LLMs are Bayesian, In Expectation, Not Realization

Leon Chlon Hassana Labs leo@hassana.io
Sarah Rashidi Hassana Labs sarah@hassana.io
Zein Khamis Hassana Labs zein@hassana.io
MarcAntonio M. Awada Harvard University mawada@hbs.edu

Abstract

Large language models demonstrate remarkable in-context learning capabilities, adapting to new tasks without parameter updates. While this phenomenon has been successfully modeled as implicit Bayesian inference, recent empirical findings reveal a fundamental contradiction: transformers systematically violate the martingale property, a cornerstone requirement of Bayesian updating on exchangeable data. This violation challenges the theoretical foundations underlying uncertainty quantification in critical applications.

We resolve this paradox through an information-theoretic framework that reconciles architectural constraints with statistical optimality. We prove that positional encodings fundamentally alter the learning problem: transformers minimize expected conditional Kolmogorov complexity $\mathbb{E}_\pi[K(X|\pi)]$ over permutations rather than the exchangeable complexity $K(X)$. This distinction explains how transformers can simultaneously violate martingale properties while achieving Bayesian-level compression efficiency.

Our theoretical analysis establishes four key results: (1) positional encodings induce martingale violations of order $\Theta(\log n/n)$; (2) transformers achieve information-theoretic optimality with excess risk $O(n^{-1/2})$ in expectation over orderings; (3) the implicit posterior representation converges to the true Bayesian posterior in the space of sufficient statistics; and (4) we derive the optimal chain-of-thought length as $k^* = \Theta(\sqrt{n}\log(1/\varepsilon))$ with explicit constants, providing a principled approach to reduce inference costs while maintaining performance. Empirical validation on GPT-3 confirms predictions (1)-(3), with transformers reaching 99% of theoretical entropy limits within 20 examples. Our framework provides practical methods for extracting calibrated uncertainty estimates from position-aware architectures and optimizing computational efficiency in deployment.

1 Introduction

The emergence of in-context learning (ICL) represents a paradigm shift in machine learning. Large language models, exemplified by GPT-3 [2], can adapt to novel tasks using only a few examples provided at inference time, without any gradient-based parameter updates. This capability has profound implications for few-shot learning, task adaptation, and the fundamental nature of learning in neural networks.

1.1 The Bayesian Framework and Its Success

A particularly elegant theoretical framework interprets ICL through the lens of Bayesian inference. [17] proposed that transformers implicitly perform posterior updates over a latent concept variable, with the pretraining distribution encoding a prior over possible tasks. This perspective has been

extended to show that transformers can implement optimal statistical procedures [1], approximate Gaussian processes [9], and achieve minimax-optimal regret bounds [19].

The Bayesian interpretation provides both conceptual clarity and practical benefits. It suggests principled approaches to uncertainty quantification, explains the sample efficiency of few-shot learning, and connects ICL to the rich literature on meta-learning and statistical estimation theory. The framework's predictive success has made it a cornerstone of our theoretical understanding of transformer capabilities.

1.2 The Martingale Violation Challenge

However, this theoretical edifice was recently challenged by [3], who demonstrated empirically that transformer-based language models systematically violate the martingale property. For exchangeable data where the order of observations carries no information, Bayesian posterior predictive distributions must satisfy:

$$\mathbb{E}[f(X_{n+1})|X_1, \ldots, X_n] = \mathbb{E}[f(X_{n+1})|X_{\pi(1)}, \ldots, X_{\pi(n)}] \tag{1}$$

for any permutation π and bounded function f. This property is not a technical detail but a fundamental mathematical consequence of Bayesian updating.

Their experiments on GPT-3.5, GPT-4, Llama-2, and other state-of-the-art models revealed consistent violations across multiple statistical tests. These findings pose a serious challenge: if transformers violate the martingale property, can they truly be performing Bayesian inference? The implications extend beyond theoretical aesthetics to practical applications in medicine, finance, and other domains where calibrated uncertainty estimates are critical.

