GenPPL treats each *token* equally. $\log \mathrm{GenPPL} = \frac{H(N,Y_{1:N}|x)}{\mathbb{E}[N]} = \frac{H(N|x)+\sum_n P(N=n)\, H(Y_{1:n}|N=n,x)}{\sum_n P(N=n)\, n} = \frac{6\log 2+\log 100}{6}$. Since most generated tokens come from the long low-diversity essays, we obtain $\mathrm{GenPPL} \approx 4$. Thus, a token on average has only about 4 plausible continuations.

The distinction arises from the averaging mechanism. BF averages uncertainty at the trajectory level, while GenPPL averages at the token level and therefore assigns greater weight to longer generations. Consequently, BF reflects the diversity of a *typical trajectory*, whereas GenPPL reflects the uncertainty of a *typical token*. These two measures therefore provide complementary views of generation space that cannot be captured by a single metric alone.

B.2 When to use BF vs. GenPPL across different tasks.

We refer readers to Table 5 for a concise practical comparison of when to use BF versus GenPPL across different tasks.

B.3 Comparison with perplexity.

Perplexity [19] is a widely used evaluation metric in the LLM literature. To avoid confusion, we briefly contrast it with GenPPL and BF. Perplexity is defined on a fixed dataset $(y_1, \ldots, y_T)$, where $y_t \in \mathcal{V}$ is the observed token and $y_{<t}$ denotes its context. Given the model distribution $q(\cdot \mid y_{<t})$, it is defined as $\mathrm{PPL} = \exp\left(\frac{1}{T}\sum_{t=1}^{T} -\log q(y_t \mid y_{<t})\right)$, where $-\log q(y_t \mid y_{<t})$ measures how surprising the true token is under the model. Thus, perplexity evaluates predictive accuracy, with lower values indicating better fit to the data. In contrast, GenPPL and BF are defined with respect to the model's generative distribution and measure generation capacity. Namely, the size and diversity of the space of possible outputs rather than accuracy on the observed data.

Table 5: When to use BF vs. GenPPL across different tasks.

Task	Preferred Metric	Level	Reason
MMLU (QA)	GenPPL	Token	Requires selecting a single correct answer. Performance depends on calibrated token probabilities. Diversity across full trajectories is unnecessary.
Story Generation	BF	Trajectory	Outputs are evaluated as complete units. Diversity across stories reflects creativity and coverage of different narrative possibilities.
Math	GenPPL BF	Token Trajectory	GenPPL is primary because correctness depends on precise step-by-step reasoning. BF is secondary and useful for analyzing diversity across alternative solution strategies.
Coding	GenPPL BF	Token Trajectory	GenPPL is primary because code correctness requires token-level precision. BF is secondary and useful for comparing multiple valid implementations.
Sentence Completion	GenPPL	Token	This is a next-token prediction task, so GenPPL aligns directly with likelihood and perplexity-based evaluation.

C Experiment details

In this section, we provide additional experimental and implementation details underlying our empirical analysis. In particular, we discuss the stopping behavior of autoregressive generation and the resulting truncation bias, as well as the statistical modeling procedures used throughout the paper. We further present detailed outputs from the Beta mixed-effects regression analysis, including coefficient estimates, interaction effects, and dispersion modeling results, together with additional discussions on uncertainty allocation, semantic diversity, and length–entropy dynamics across model families and tasks.

C.1 Stopping criteria and empirical truncation bias

To ensure the generative process remains consistent with the theoretical framework, we include the `EOS` token in the sampling vocabulary $\mathcal{V} \cup \{\texttt{EOS}\}$. Crucially, during top-$k$ sampling, a rollout only terminates if `EOS` is present within the top k most probable tokens and is subsequently sampled.

While our theoretical framework allows for infinite generation length, practical estimation requires a finite truncation point $T_{\max}$. To ensure that our estimator $\widehat{\mathrm{CE}}^{\star}_{\max,P,M}$ remains a consistent proxy for the true Canopy Entropy $\mathrm{CE}^{\star}$, we set a generous token budget of $T_{\max} = 4000$. As shown in Table 4, the empirical truncation rate remains low across all model families and tasks. Notably, fine-tuned models exhibit a 0.00% truncation rate in almost all settings, confirming that aligned models naturally terminate well within the provided budget. While truncation rates are higher for base models, they remain sufficiently low (≤ 0.0432). Overall, these results demonstrate that the truncation bias term $g(T_{\max})$ defined in Theorem 2 is sufficiently small, ensuring that the observed differences in generation space are not artifacts of the maximum length constraint.

C.2 Completion traps

Instead of answering the instruction, base models sometimes treat the prompt as an unfinished text prefix and continue it with a short completion fragment, or terminate immediately when the text appears complete. Here we present some empirical examples where base models fall into such "completion traps". Some examples are listed below.

Prompt:

> Come up with an original and creative solution for the following real-world problem: Clara, a junior pre-med student, is working part-time and taking a 15 hour credit load at school *[...]* Clara is not sure how to solve her problem.

Qwen3-8B Base:

```
"In order to entice donors, you would like to take the
following approach:"
```

Llama-3.1-8B Base:

```
"How can she decide what steps she should take to solve her
problem?"
```

Gemma-3-12B Base:

```
"There will be partial credit for good ideas that might just
work (even if they're not perfect).  Your solution will be
judged on its creativity and how well it works."
```

Coding prompts were especially vulnerable to premature termination caused by structural delimiters. Prompts ending with a closing code block delimiter (‘‘‘) frequently produced empty responses from base models, as they likely interpret the closing backticks as an end-of-document signal from the training distribution. To ensure valid evaluations, we manually stripped these trailing delimiters to force the model to generate the intended logic rather than treating the task as already finalized.

C.3 Chat-style replies for sentence completion

Conversely, instruct models sometimes treat a sentence-completion prompt as a conversational query. Since the unfinished sentence is passed as a user message through the chat template, the models often comment on the scene, offer to elaborate, or ask what the user wants, rather than continuing the text.

C.4 Bootstrap procedure

All standard errors and confidence intervals reported in Table 1, Table 2, and Table 3 are estimated using a prompt-level cluster bootstrap with $B = 2000$ replicates. In each replicate, we sample $P = 100$ prompt indices uniformly with replacement from the original set of prompts. All $M = 100$ rollouts associated with a sampled prompt are resampled jointly as a single cluster, preserving the within-prompt rollout structure.

Point estimates are always computed on the original, non-resampled data; bootstrap replicates are used only for uncertainty quantification. Reported standard errors correspond to the sample standard deviation of the bootstrap replicates with degrees of freedom = $n - 1$.

C.5 Beta mixed-effects model for semantic diversity

The analysis is based on 2400 observations, corresponding to 4 tasks, 3 model families, and 2 model variants (base and instruct), with 100 prompts evaluated for each configuration.

Conditional mean modeling. Formally, for each task–prompt–model tuple (t, p, m), we model $D_{tpm} \sim \text{Beta}(\mu_{tpm}\phi_{tpm}, (1-\mu_{tpm})\phi_{tpm})$, where $\mu_{tpm} \in (0, 1)$ denotes the conditional mean and $\phi_{tpm} > 0$ is the precision parameter. The conditional mean model is specified as $\text{logit}(\mu_{tpm}) = \alpha + \beta_1 R_{tpm} + \beta_2 \mathbf{1}\{m = \text{FT}\} + f_0\left(\text{invN}_{tpm}\right) + \sum_k \theta_k \mathbf{1}\{t = k\} + \beta_3 (R_{tpm} \cdot \mathbf{1}\{m = \text{FT}\}) + f_1\left(\text{invN}_{tpm}\right) \mathbf{1}\{m = \text{FT}\} + u_p + v_m$ where R_{tpm} is the entropy rate, invN_{tpm} is the inverse sequence length, and $f_0(\cdot)$, $f_1(\cdot)$ are natural spline functions corresponding to nonlinear length effects and their interaction with instruction tuning. The coefficients θ_k represent fixed task effects. To account for hierarchical dependence, we include random intercepts $u_p \sim \mathcal{N}(0, \sigma_p^2), v_m \sim \mathcal{N}(0, \sigma_m^2)$, for prompt identity and model family, respectively.

In addition to the finding presented in section 5, we also observe strong nonlinear dependence on inverse sequence length. Several spline terms and their interactions with fine-tuning are significant,

Table 6: Full table of conditional mean model estimates for the Beta mixed-effects regression.

Term	Estimate	Std. Error	p-value
Intercept	-0.900	0.071	$< 2 \times 10^{-16}$
$\boldsymbol{R}$ **($\boldsymbol{\beta_1}$)**	**0.688**	0.033	$< 2 \times 10^{-16}$
Instruct ($\boldsymbol{\beta_2}$)	**-0.325**	0.087	1.87×10^{-4}
$\mathrm{ns}(\mathrm{invN}, 4)_1$	0.169	0.040	2.04×10^{-5}
$\mathrm{ns}(\mathrm{invN}, 4)_2$	0.448	0.236	0.058
$\mathrm{ns}(\mathrm{invN}, 4)_3$	0.368	1.499	0.806
$\mathrm{ns}(\mathrm{invN}, 4)_4$	0.011	3.156	0.997
Coding	-0.035	0.036	0.329
Math	0.180	0.034	1.53×10^{-7}
Stories	-0.252	0.028	$< 2 \times 10^{-16}$
$\boldsymbol{R}\times$ **Instruct ($\boldsymbol{\beta_3}$)**	**1.323**	0.054	$< 2 \times 10^{-16}$
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_1$	-0.099	0.077	0.199
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_2$	1.305	0.333	8.76×10^{-5}
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_3$	1.237	1.514	0.414
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_4$	1.059	3.159	0.737

confirming that the relationship between diversity and length is highly nonlinear and differs between base and fine-tuned models. This reinforces that simple linear controls for length are insufficient.

Task-level effects are also pronounced. Relative to completion, mathematical reasoning exhibits significantly higher semantic diversity ($+0.180$, $p < 10^{-6}$), while story generation shows substantially lower embedding-based diversity (-0.252, $p < 2 \times 10^{-16}$). This suggests that open-ended generation can remain semantically concentrated, despite its greater freedom of expression (see Table 6).

Finally, random effects indicate that prompt-level variability remains larger than model-level variability, highlighting the strong influence of prompt semantics on generation diversity (see Table 7).

Table 7: Estimated random intercept variances for prompt identity and model family in the Beta mixed-effects regression model.

Random effect	Group	Variance	Std. Dev.
Intercept	prompt_uid	0.0127	0.112
Intercept	model_name	0.00714	0.0845

Dispersion modeling. The dispersion model specifies how the precision parameter ϕ_{tpm} varies across model variants, tasks, sequence length, entropy rate, and model family. Specifically, we use the log-link specification $\log(\phi_{tpm}) = \gamma_0 + \gamma_1 \mathbf{1}\{m = \mathrm{FT}\} + \sum_k \delta_k \mathbf{1}\{t = k\} + h(\mathrm{invN}_{tpm}) + \gamma_2 R_{tpm} + \sum_j \eta_j \mathbf{1}\{\mathrm{fam}(m) = j\} + \sum_k \zeta_k \mathbf{1}\{m = \mathrm{FT}\}\mathbf{1}\{t = k\} + \gamma_3 \left(R_{tpm} \cdot \mathbf{1}\{m = \mathrm{FT}\}\right)$, where $h(\cdot)$ is a natural spline function capturing nonlinear length-dependent dispersion effects. The task indicators δ_k allow the residual precision to vary across tasks, while the model-family indicators η_j account for systematic differences in dispersion across model families. The interaction terms between model variant and task allow fine-tuning to affect dispersion differently across tasks, and the interaction between entropy rate and model variant tests whether the relationship between entropy rate and precision changes after instruction tuning.

Empirically, the dispersion model reveals substantial heterogeneity in residual precision. Instruction-tuned models have significantly higher baseline precision than base models ($\hat{\gamma}_1 = 1.315$, $p < 0.001$), indicating lower conditional variability after accounting for the mean structure. Precision also differs strongly by task, with coding, math, and story generation all showing significantly higher precision relative to the reference task. The spline terms for inverse length indicate that dispersion varies nonlinearly with output length, while the positive main effect of entropy rate ($\hat{\gamma}_2 = 1.016$, $p < 0.001$) suggests that higher entropy rate is associated with greater precision among base models. However, the negative interaction between instruction tuning and entropy rate ($\hat{\gamma}_3 = -1.878$, $p < 0.001$) shows that this relationship is substantially weakened or reversed for fine-tuned models. The significant

model-variant-by-task interactions further indicate that fine-tuning changes residual variability in a task-dependent manner (see Table 8).

Table 8: Dispersion model estimates for the Beta mixed-effects regression. The dispersion submodel uses a log link for the precision parameter ϕ_{tpm}.

Term	Estimate	Std. Error	p-value
Intercept	2.658	0.226	$< 2 \times 10^{-16}$
Instruct	1.315	0.255	2.48×10^{-7}
Coding	2.602	0.213	$< 2 \times 10^{-16}$
Math	2.147	0.209	$< 2 \times 10^{-16}$
Stories	1.677	0.148	$< 2 \times 10^{-16}$
$\mathrm{ns}(\mathrm{invN}, 3)_1$	-3.464	0.562	7.20×10^{-10}
$\mathrm{ns}(\mathrm{invN}, 3)_2$	-2.097	0.904	0.020
$\mathrm{ns}(\mathrm{invN}, 3)_3$	2.192	1.916	0.253
R	1.016	0.126	5.97×10^{-16}
Llama-3.1-8B	0.909	0.113	9.16×10^{-16}
Qwen3-8B	0.622	0.114	5.42×10^{-8}
Instruct $\times$ Coding	-2.119	0.288	1.88×10^{-13}
Instruct $\times$ Math	-1.589	0.284	2.14×10^{-8}
Instruct $\times$ Stories	-2.790	0.208	$< 2 \times 10^{-16}$
Instruct $\times R$	-1.878	0.227	$< 2 \times 10^{-16}$

Residual diagnosis. The figure (see Figure 6) presents a DHARMa residual diagnostic QQ plot [15] for the fitted Beta mixed-effects regression model, computed from 10,000 simulated datasets, comparing the empirical residual distribution against the expected uniform distribution. The residuals closely follow the diagonal reference line, indicating good overall model calibration and no substantial systematic deviation from the assumed distribution. The associated diagnostic tests further support model adequacy: the Kolmogorov–Smirnov test ($p = 0.473$) suggests no significant deviation from uniformity, the dispersion test ($p = 0.110$) indicates no significant over- or under-dispersion, and the outlier test ($p = 0.084$) shows no evidence of excessive outliers. We note, however, that while the KS and dispersion results remain stable across random seeds, the outlier test is noticeably more sensitive to the seed used for simulation. Overall, these diagnostics suggest that the fitted model provides an adequate characterization of the observed data and that the regression results are statistically reliable.