1.3 Our Contribution: An Information-Theoretic Resolution

We propose that this apparent contradiction can be resolved by adopting an algorithmic information theory perspective. Our key insight is that positional encodings, which are ubiquitous in transformer architectures, fundamentally alter the information-theoretic structure of the learning problem. While classical Bayesian inference assumes exchangeable data, positional encodings explicitly break this symmetry by making the model's computations depend on the order of inputs.

We formalize this through the distinction between two complexity measures:

- The Kolmogorov complexity $K(X)$ of a sequence, which is permutation-invariant for exchangeable data
- The conditional complexity $K(X|\pi)$ given a specific ordering π

We prove that transformers with positional encodings minimize:

$$\mathbb{E}_{\pi \sim \mathcal{U}(S_n)}[K(X|\pi)] = K(X) + I(X; \pi) \tag{2}$$

where $\mathcal{U}(S_n)$ denotes the uniform distribution over permutations consistent with sufficient statistics, and $I(X; \pi)$ represents the mutual information between sequences and their orderings.

This formulation reveals why transformers can simultaneously:

1. Violate martingale properties (which require identical behavior across all orderings)
2. Achieve near-optimal compression rates characteristic of Bayesian inference
3. Implement implicit posterior representations in the space of sufficient statistics

1.4 Summary of Results

Our main contributions are:

1. Theoretical Reconciliation: We provide the first rigorous explanation for the coexistence of martingale violations and Bayesian-like behavior. We quantify martingale violations as $\Theta(\log n/n)$ and prove that transformers achieve Minimum Description Length (MDL) optimality with excess risk $O(n^{-1/2})$.

2. Information-Theoretic Framework: We establish that transformers are "Bayesian in expectation, not in realization." They achieve optimal compression when averaged over orderings while necessarily violating exchangeability for any specific ordering due to architectural constraints.

3. Optimal Chain-of-Thought Length: We derive a closed-form expression for the optimal number of intermediate reasoning tokens: $k^* = \Theta(\sqrt{n}\log(1/\varepsilon))$ with explicit constants. This result has immediate practical implications for reducing inference costs, which is a critical concern as chain-of-thought prompting becomes standard practice but can increase computational expenses by 10-100× per query. Moreover, we prove an incompleteness theorem showing that chain-of-thought is not just useful but theoretically necessary for transformers to compute functions with complexity exceeding their parameter count.

4. Empirical Validation: Through experiments on GPT-3, we demonstrate that:

- Martingale violations follow our predicted $\Theta(\log n/n)$ scaling with $R^2 > 0.75$
- Transformers reach 99% of theoretical entropy limits within 20 examples
- Position-dependent processing enhances rather than hinders statistical efficiency

5. Practical Algorithms: We provide concrete methods for extracting calibrated uncertainty estimates:

- Permutation averaging achieves $4\times$ variance reduction with $k \approx 20$ shuffles
- Sufficient statistic conditioning reduces position bias by $\approx 85\%$
- Debiasing techniques can mitigate periodic artifacts from rotary embeddings
- Algorithm for computing optimal CoT length with stability guarantees

1.5 Paper Organization

Section 2 reviews the relevant background on in-context learning, Bayesian interpretations, and the martingale critique. Section 3 presents our main theoretical results, including the characterization of martingale violations and the MDL optimality analysis. Section 4 derives the optimal chain-of-thought length with explicit constants and finite-sample guarantees. Section 5 provides empirical validation through controlled experiments. Section 6 discusses implications and limitations. Detailed proofs appear in the appendix.

2 Background and Related Work

2.1 In-Context Learning: Empirical Phenomena and Mechanistic Understanding

In-context learning enables large language models to adapt to new tasks using only examples provided in the input prompt. Formally, given example pairs $(x_1, y_1), \ldots, (x_n, y_n)$ and a query x_{n+1}, an ICL-capable model produces:

$$\hat{y}_{n+1} = f_\theta(x_1, y_1, \ldots, x_n, y_n, x_{n+1}) \tag{3}$$

where f_θ represents the transformer with fixed, pretrained parameters θ.

Recent mechanistic studies have identified key architectural components underlying ICL. [10] discovered "induction heads," which are attention patterns that copy tokens based on previous occurrences, emerging during a phase transition in training. [15] demonstrated that transformer forward passes can implement gradient descent, suggesting ICL may involve implicit optimization. [4] showed that transformers trained from scratch learn to perform linear regression, decision trees, and other algorithms in-context, matching optimal estimator performance.

2.2 The Bayesian Interpretation: Theoretical Foundations and Extensions

The Bayesian framework for ICL, introduced by [17], decomposes the pretraining distribution as:

$$p_{\text{pretrain}}(x_1, y_1, \ldots, x_n, y_n) = \int p(x_1, y_1, \ldots, x_n, y_n|\theta)p(\theta)d\theta \tag{4}$$

where $p(\theta)$ represents a prior over tasks induced by pretraining data.

During inference, the model approximates the posterior predictive distribution:

$$p(y_{n+1}|x_{n+1}, \mathcal{D}_n) = \int p(y_{n+1}|x_{n+1}, \theta)p(\theta|\mathcal{D}_n)d\theta \tag{5}$$

where $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$ denotes the in-context examples.

This framework has been extended in several directions:

- [9] introduced Prior-Data Fitted Networks that directly approximate Bayesian posteriors
- [19] proved that ICL implements Bayesian model averaging with optimal regret bounds
- [1] showed transformers can select and implement appropriate statistical estimators based on data characteristics

2.3 The Martingale Property and Its Violations

For exchangeable sequences where $p(x_1, \ldots, x_n) = p(x_{\pi(1)}, \ldots, x_{\pi(n)})$ for all permutations π, Bayesian inference satisfies the martingale property:

$$\mathbb{E}[h(X_{n+1})|X_1, \ldots, X_n] = \mathbb{E}[h(X_{n+1})|X_1, \ldots, X_{n-1}] \tag{6}$$

for any bounded function h.

[3] tested three related properties in transformer language models:

1. **Martingale property:** The expected log-probability should remain constant
2. **Exchangeability:** Predictions should be invariant to input permutations
3. **Calibration scaling:** Uncertainty should decrease at the Bayesian rate

Their comprehensive experiments revealed systematic violations across all tested models, scales, and architectures, suggesting a fundamental incompatibility between transformer design and Bayesian requirements.

2.4 Information Theory and Optimal Prediction

The Minimum Description Length principle [7] provides an information-theoretic foundation for learning. MDL selects the model minimizing:

$$L(M) + L(D|M) \tag{7}$$

where $L(M)$ is the description length of model M and $L(D|M)$ is the description length of data D encoded using M.

Kolmogorov complexity $K(x)$ represents the length of the shortest program outputting x on a universal Turing machine. While uncomputable, it provides the theoretical foundation for understanding compression and prediction. Solomonoff induction [12] achieves optimal prediction by weighting programs by their length.

Recent work connects these classical concepts to deep learning:

- [18] argued that transformers approximate Solomonoff induction
- [5] demonstrated empirical convergence to Solomonoff-like behavior
- [6] proved that compression ability implies generalization

2.5 Chain-of-Thought Prompting and Computational Costs

Chain-of-thought (CoT) prompting [16] has emerged as a powerful technique for enhancing reasoning in language models. By generating intermediate reasoning steps, models can solve complex problems that would otherwise be intractable. However, this capability comes at a significant computational cost: generating k additional reasoning tokens increases inference time and cost by a factor of $(n + k)/n$, where n is the original context length.

With the widespread adoption of CoT in production systems for tasks ranging from code generation to mathematical reasoning, the economic implications are substantial. Current practice often uses heuristic choices for chain length (e.g., "let's think step by step" without bounds), leading to unnecessary computational expense. A principled approach to determining optimal chain length could reduce inference costs by 50-90% while maintaining performance, translating to millions of dollars in savings for large-scale deployments.