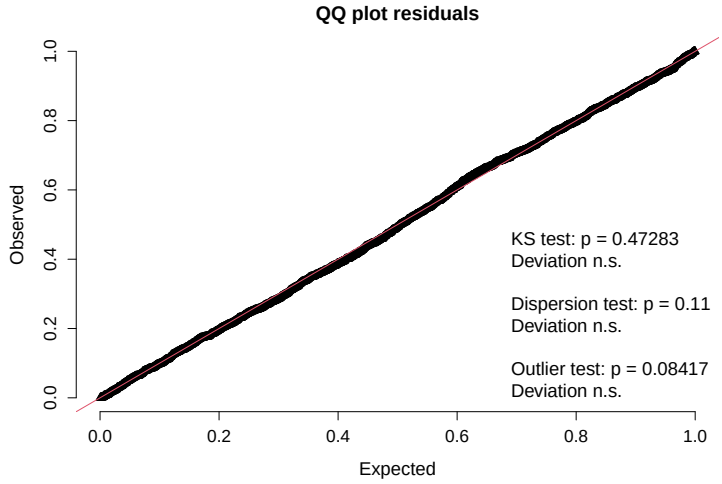

Figure 6: DHARMa residual diagnostics for the fitted Beta mixed-effects regression model. The QQ plot shows no significant deviation from the expected uniform residual distribution.

C.6 Compute resources and runtime

All experiments were run on a single internal SLURM cluster. Each generation job uses one NVIDIA GPU (A100, H100, or H200), 8 CPU cores, and 100 GB host RAM, with a 6-hour wall-clock timeout per job. Overall, we estimate total project compute at approximately 30-40 GPU-hours, overwhelmingly dominated by rollout generation; metric computation and statistical analysis were negligible by comparison and can be completed on CPU.

D Additional robustness and sensitivity analyses

D.1 Sensitivity to Top-k Truncation

Our main experiments estimate next-token entropy using the renormalized distribution over the top $k = 100$ tokens. Since truncating the probability distribution may underestimate entropy, we examine the sensitivity of our findings to this approximation by repeating the same experiments after increasing k to 5000.

Global uncertainty. Table 9 reports the results for GenPPL, BF, and CE* under $k = 5000$. Consistent with our main findings, instruction tuning reduces GenPPL and BF across all 12 model–task comparisons, and the reductions are generally larger than under top-$k = 100$. CE* decreases in 11 of 12 comparisons, with the only exception being Gemma-3-12B on story generation, which shows a small increase of $4.43\% \pm 6.8\%$.

Table 9: Generation-space uncertainty under top-$k = 5000$. Standard errors are estimated via prompt-cluster bootstrapping as in subsection C.4. Blue cells indicate reductions in uncertainty after instruction tuning, while red cells indicate increases.

Task	Type	GenPPL			BF			CE*		
		Qwen3-8B	Llama-3.1-8B	Gemma-3-12B	Qwen3-8B	Llama-3.1-8B	Gemma-3-12B	Qwen3-8B	Llama-3.1-8B	Gemma-3-12B
Math	Base	1.26 ± 0.013	7.76 ± 0.15	6.32 ± 0.087	1.24 ± 0.0067	9.29 ± 0.23	6.34 ± 0.077	65.1 ± 3.7	830 ± 31	593 ± 14
	Instruct	1.17 ± 0.0073	4.95 ± 1.5	1.16 ± 0.0091	1.16 ± 0.0057	1.63 ± 0.052	1.15 ± 0.0052	41.0 ± 2.7	346 ± 84	56.7 ± 4.7
	% Change	-7.26 ± 0.94%	-36.2 ± 20%	-81.6 ± 0.28%	-6.52 ± 0.40%	-82.4 ± 0.65%	-81.9 ± 0.20%	-37.0 ± 4.1%	-58.3 ± 10%	-90.4 ± 0.88%
Coding	Base	1.93 ± 0.022	5.47 ± 0.13	3.39 ± 0.045	2.35 ± 0.043	7.04 ± 0.084	3.88 ± 0.049	288 ± 8.9	551 ± 21	675 ± 33
	Instruct	1.23 ± 0.0054	1.30 ± 0.013	1.04 ± 0.0023	1.21 ± 0.0061	1.28 ± 0.0085	1.04 ± 0.0018	81.9 ± 4.0	138 ± 6.1	16.2 ± 1.0
	% Change	-36.5 ± 0.76%	-76.2 ± 0.57%	-69.2 ± 0.40%	-48.5 ± 0.99%	-81.8 ± 0.24%	-73.1 ± 0.33%	-71.6 ± 1.2%	-74.9 ± 1.3%	-97.6 ± 0.20%
Sentence Completion	Base	14.8 ± 0.78	19.9 ± 0.52	13.8 ± 0.15	7.42 ± 0.31	18.0 ± 0.47	14.4 ± 0.17	1926 ± 110	1631 ± 60	1359 ± 44
	Instruct	1.69 ± 0.027	2.74 ± 0.13	1.79 ± 0.020	1.56 ± 0.018	2.89 ± 0.070	1.72 ± 0.024	114 ± 11	219 ± 14	517 ± 30
	% Change	-88.6 ± 0.57%	-86.2 ± 0.73%	-87.0 ± 0.17%	-79.0 ± 0.81%	-84.0 ± 0.51%	-88.0 ± 0.20%	-94.1 ± 0.48%	-86.6 ± 1.0%	-61.9 ± 2.3%
Story Generation	Base	5.31 ± 0.29	20.1 ± 0.31	11.7 ± 0.11	5.57 ± 0.20	17.9 ± 0.26	12.0 ± 0.10	626 ± 34	1996 ± 53	878 ± 29
	Instruct	2.35 ± 0.043	8.57 ± 0.87	2.26 ± 0.027	2.16 ± 0.043	6.04 ± 0.41	2.20 ± 0.030	469 ± 29	973 ± 60	917 ± 40
	% Change	-55.7 ± 1.8%	-57.4 ± 4.3%	-80.7 ± 0.23%	-61.3 ± 1.4%	-66.3 ± 2.4%	-81.6 ± 0.23%	-25.0 ± 2.8%	-51.2 ± 2.8%	4.43 ± 6.8%

Length–entropy coupling. Table 10 reports the corresponding prompt-controlled length–entropy correlations. $\rho(N, r_N)$ increases after instruction tuning in the same 11 of the 12 model–task comparisons as in the original $k = 100$ setting. This consistency under a substantially larger k suggests that the predominantly positive shift in length–entropy coupling is not an artifact of the original top-100 trucation.