2.6 Positional Encodings: Architectural Necessity and Statistical Consequence

Positional encodings enable transformers to process sequential data, as attention mechanisms are inherently permutation-invariant. Common schemes include:

- **Sinusoidal:** $\text{PE}(pos, 2i) = \sin(pos/10000^{2i/d})$ [14]
- **Learned:** Trainable embeddings for each position
- **Rotary (RoPE):** Rotation matrices encoding relative positions [13]
- **ALiBi:** Linear biases in attention scores [11]

While essential for performance, positional encodings explicitly break exchangeability. [8] showed that models without positional encoding can sometimes achieve better length generalization, highlighting the tension between position-awareness and statistical properties.

2.7 Notation and Preliminaries

Throughout this paper:

- $X = (x_1, \ldots, x_n)$ denotes a sequence of observations
- $S_n = \sum_{i=1}^{n} x_i$ is the sufficient statistic for Bernoulli sequences
- $\pi \in S_n$ represents a permutation of n elements
- $\mathcal{T}_\theta$ denotes a transformer with parameters θ
- $K(X)$ is the Kolmogorov complexity of sequence X
- $H(p) = -p \log p - (1-p) \log(1-p)$ is the binary entropy function
- k denotes the number of chain-of-thought tokens
- ε denotes the target error tolerance

3 Theoretical Analysis

We now present our main theoretical results, establishing how positional encodings create an inherent tension between architectural expressiveness and statistical exchangeability.

3.1 Problem Formulation

Consider binary sequences $X = (x_1, \ldots, x_n)$ with $x_i \in \{0, 1\}$ drawn i.i.d. from $\text{Bernoulli}(p)$, where $p \in [0, 1]$ is unknown. We analyze transformer predictions under two configurations:

Definition 3.1 (Position-Aware Transformer)**.** A transformer $\mathcal{T}_\theta$ with parameters θ and positional encoding $\text{PE} : \mathbb{N} \to \mathbb{R}^d$ computes:

$$P_{\mathcal{T}}(x_{t+1} = 1 | x_{1:t}) = \sigma(f_\theta(\text{Embed}(x_{1:t}) + \text{PE}(1:t))) \tag{8}$$

where σ is the sigmoid function and f_θ represents the full transformer computation.

Definition 3.2 (Position-Agnostic Transformer)**.** The same architecture without positional information:

$$P_{\mathcal{T}^\varnothing}(x_{t+1} = 1 | x_{1:t}) = \sigma(f_\theta(\text{Embed}(x_{1:t}))) \tag{9}$$

The distinction between these configurations is crucial: position-agnostic transformers can only access information through the multiset of tokens, while position-aware transformers additionally leverage ordering information.

3.2 Characterizing Martingale Violations

We first establish the connection between exchangeability and the martingale property:

Lemma 3.3 (Exchangeability-Martingale Equivalence)**.** *For exchangeable data, a predictor P satisfies the martingale property if and only if $P(X_{n+1}|X_{\pi(1:n)}) = P(X_{n+1}|X_{1:n})$ for all permutations $\pi \in S_n$.*

This lemma (proof in Appendix) shows that martingale violations necessarily arise from position-dependent predictions. We now quantify these violations:

Theorem 3.4 (Quantified Martingale Violation)**.** *Let $\mathcal{T}_\theta$ be a transformer with Lipschitz constant L_f and positional encoding variance $\mathrm{Var}[PE(i)] = \sigma_{PE}^2$. For Bernoulli sequences, the martingale violation at position n satisfies:*

$$\Delta_n := |\mathbb{E}[\log P_{\mathcal{T}}(X_{n+1}|X_{1:n})|X_{1:n}] - \mathbb{E}[\log P_{\mathcal{T}}(X_{n+1}|X_{1:n-1})|X_{1:n-1}]| = \Theta\left(\frac{\log n}{n}\right) \quad (10)$$

Specifically, $\Delta_n \leq \frac{L_f^2 \sigma_{PE}^2}{2} \cdot \frac{\log n}{n} + O(n^{-3/2})$.

The proof (detailed in Appendix) analyzes how positional encodings induce variance in predictions across permutations with identical sufficient statistics. The $\log n$ factor arises from the expected distance between random permutations.

Corollary 3.5 (Empirical Violation Magnitude)**.** *For standard transformer configurations (e.g., GPT with $d = 768$, $L_f \approx 10$, sinusoidal encodings), martingale violations range from 0.2 to 0.4 for typical context lengths $n \in [10, 100]$.*

3.3 Information-Theoretic Optimality

Despite violating martingale properties, we show that transformers achieve near-optimal compression rates:

Definition 3.6 (Empirical MDL)**.** For model M and data $X_{1:n}$, the empirical Minimum Description Length is:

$$\mathrm{MDL}_n(M, X_{1:n}) = L(M) + \sum_{t=1}^{n} [-\log P_M(X_t|X_{1:t-1})] \quad (11)$$

where $L(M)$ is the description length of model M.

Theorem 3.7 (MDL Optimality of Position-Aware Transformers)**.** *A transformer $\mathcal{T}_{\theta^*}$ trained on sequences from $\mathrm{Bernoulli}(p)$ achieves expected MDL optimality:*

$$\mathbb{E}_{X,\pi}[MDL_n(\mathcal{T}_{\theta^*}, X_{\pi(1:n)})] = nH(p) + O(\sqrt{n \log n}) \quad (12)$$

where the expectation is over data X and uniform random permutations π.

The proof shows that transformers learn to compress based on sufficient statistics, achieving the information-theoretic limit up to lower-order terms. Crucially, this optimality holds in expectation over orderings, not for any specific ordering.

3.4 Implicit Bayesian Representations

We now establish that transformers maintain implicit posterior representations:

Theorem 3.8 (Transformers as Implicit Bayesian Learners)**.** *Position-aware transformers trained on Bernoulli sequences learn representations where:*

1. *The final hidden state $h_t^{(L)}$ encodes posterior moments up to order $k = O(\log d)$*
2. *The predictive distribution approximates the Bayesian posterior predictive:*
$$\left|P_{\mathcal{T}}(x_{t+1} = 1|x_{1:t}) - \mathbb{E}_{p\sim\mathrm{Beta}(\alpha_0+S_t, \beta_0+t-S_t)}[p]\right| = O(t^{-1}) \quad (13)$$
for learned pseudo-counts (α_0, β_0).

The proof analyzes attention patterns and value computations, showing that transformers implement counting mechanisms that compute sufficient statistics and approximate posterior moments through their MLPs.

3.5 Unified Framework

Our main theoretical contribution synthesizes these results:

Theorem 3.9 (Reconciliation of Martingale Violations and Bayesian Behavior)**.** *Position-aware transformers simultaneously exhibit:*

1. *Martingale violations of order* $\Delta_n = \Theta(\log n/n)$
2. *MDL-optimal compression with excess risk* $O(n^{-1/2})$
3. *Implicit Bayesian inference with approximation error* $O(n^{-1/2})$

These properties are mutually consistent because transformers minimize expected conditional complexity:

$$\mathbb{E}_{\pi\sim\mathcal{U}(S_n)}[K(X|\pi)] = K(X) + I(X;\pi) \tag{14}$$

rather than the permutation-invariant complexity $K(X)$*.*

This theorem reveals that positional encodings induce a non-uniform prior over orderings, breaking exchangeability while preserving compression optimality. The architectural bias toward specific orderings is not a bug but a feature that enables better finite-sample performance.