Table 10: Base–instruct comparison of the length–entropy Pearson correlation $\rho(N, r_N)$ under top-$k = 5000$. Each entry reports the estimate ± standard error, with the absolute change computed as the instruct value minus the corresponding base value.

Metric	Model	MATH			CODING			SENTENCE COMPLETION			STORY GENERATION		
		Base	Instruct	Abs. Change	Base	Instruct	Abs. Change	Base	Instruct	Abs. Change	Base	Instruct	Abs. Change
$\rho(N, r_N)$	Qwen	0.021 ± 0.022	0.096 ± 0.039	0.075 ± 0.043	-0.28 ± 0.017	0.62 ± 0.033	0.90 ± 0.036	0.46 ± 0.017	0.52 ± 0.024	0.066 ± 0.029	-0.063 ± 0.020	0.19 ± 0.021	0.25 ± 0.027
	Llama	-0.10 ± 0.017	0.69 ± 0.046	0.80 ± 0.052	-0.20 ± 0.014	0.33 ± 0.026	0.53 ± 0.027	0.12 ± 0.014	-0.18 ± 0.025	-0.29 ± 0.028	0.20 ± 0.015	0.41 ± 0.025	0.21 ± 0.027
	Gemma	0.015 ± 0.016	0.21 ± 0.038	0.19 ± 0.041	-0.15 ± 0.0091	0.39 ± 0.035	0.54 ± 0.034	-0.12 ± 0.014	0.16 ± 0.039	0.27 ± 0.042	-0.034 ± 0.013	-0.022 ± 0.027	0.013 ± 0.029

Semantic diversity. We refit the main Beta mixed-effects regression under $k = 5000$, keeping the same specification. As shown in Table 11, the entropy rate coefficient remains positive ($\beta_1 = 0.562$, $p < 2 \times 10^{-16}$), with a strongly positive entropy-rate × fine-tuning interaction ($\beta_3 = 1.497$, $p < 2 \times 10^{-16}$). The entropy rate slope thus increases from 0.562 for base models to 2.059 for fine-tuned models (3.66×), consistent with our main finding that fine-tuned models translate token

uncertainty into semantic diversity more effectively. The main effect of fine-tuning remains negative, although it is only marginally significant ($\beta_2 = -0.181$, $p = 0.074$). Residual diagnostics show no significant deviations in the KS ($p = 0.16$) or outlier ($p = 0.084$) tests, although the dispersion test detects residual overdispersion ($p < 2.2 \times 10^{-16}$).

Table 11: Full table of conditional mean model estimates for the Beta mixed-effects regression with top-$k = 5000$.

Term	Estimate	Std. Error	p-value
Intercept	-0.823	0.096	$< 2 \times 10^{-16}$
$\boldsymbol{R}$ **($\boldsymbol{\beta_1}$)**	**0.562**	0.037	$< 2 \times 10^{-16}$
Instruct ($\boldsymbol{\beta_2}$)	**-0.181**	0.102	0.074
$\mathrm{ns}(\mathrm{invN}, 4)_1$	0.404	0.038	$< 2 \times 10^{-16}$
$\mathrm{ns}(\mathrm{invN}, 4)_2$	1.080	0.274	7.91×10^{-5}
$\mathrm{ns}(\mathrm{invN}, 4)_3$	-0.621	1.450	0.668
$\mathrm{ns}(\mathrm{invN}, 4)_4$	-3.039	3.076	0.323
Coding	-0.135	0.040	7.19×10^{-4}
Math	0.073	0.032	0.021
Stories	-0.208	0.022	$< 2 \times 10^{-16}$
$\boldsymbol{R}\times$ **Instruct ($\boldsymbol{\beta_3}$)**	**1.497**	0.055	$< 2 \times 10^{-16}$
Instruct $\times \mathrm{ns}(\mathrm{invN}, 4)_1$	-0.479	0.079	1.51×10^{-9}
Instruct $\times \mathrm{ns}(\mathrm{invN}, 4)_2$	1.358	0.346	8.80×10^{-5}
Instruct $\times \mathrm{ns}(\mathrm{invN}, 4)_3$	2.232	1.463	0.127
Instruct $\times \mathrm{ns}(\mathrm{invN}, 4)_4$	3.834	3.076	0.213

D.2 Robustness to few-shot prompting

We extend our analysis to five-shot prompting, where base models exhibit substantially different generation and stopping behaviors compared to the zero-shot setting. Rather than falling into "completion traps" (subsection C.2), base models frequently continue generating additional question–answer pairs, resulting in longer generation lengths and a greater proportion of long trajectories.

Global uncertainty. Table 12 reports the results for GenPPL, BF, and CE* under five-shot prompting. Consistent with our main findings, instruction tuning reduces GenPPL and BF across all 12 model–task comparisons. CE* also decreases in all 12 comparisons, including story generation, where the short few-shot example answers lead fine-tuned models to produce substantially shorter outputs, reducing the contribution of trajectory length to total uncertainty.

Table 12: Generation-space uncertainty under few-shot prompting. Standard errors are estimated via prompt-cluster bootstrapping as in subsection C.4. Blue cells indicate reductions in uncertainty after instruction tuning, while red cells indicate increases.