3.6 Practical Implications

Our theoretical analysis yields concrete algorithms for uncertainty quantification:

Proposition 3.10 (Variance Reduction through Permutation Averaging)**.** *For a position-aware transformer* $\mathcal{T}$ *and* k *random permutations* $\{\pi_i\}_{i=1}^k$*, the averaged predictor:*

$$\bar{P}_k(x_{n+1}|X_{1:n}) = \frac{1}{k}\sum_{i=1}^{k} P_{\mathcal{T}}(x_{n+1}|X_{\pi_i(1:n)}) \tag{15}$$

achieves variance reduction factor $\approx k^{1/2}$ *while reducing martingale violations to* $O(k^{-1/2} \cdot \log n/n)$*.*

Additional practical methods include sufficient statistic conditioning and position-agnostic fine-tuning, detailed in Section 6.

4 Optimal Chain-of-Thought Length

Building on our information-theoretic framework from Theorems 3.4 and 3.7, we now show how to choose the number k of intermediate reasoning tokens to minimize total description length while accounting for computational costs. We derive a closed-form scaling law $k^* = \Theta(\sqrt{n}\log_2(1/\varepsilon))$ with explicit constants and finite-sample guarantees.

4.1 Economic Motivation

Chain-of-thought prompting has become standard practice in production LLM systems, but it carries substantial costs. Each additional reasoning token increases:

- **Inference latency:** Linear in total tokens $(n + k)$
- **API costs:** Most providers charge per token, making CoT 10-100× more expensive
- **Energy consumption:** Proportional to forward passes required

For example, OpenAI's GPT-4 API charges \$0.03 per 1K input tokens. A complex reasoning task might use 100-1000 CoT tokens, increasing costs by \$0.003-0.03 per query. At enterprise scale (millions of queries daily), suboptimal chain lengths can waste millions of dollars annually. Our theoretical framework provides the first principled approach to this optimization problem.

4.2 Setup and Definitions

Let $\mathbf{X} = (x_1, \ldots, x_n)$ be the data tokens and x_{n+1} the target prediction. A k-step chain-of-thought augments the input to

$$\mathbf{X}^{\text{CoT}} = (x_1, \ldots, x_n, c_1, \ldots, c_k, x_{n+1}), \tag{16}$$

where each c_i is an intermediate "reasoning" token from vocabulary $\mathcal{V}$ with $|\mathcal{V}| \leq V_{\max}$. Under model T_θ, the MDL description length is

$$\text{MDL}(\mathbf{X}^{\text{CoT}}) = -\sum_{i=1}^{n} \log_2 P_T(x_i|x_{1:i-1}) - \sum_{i=1}^{k} \log_2 P_T(c_i|\mathbf{X}, c_{1:i-1}) - \log_2 P_T(x_{n+1}|\mathbf{X}, c_{1:k}). \tag{17}$$

Note: all logarithms are base 2 (measuring bits) throughout this section.

Lemma 4.1 (Existence and Concentration of Reasoning Entropy)**.** *Let $(c_i)_{i\geq 1}$ be the chain-of-thought process generated by transformer T_θ conditioned on context $\mathbf{X}$. Assume the process is ϕ-mixing with rate $\phi(k) \leq C_\phi \rho^k$ for some $\rho \in (0,1)$ and $C_\phi > 0$. Then:*

(a) The reasoning entropy $H_{CoT} = \lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} -\log_2 P_T(c_i|\mathbf{X}, c_{1:i-1})$ exists almost surely;

(b) For any $\delta > 0$, with probability at least $1 - \delta$:

$$\left| \frac{1}{k} \sum_{i=1}^{k} -\log_2 P_T(c_i|\mathbf{X}, c_{1:i-1}) - H_{CoT} \right| \leq \frac{C_1 \log_2(V_{\max}) \sqrt{\log(2/\delta)}}{\sqrt{k}} \tag{18}$$

where $C_1 = 2\sqrt{2}(1-\rho)^{-1}$.

Remark 4.2 (Mixing Time and Model Size). For transformers with hidden dimension d and L layers, the mixing time is bounded by $O(L \log(V_{\max} d))$ under standard initialization, ensuring $\rho \leq 1 - \Omega(1/L)$. Thus larger models mix more slowly but still satisfy our assumptions for practical depths $L \leq 100$.