Task	Type	GenPPL			BF			CE*		
		Qwen3-8B	Llama-3.1-8B	Gemma-3-12B	Qwen3-8B	Llama-3.1-8B	Gemma-3-12B	Qwen3-8B	Llama-3.1-8B	Gemma-3-12B
Math	Base	1.37 ± 0.0023	3.68 ± 0.014	3.86 ± 0.012	1.38 ± 0.0028	3.61 ± 0.014	3.53 ± 0.013	380 ± 3.3	2413 ± 19	2560 ± 22
	Instruct	1.10 ± 0.0041	1.74 ± 0.065	1.15 ± 0.024	1.09 ± 0.0034	1.48 ± 0.021	1.10 ± 0.0070	12.7 ± 0.88	93.6 ± 9.6	17.4 ± 3.4
	% Change	$-19.8 \pm 0.30\%$	$-52.7 \pm 1.8\%$	$-70.3 \pm 0.61\%$	$-20.7 \pm 0.25\%$	$-58.9 \pm 0.54\%$	$-68.7 \pm 0.19\%$	$-96.7 \pm 0.22\%$	$-96.1 \pm 0.39\%$	$-99.3 \pm 0.13\%$
Coding	Base	1.95 ± 0.0082	2.26 ± 0.0098	2.25 ± 0.0070	2.01 ± 0.012	2.10 ± 0.010	2.13 ± 0.0078	564 ± 7.6	828 ± 11	2222 ± 15
	Instruct	1.06 ± 0.0021	1.42 ± 0.010	1.05 ± 0.0020	1.06 ± 0.0019	1.41 ± 0.0092	1.05 ± 0.0018	10.6 ± 0.61	97.0 ± 3.3	11.7 ± 0.69
	% Change	$-45.8 \pm 0.22\%$	$-37.0 \pm 0.45\%$	$-53.3 \pm 0.15\%$	$-47.5 \pm 0.31\%$	$-33.1 \pm 0.38\%$	$-50.7 \pm 0.19\%$	$-98.1 \pm 0.11\%$	$-88.3 \pm 0.43\%$	$-99.5 \pm 0.031\%$
Sentence Completion	Base	6.76 ± 0.032	6.30 ± 0.043	9.26 ± 0.027	6.42 ± 0.036	7.44 ± 0.040	9.45 ± 0.038	3006 ± 36	2157 ± 25	1122 ± 15
	Instruct	1.66 ± 0.0045	2.83 ± 0.022	1.90 ± 0.0098	1.66 ± 0.0047	2.86 ± 0.024	1.86 ± 0.012	367 ± 5.4	298 ± 2.1	448 ± 11
	% Change	$-75.4 \pm 0.12\%$	$-55.1 \pm 0.46\%$	$-79.5 \pm 0.11\%$	$-74.2 \pm 0.16\%$	$-61.5 \pm 0.37\%$	$-80.3 \pm 0.13\%$	$-87.8 \pm 0.20\%$	$-86.2 \pm 0.20\%$	$-60.0 \pm 1.1\%$
Story Generation	Base	4.70 ± 0.053	5.13 ± 0.055	6.98 ± 0.044	5.09 ± 0.067	6.16 ± 0.068	7.02 ± 0.056	323 ± 7.7	1836 ± 30	995 ± 19
	Instruct	1.68 ± 0.015	2.86 ± 0.054	2.30 ± 0.020	1.63 ± 0.014	3.03 ± 0.049	2.19 ± 0.027	64.4 ± 5.2	217 ± 9.0	537 ± 30
	% Change	$-64.2 \pm 0.36\%$	$-44.3 \pm 1.3\%$	$-67.1 \pm 0.40\%$	$-67.9 \pm 0.29\%$	$-50.8 \pm 0.95\%$	$-68.8 \pm 0.37\%$	$-80.1 \pm 1.3\%$	$-88.2 \pm 0.44\%$	$-46.0 \pm 2.5\%$

Length–entropy coupling. As shown in Table 13, $\rho(N, r_N)$ continues to increase after instruction tuning in 9 of the 12 model–task comparisons, despite base models frequently generating additional question–answer pairs that substantially extend their outputs. This suggests that the predominantly positive shift in length–entropy coupling is robust to changes in prompting strategy and generation behavior.

Table 13: Base–instruct comparison of the length–entropy Pearson correlation $\rho(N, r_N)$ under few-shot prompting. Each entry reports the estimate $\pm$ standard error, with the absolute change computed as the instruct value minus the corresponding base value.

Metric	Model	MATH			CODING			SENTENCE COMPLETION			STORY GENERATION		
		Base	Instruct	Abs. Change	Base	Instruct	Abs. Change	Base	Instruct	Abs. Change	Base	Instruct	Abs. Change
$\rho(N, r_N)$	Qwen	-0.053 ± 0.014	0.14 ± 0.044	0.20 ± 0.045	-0.045 ± 0.0088	0.37 ± 0.042	0.41 ± 0.042	0.11 ± 0.0095	0.16 ± 0.016	0.044 ± 0.018	-0.25 ± 0.016	0.086 ± 0.021	0.34 ± 0.030
	Llama	0.095 ± 0.011	0.49 ± 0.034	0.40 ± 0.036	0.33 ± 0.014	0.29 ± 0.021	-0.040 ± 0.024	-0.44 ± 0.0093	-0.086 ± 0.014	0.35 ± 0.016	-0.44 ± 0.012	-0.22 ± 0.017	0.22 ± 0.016
	Gemma	0.51 ± 0.0090	0.23 ± 0.040	-0.28 ± 0.042	0.58 ± 0.010	0.49 ± 0.035	-0.083 ± 0.035	-0.12 ± 0.013	0.31 ± 0.024	0.43 ± 0.027	-0.043 ± 0.012	0.11 ± 0.023	0.16 ± 0.023

Semantic diversity. The substantially different generation behaviors under five-shot prompting introduce additional heterogeneity in response lengths and semantic diversity across model–task combinations. We therefore adapt the model to better account for these differences:

$$\begin{aligned}\operatorname{logit}(\mu_{tpm}) &= \beta_1 R_{tpm} + \beta_3 R_{tpm}\mathbf{1}\{m = \mathrm{FT}\} \\ &\quad + f_0(\log N_{tpm}) + f_1(\log N_{tpm})\mathbf{1}\{m = \mathrm{FT}\} \\ &\quad + \alpha_{tm} + \psi s_{tpm} + u_p, \\ \log(\phi_{tpm}) &= \delta_{tm} + h(\log N_{tpm}) + \gamma_2 R_{tpm} \\ &\quad + \gamma_3 R_{tpm}\mathbf{1}\{m = \mathrm{FT}\} + \kappa s_{tpm}.\end{aligned}$$

Here, (t, p, m) indexes the task, prompt, and model, respectively. We introduce model–task-specific intercepts in both the conditional mean and dispersion models, α_{tm} and δ_{tm}, respectively, to account for differences in baseline semantic diversity across model–task combinations and in its variability across prompts within each combination. We also replace inverse sequence length with log sequence length to better distinguish long generations, as output lengths vary substantially more under five-shot prompting. The additional term s_{tpm}, the standard deviation of log sequence lengths across rollouts for each task–prompt–model combination, captures within-prompt length variability arising from the mixture of shorter responses and extended question–answer continuations. As before, R_{tpm}, $\log N_{tpm}$, and s_{tpm} are standardized, and f_0, f_1, and h denote natural spline functions.

As shown in Table 14, the entropy rate coefficient remains positive ($\beta_1 = 0.115$, $p = 5.28 \times 10^{-6}$), with a strongly positive entropy rate $\times$ fine-tuning interaction ($\beta_3 = 0.483$, $p < 2 \times 10^{-16}$). The entropy rate slope thus increases from 0.115 for base models to 0.598 for fine-tuned models ($5.2\times$), consistent with our main finding that fine-tuned models exhibit a stronger relationship between entropy rate and semantic diversity. Residual diagnostics indicate an adequate fit (KS $p = 0.13$, dispersion $p = 0.81$, outlier $p = 0.084$).

Table 14: Conditional mean model estimates for the Beta mixed-effects regression under few-shot prompting, omitting the model–task intercepts α_{tm} (24 levels).

Term	Estimate	Std. Error	p-value
$\boldsymbol{R}$ **($\boldsymbol{\beta_1}$)**	**0.115**	0.025	5.28×10^{-6}
$\boldsymbol{R}\times$ **Instruct ($\boldsymbol{\beta_3}$)**	**0.483**	0.059	$< 2 \times 10^{-16}$
Length spread s (ψ)	0.282	0.009	$< 2 \times 10^{-16}$
$\mathrm{ns}(\log N, 4)_1$	-0.277	0.250	0.269
$\mathrm{ns}(\log N, 4)_2$	-0.471	0.175	0.007
$\mathrm{ns}(\log N, 4)_3$	-0.469	0.501	0.349
$\mathrm{ns}(\log N, 4)_4$	-1.175	0.112	$< 2 \times 10^{-16}$
Instruct $\times\mathrm{ns}(\log N, 4)_1$	0.192	0.282	0.497
Instruct $\times\mathrm{ns}(\log N, 4)_2$	-1.970	0.261	4.66×10^{-14}
Instruct $\times\mathrm{ns}(\log N, 4)_3$	3.292	1.049	0.002
Instruct $\times\mathrm{ns}(\log N, 4)_4$	7.576	1.514	5.59×10^{-7}

D.3 Robustness of the entropy rate–semantic diversity relationship

D.3.1 Alternative regression specifications

We assess sensitivity to treating model family as a random effect by replacing random intercepts v_m with fixed family effects, keeping all other terms unchanged. Here, (t, p, m) indexes the task, prompt,

and model, respectively. The conditional mean model is then specified as

$$\begin{aligned}\operatorname{logit}(\mu_{tpm}) = \alpha + \beta_1 R_{tpm} + \beta_2 \mathbf{1}\{m = \mathrm{FT}\} + f_0(\mathrm{invN}_{tpm}) \\ + \sum_k \theta_k \mathbf{1}\{t = k\} + \beta_3 (R_{tpm} \cdot \mathbf{1}\{m = \mathrm{FT}\}) \\ + f_1(\mathrm{invN}_{tpm}) \mathbf{1}\{m = \mathrm{FT}\} + u_p + \sum_j \omega_j \mathbf{1}\{\mathrm{fam}(m) = j\},\end{aligned}$$

where ω_j are the fixed effect coefficients for model family $\mathrm{fam}(m) = j \in \{$ Llama-3.1-8B, Qwen3-8B $\}$, with Gemma-3-12B as the reference. The dispersion model remains unchanged.

As shown in Table 15, the entropy rate coefficient remains positive ($\beta_1 = 0.686$, $p < 2 \times 10^{-16}$), with a strongly positive entropy-rate $\times$ fine-tuning interaction ($\beta_3 = 1.321$, $p < 2 \times 10^{-16}$). The entropy rate slope thus increases from 0.686 for base models to 2.007 for fine-tuned models ($2.93\times$), and the main effect of fine-tuning remains negative ($\beta_2 = -0.330$, $p = 1.59 \times 10^{-4}$). The KS test indicates well-behaved residuals ($p = 0.35$), while the dispersion ($p = 0.038$) and outlier ($p = 0.0015$) tests detect some deviations.

Table 15: Full table of conditional mean model estimates for the Beta mixed-effects regression with fixed model-family effects.

Term	Estimate	Std. Error	p-value
Intercept	-0.779	0.052	$< 2 \times 10^{-16}$
$\boldsymbol{R}$ **($\boldsymbol{\beta_1}$)**	**0.686**	0.034	$< 2 \times 10^{-16}$
Instruct ($\boldsymbol{\beta_2}$)	**-0.330**	0.087	1.59×10^{-4}
$\mathrm{ns}(\mathrm{invN}, 4)_1$	0.171	0.040	1.59×10^{-5}
$\mathrm{ns}(\mathrm{invN}, 4)_2$	0.456	0.236	0.053
$\mathrm{ns}(\mathrm{invN}, 4)_3$	0.367	1.497	0.807
$\mathrm{ns}(\mathrm{invN}, 4)_4$	-0.009	3.151	0.998
Coding	-0.037	0.037	0.318
Math	0.179	0.034	2.06×10^{-7}
Stories	-0.252	0.028	$< 2 \times 10^{-16}$
Llama-3.1-8B	-0.177	0.018	$< 2 \times 10^{-16}$
Qwen3-8B	-0.192	0.031	3.76×10^{-10}
$\boldsymbol{R}\times$ **Instruct ($\boldsymbol{\beta_3}$)**	**1.321**	0.054	$< 2 \times 10^{-16}$
Instruct $\times \mathrm{ns}(\mathrm{invN}, 4)_1$	-0.097	0.077	0.211
Instruct $\times \mathrm{ns}(\mathrm{invN}, 4)_2$	1.310	0.333	8.17×10^{-5}
Instruct $\times \mathrm{ns}(\mathrm{invN}, 4)_3$	1.246	1.512	0.409
Instruct $\times \mathrm{ns}(\mathrm{invN}, 4)_4$	1.072	3.154	0.734

D.3.2 Alternative embedding choices

ModernBERT with clustering prefix. To assess sensitivity to the embedding prefix, we recompute semantic diversity D_{tpm} using the same model modernbert-embed-large by replaceing the original `"search_document:"` prefix with `"clustering:"`, which is designed for grouping semantically similar texts, while keeping the regression specification unchanged.

As shown in Table 16, the entropy-rate coefficient remains positive ($\beta_1 = 0.515$, $p < 2 \times 10^{-16}$), with a strongly positive entropy-rate $\times$ fine-tuning interaction ($\beta_3 = 1.106$, $p < 2 \times 10^{-16}$). The entropy-rate slope thus increases from 0.515 for base models to 1.621 for fine-tuned models ($3.15\times$), while the main effect of fine-tuning remains negative ($\beta_2 = -0.149$, $p = 0.027$). Residual diagnostics show no significant departures in the KS ($p = 0.085$) or dispersion ($p = 0.68$) tests, while the outlier test detects a small deviation ($p = 0.013$).

E5-Mistral. To further examine whether our findings depend on the embedding model, we recompute semantic diversity D_{tpm} using e5-mistral-7b-instruct [40] with the instruction `"Instruct: Retrieve semantically similar text.\nQuery:"`, as used in the original E5-Mistral semantic similarity evaluation, and keep the regression specification unchanged.

Table 16: Full table of conditional mean model estimates for the Beta mixed-effects regression with semantic diversity computed using the ModernBERT clustering prefix.

Term	Estimate	Std. Error	p-value
Intercept	-1.472	0.045	$< 2 \times 10^{-16}$
$\boldsymbol{R}$ **($\boldsymbol{\beta_1}$)**	**0.515**	0.025	$< 2 \times 10^{-16}$
Instruct ($\boldsymbol{\beta_2}$)	**-0.149**	0.068	0.027
$\mathrm{ns}(\mathrm{invN}, 4)_1$	-0.032	0.030	0.284
$\mathrm{ns}(\mathrm{invN}, 4)_2$	0.137	0.171	0.423
$\mathrm{ns}(\mathrm{invN}, 4)_3$	-0.331	1.119	0.767
$\mathrm{ns}(\mathrm{invN}, 4)_4$	-0.566	2.357	0.810
Coding	0.258	0.029	$< 2 \times 10^{-16}$
Math	0.171	0.026	7.93×10^{-11}
Stories	-0.160	0.023	1.29×10^{-12}
$\boldsymbol{R}\times$ **Instruct ($\boldsymbol{\beta_3}$)**	**1.106**	0.047	$< 2 \times 10^{-16}$
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_1$	-0.346	0.065	1.11×10^{-7}
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_2$	0.701	0.346	0.043
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_3$	2.132	1.471	0.147
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_4$	4.421	3.165	0.162

As shown in Table 17, the entropy rate coefficient remains positive ($\beta_1 = 0.295$, $p < 2 \times 10^{-16}$), with a strongly positive entropy-rate $\times$ fine-tuning interaction ($\beta_3 = 1.061$, $p < 2 \times 10^{-16}$). The entropy rate slope thus increases from 0.295 for base models to 1.356 for fine-tuned models ($4.60\times$), while the main effect of fine-tuning remains negative ($\beta_2 = -0.486$, $p = 2.43 \times 10^{-12}$). Residual diagnostics show no significant deviations in the KS ($p = 0.079$), dispersion ($p = 0.81$), or outlier ($p = 0.38$) tests.