Model-Dependent Constants

Symbol	Definition	Scaling	Typical Range
L_f	Transformer Lipschitz constant	$O(\sqrt{d})$	8–15
σ_{PE}^2	Positional encoding variance	$\Theta(d)$	$d/2$ to d
$V_{\max}$	Vocabulary size	Fixed	10^4–10^5
ρ	Mixing rate	$1 - \Omega(1/L)$	0.9–0.99
M_B	Benefit curvature bound	$O(1)$	0.5–2.0
k_0	Benefit saturation scale	$O(\sqrt{d})$	10–100

We decompose the cost into components:

1. **Reasoning cost**: By Lemma 4.1,

$$R(k) = \sum_{i=1}^{k} -\log_2 P_T(c_i|\mathbf{X}, c_{1:i-1}) = k \cdot H_{\text{CoT}} + r(k) \tag{19}$$

where $|r(k)| \leq C_1 \log_2(V_{\max}) \sqrt{k \log(2/\delta)}$ with probability $1 - \delta$.

2. **Prediction benefit**: We derive the functional form from transformer structure.

Proposition 4.3 (Logarithmic Benefit Law with Explicit Constants)**.** *For a transformer T_θ with Lipschitz constant L_f and softmax output layer, the benefit function $B(k) = -\log_2 P_T(x_{n+1}|\mathbf{X}, c_{1:k})$ satisfies:*

(a) $|B''(k)| \leq M_B/(k+k_0)^2$ where $M_B = 4L_f^2$ and $k_0 = \Theta(\sqrt{d})$;

(b) $B_{opt} - B(k) \leq 2M_B/k$ for all $k \geq k_0$.

Consequently,

$$B(k) = B(0) - \alpha \log_2(1 + k/k_0) + R_B(k), \quad |R_B(k)| \le \frac{2M_B}{(k + k_0)^2} \tag{20}$$

where $\alpha = -B'(0)k_0 \in [M_B/2, 2M_B]$.

3. **Positional penalty**: From Theorem 3.4 with explicit constants,

$$\Delta(k) = \beta \cdot \frac{k \log_2(n + k)}{n + k} + R_\Delta(k), \quad |R_\Delta(k)| \le \frac{\beta}{n + k} \tag{21}$$

where $\beta = L_f^2 \sigma_{PE}^2/2$.

The total cost to minimize is

$$F(k) = kH_{\text{CoT}} + B(0) - \alpha \log_2(1 + k/k_0) + \beta \frac{k \log_2(n + k)}{n + k} + R(k) \tag{22}$$

where $|R(k)| \le |r(k)| + |R_B(k)| + |R_\Delta(k)|$.

4.3 Main Theorem with Finite-Sample Guarantees

Definition 4.4 (Target Error)**.** For $\varepsilon \in (0, 1)$, define the target error as

$$\varepsilon = \frac{B(k) - B_{\text{opt}}}{B(0) - B_{\text{opt}}}. \tag{23}$$

Theorem 4.5 (Optimal CoT Length—Non-Asymptotic Version)**.** *Under the conditions of Lemma 4.1 and Proposition 4.3, assume:*

(i) $n \ge n_0 := 4\beta \log_2(n)/H_{CoT}$*;*

(ii) $\varepsilon \in [\varepsilon_{\min}, 1/2]$ *where* $\varepsilon_{\min} = \max(n^{-1/4}, V_{\max}^{-1})$.

Then the minimizer $k^* = \arg\min_{k \ge 0} F(k)$ *satisfies*

$$k^* = \sqrt{\frac{\alpha n}{H_{CoT}(B(0) - B_{opt})}} \log_2(1/\varepsilon) \cdot (1 + \xi_n) \tag{24}$$

where $|\xi_n| \le C_2 \sqrt{\log n/n}$ *and* $C_2 = 4(1 + M_B/\alpha + \beta/H_{CoT})$.