Table 17: Full table of conditional mean model estimates for the Beta mixed-effects regression with semantic diversity computed using e5-mistral-7b-instruct embeddings.

Term	Estimate	Std. Error	p-value
Intercept	-1.131	0.044	$< 2 \times 10^{-16}$
$\boldsymbol{R}$ **($\boldsymbol{\beta_1}$)**	**0.295**	0.025	$< 2 \times 10^{-16}$
Instruct ($\boldsymbol{\beta_2}$)	**-0.486**	0.069	2.43×10^{-12}
$\mathrm{ns}(\mathrm{invN}, 4)_1$	0.121	0.026	4.28×10^{-6}
$\mathrm{ns}(\mathrm{invN}, 4)_2$	0.141	0.149	0.342
$\mathrm{ns}(\mathrm{invN}, 4)_3$	1.326	0.765	0.083
$\mathrm{ns}(\mathrm{invN}, 4)_4$	2.387	1.630	0.143
Coding	-0.196	0.026	4.59×10^{-14}
Math	-0.181	0.023	1.18×10^{-14}
Stories	-0.129	0.017	1.09×10^{-14}
$\boldsymbol{R}\times$ **Instruct ($\boldsymbol{\beta_3}$)**	**1.061**	0.039	$< 2 \times 10^{-16}$
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_1$	-0.171	0.060	0.004
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_2$	1.304	0.258	4.16×10^{-7}
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_3$	-0.140	0.979	0.886
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_4$	-0.617	2.110	0.770

D.3.3 Alternative semantic diversity metrics

Normalized Vendi Score. As an alternative to average pairwise cosine distance, we evaluate semantic diversity using the Vendi Score [11], which measures the effective number of distinct outputs based on the eigenvalue distribution of their similarity matrix. For each task–prompt–model tuple (t, p, m), let $K \in \mathbb{R}^{M \times M}$ denote the cosine-similarity matrix of the M rollout embeddings,

and let $\xi_1, \ldots, \xi_M$ be the eigenvalues of K/M. The Vendi Score is then defined as

$$\mathrm{VS}_{tpm} = \exp\left(-\sum_{i=1}^{M} \xi_i \log \xi_i\right),$$

ranging from 1 for identical outputs to M for mutually orthogonal embeddings. It captures the overall similarity structure among M rollouts rather than individual pairwise relationships. We then normalize the score via $(\mathrm{VS}_{tpm} - 1)/(M - 1)$ to $[0, 1]$ for the Beta regression with specifications unchanged.

As shown in Table 18, the entropy rate coefficient remains positive ($\beta_1 = 0.576$, $p < 2 \times 10^{-16}$), with a strongly positive entropy-rate $\times$ fine-tuning interaction ($\beta_3 = 1.040$, $p < 2 \times 10^{-16}$). The entropy rate slope thus increases from 0.576 for base models to 1.615 for fine-tuned models ($2.81\times$), while the main effect of fine-tuning remains negative ($\beta_2 = -1.157$, $p < 2 \times 10^{-16}$). Residual diagnostics show no significant deviation in the KS test ($p = 0.070$), while the dispersion ($p = 4 \times 10^{-4}$) and outlier ($p = 0.0015$) tests detect some deviations.

Table 18: Full table of conditional mean model estimates for the Beta mixed-effects regression with the normalized Vendi Score as the semantic diversity measure.

Term	Estimate	Std. Error	p-value
Intercept	-2.517	0.062	$< 2 \times 10^{-16}$
$\boldsymbol{R}$ **($\boldsymbol{\beta_1}$)**	**0.576**	0.032	$< 2 \times 10^{-16}$
Instruct ($\boldsymbol{\beta_2}$)	**-1.157**	0.089	$< 2 \times 10^{-16}$
$\mathrm{ns}(\mathrm{invN}, 4)_1$	0.150	0.049	0.002
$\mathrm{ns}(\mathrm{invN}, 4)_2$	-0.433	0.348	0.214
$\mathrm{ns}(\mathrm{invN}, 4)_3$	2.202	1.881	0.242
$\mathrm{ns}(\mathrm{invN}, 4)_4$	4.904	4.010	0.221
Coding	-0.727	0.041	$< 2 \times 10^{-16}$
Math	-0.619	0.042	$< 2 \times 10^{-16}$
Stories	-0.304	0.036	$< 2 \times 10^{-16}$
$\boldsymbol{R}\times$ **Instruct ($\boldsymbol{\beta_3}$)**	**1.040**	0.044	$< 2 \times 10^{-16}$
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_1$	0.169	0.078	0.030
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_2$	0.795	0.402	0.048
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_3$	2.112	1.978	0.286
Instruct $\times\mathrm{ns}(\mathrm{invN}, 4)_4$	2.215	4.224	0.600

Relaxed Semantic Entropy. As an embedding-free alternative, we further evaluate semantic diversity using Relaxed Semantic Entropy [2], which groups semantically equivalent outputs into clusters and measures the entropy over the resulting clusters. For each task–prompt–model tuple (t, p, m), we sample $\tilde{M} = 24$ of the $M = 100$ rollouts and use a few-shot prompted LLM judge GPT-5.6-Luna to assess pairwise semantic similarity. Two responses are connected only if the judge considers them similar in both presentation orders, and semantic clusters are identified as the connected components of the resulting graph. We then compute Relaxed Semantic Entropy based on the empirical proportions of rollouts belonging to each cluster:

$$\mathrm{RSE}_{tpm} = -\sum_{c} \frac{|c|}{\tilde{M}} \log \frac{|c|}{\tilde{M}},$$

where $|c|$ denotes the number of rollouts in semantic cluster c. The score ranges from 0 when all rollouts belong to one semantic cluster to $\log \tilde{M}$ when every rollout forms a distinct cluster. We then normalize the score via $\mathrm{RSE}_{tpm}/\log \tilde{M}$ to $[0, 1]$ for regression analysis. [7]

We model normalized RSE using ordered-beta mixed-effects regression [21], which accommodates both continuous values and exact boundary observations at 0 and 1. Moreover, since the LLM judge evaluates semantic equivalence in the context of each prompt, the resulting RSE can be influenced by prompt-specific judging criteria. To account for this, we decompose entropy rate and inverse

[7] RSE results use an earlier rollout set with minor prompt tokenization differences.

sequence length into within-prompt and between-prompt components, distinguishing variation among *different models on the same prompt* from variation across *different prompts*.

Specifically, for each task–prompt pair (t, p), let R^{b}_{tp} and $\mathrm{invN}^{\mathrm{b}}_{tp}$ denote the averages of the standardized R_{tpm} and invN_{tpm} across the six models, where each model's values are computed over all M rollouts, and define $R^{\mathrm{w}}_{tpm} = R_{tpm} - R^{\mathrm{b}}_{tp}$ and $\mathrm{invN}^{\mathrm{w}}_{tpm} = \mathrm{invN}_{tpm} - \mathrm{invN}^{\mathrm{b}}_{tp}$ as the within-prompt deviations. The conditional mean model is then specified as

$$\begin{aligned}\mathrm{logit}(\mu_{tpm}) = \alpha &+ \beta_1 R^{\mathrm{w}}_{tpm} + \beta_2 \mathbf{1}\{m = \mathrm{FT}\} + \beta_3 (R^{\mathrm{w}}_{tpm} \cdot \mathbf{1}\{m = \mathrm{FT}\}) \\ &+ \beta_4 R^{\mathrm{b}}_{tp} + \beta_5 (R^{\mathrm{b}}_{tp} \cdot \mathbf{1}\{m = \mathrm{FT}\}) + f_0(\mathrm{invN}^{\mathrm{w}}_{tpm}) \\ &+ f_1(\mathrm{invN}^{\mathrm{w}}_{tpm}) \mathbf{1}\{m = \mathrm{FT}\} + \beta_6 \mathrm{invN}^{\mathrm{b}}_{tp} \\ &+ \sum_k \theta_k \mathbf{1}\{t = k\} + \sum_j \omega_j \mathbf{1}\{\mathrm{fam}(m) = j\} + u_p,\end{aligned}$$

where f_0 and f_1 are natural splines with three degrees of freedom, ω_j are model-family fixed-effect coefficients with Gemma-3-12B as the reference, and u_p is a random prompt intercept. We also use a reduced dispersion model that excludes the length and entropy-rate terms:

$$\begin{aligned}\log(\phi_{tpm}) = \gamma_0 &+ \gamma_1 \mathbf{1}\{m = \mathrm{FT}\} + \sum_k \delta_k \mathbf{1}\{t = k\} \\ &+ \sum_j \eta_j \mathbf{1}\{\mathrm{fam}(m) = j\} + \sum_k \zeta_k \mathbf{1}\{m = \mathrm{FT}\} \mathbf{1}\{t = k\}.\end{aligned}$$

As shown in Table 19, the within-prompt entropy rate coefficient is positive ($\beta_1 = 0.659$, $p < 2 \times 10^{-16}$), with a positive entropy-rate $\times$ fine-tuning interaction ($\beta_3 = 0.446$, $p = 5.20 \times 10^{-4}$). The within-prompt entropy rate slope thus increases from 0.659 for base models to 1.104 for fine-tuned models ($1.68\times$), while the main effect of fine-tuning is negative ($\beta_2 = -2.910$, $p = 0.037$), consistent with our main findings. Residual diagnostics show no significant deviation in the KS test ($p = 0.11$) and no outliers ($p = 1$), while the dispersion test indicates some residual underdispersion ($p < 2.2 \times 10^{-16}$).

Table 19: Full table of conditional mean model estimates for the ordered Beta mixed-effects regression with relaxed semantic entropy as the semantic diversity measure.

Term	Estimate	Std. Error	p-value
Intercept	-0.293	0.510	0.565
R^{w} (β_1)	**0.659**	0.073	$< 2 \times 10^{-16}$
Instruct (β_2)	**-2.910**	1.398	0.037
R^{b} (β_4)	-0.287	0.290	0.322
$\mathrm{ns}(\mathrm{invN}^{\mathrm{w}}, 3)_1$	-8.125	0.661	$< 2 \times 10^{-16}$
$\mathrm{ns}(\mathrm{invN}^{\mathrm{w}}, 3)_2$	37.842	7.630	7.06×10^{-7}
$\mathrm{ns}(\mathrm{invN}^{\mathrm{w}}, 3)_3$	76.677	15.205	4.59×10^{-7}
$\mathrm{invN}^{\mathrm{b}}$ (β_6)	0.150	0.074	0.044
Coding	0.599	0.274	0.029
Math	-0.641	0.253	0.011
Stories	0.049	0.105	0.640
Llama-3.1-8B	-0.062	0.064	0.330
Qwen3-8B	0.082	0.072	0.255
$R^{\mathrm{w}} \times$ Instruct (β_3)	**0.446**	0.128	5.20×10^{-4}
$R^{\mathrm{b}} \times$ Instruct (β_5)	-0.546	0.097	1.78×10^{-8}
Instruct $\times \mathrm{ns}(\mathrm{invN}^{\mathrm{w}}, 3)_1$	8.867	0.990	$< 2 \times 10^{-16}$
Instruct $\times \mathrm{ns}(\mathrm{invN}^{\mathrm{w}}, 3)_2$	-31.941	8.309	1.21×10^{-4}
Instruct $\times \mathrm{ns}(\mathrm{invN}^{\mathrm{w}}, 3)_3$	-73.760	15.356	1.56×10^{-6}

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The proposed metrics and theoretical framework are introduced in section 3, and the empirical findings supporting the main claims are presented in section 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of our work in section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Assumptions are discussed in section 4 with the complete proof detailed in Appendix A.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Experiment design with hyperparameters is included in section 5, with more details in Appendix C. We also provide the complete codebase and full experimental configurations in the supplementary material.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We provide the complete codebase for all experiments and full experimental configurations in the supplementary material. Detailed setup instructions, required environments, and execution commands are documented in the README.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Experiment design with hyperparameters is included in section 5, with more details in Appendix C. We also provide the complete codebase, prompts, and full experimental configurations in the supplementary material.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: All result tables report statistical quantities including standard errors, confidence intervals, and p-values, supported by bootstrap analyses and residual diagnostic tests.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We document the compute resources and runtime in subsection C.6.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The study utilizes publicly available models and data, ensuring research integrity and transparency. No human subjects, private data, or ethically sensitive collection methods were employed.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss how our framework improves the understanding of LLM alignment and information efficiency. We also explicitly address limitations regarding factuality and human preference to prevent misinterpretation of our metrics in section 6.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This work evaluates existing publicly released model families on public benchmarks. It does not introduce new pre-trained models or scraped datasets.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All utilized model families and task benchmarks are explicitly credited and cited in the References section.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We provide the complete codebase for all experiments and full experimental configurations in the supplementary material. Detailed setup instructions, required environments, and execution commands are documented in the README.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This research consists entirely of information-theoretic analyses and model evaluations on standard benchmarks. No human subjects or crowdsourcing platforms were used in any part of this work.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: This research does not involve crowdsourcing or research with human subjects; it is strictly an information-theoretic analysis of existing large language models.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: The study evaluates LLaMA-3.1-8B, Qwen3-8B, and Gemma-3-12B as the primary objects of analysis, using them to generate the generation trajectories required for our empirical findings. All model versions and decoding configurations are explicitly disclosed in section 5.

Guidelines:

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