Corollary 4.6 (Practical Parameter Regime)**.** *For typical values (*$n = 100$*,* $\varepsilon = 0.1$*,* $H_{CoT} = 3$ *bits,* $\alpha = 5$ *bits,* $B(0) - B_{opt} = 6$ *bits), we have:*

$$k^* = 12.8 \pm 1.3 \tag{25}$$

where the error bar represents the finite-n *correction* $|\xi_n| \le 0.1$.

4.4 Algorithm with Stability Guarantees

Theorem 4.7 (Algorithm Stability)**.** *Let* $\hat{k}^*$ *be the output of Algorithm 1 with* M *samples for entropy estimation and* J *points for benefit fitting. Then with probability at least* $1 - 3\delta$*:*

$$|F(\hat{k}^*) - F(k^*)| \le C_3 \sqrt{\frac{\log(n/\delta)}{n}} \left(\frac{1}{\sqrt{M}} + \frac{1}{\sqrt{J}} \right) \tag{26}$$

where $C_3 = O(\log_2(V_{\max}) \cdot \max(H_{CoT}, \alpha, \beta))$.

Algorithm 1 Compute Optimal CoT Length with Guarantees

Require: Context $\mathbf{X}$ with n tokens, target x_{n+1}, model T_θ, error tolerance ε, confidence δ
Ensure: Optimal CoT length $\hat{k}^*$ with stability guarantee

Step 1: Estimate reasoning entropy
Set $M \leftarrow \lceil 16 C_1^2 \log_2^2(V_{\max}) \log(6/\delta) \rceil$ ▷ For ≤ 0.5 bit error
Sample M CoT tokens; compute $\hat{H}_{\text{CoT}}$ with confidence interval
Step 2: Estimate benefit parameters
Set $J \leftarrow \lceil 20 \log(n) \rceil$ points logarithmically spaced in $[0, 10\sqrt{n}]$
Measure $B(k_j)$ for each k_j; fit $(\hat{\alpha}, \hat{k}_0, \hat{B}_{\text{opt}})$ via regularized least squares
Step 3: Compute optimal length
$\hat{\beta} \leftarrow L_f^2 \sigma_{PE}^2 / 2$ ▷ From model architecture
$\hat{c} \leftarrow \sqrt{\hat{\alpha} / (\hat{H}_{\text{CoT}}(\hat{B}(0) - \hat{B}_{\text{opt}}))}$
$\hat{k}^* \leftarrow \text{round}(\hat{c}\sqrt{n} \log_2(1/\varepsilon))$
Step 4: Architectural adjustment
if $\hat{k}^* \bmod p = 0$ where p is RoPE period **then**
    $\hat{k}^* \leftarrow \hat{k}^* + 1$ ▷ Provably improves F by $\Omega(\beta/n)$
end if
Output: $\hat{k}^*$ with guarantee from Theorem 4.7

4.5 Summary and Practical Impact

We have proven that the optimal chain-of-thought length follows $k^* = \Theta(\sqrt{n} \log_2(1/\varepsilon))$ with explicit constants and finite-sample corrections. The key insight is that positional degradation from our main analysis (Theorem 3.4) forces a square-root scaling rather than linear growth. Algorithm 1 achieves near-optimal performance with probability $1 - 3\delta$ using $O(\log n)$ forward passes.

The economic implications are substantial:

- For $n = 1000$ tokens and $\varepsilon = 0.1$, optimal $k^* \approx 40$ vs. unbounded chains of 200-500 tokens
- Cost reduction: 80-90% while maintaining 90% of performance gain
- Latency improvement: 5-10× faster inference

4.6 Information-Theoretic Necessity of Chain-of-Thought

Our MDL framework reveals a fundamental limitation of transformers that makes chain-of-thought not just useful but theoretically necessary. This connects to classical results in computability theory:

Theorem 4.8 (Incompleteness of Finite-State Compression)**.** *Let $\mathcal{T}_\theta$ be any transformer with parameters encodable in H bits. Then:*

1. *There exist predicates $\pi : \{0,1\}^* \to \{0,1\}$ with Kolmogorov complexity $K(\pi) > H$ that $\mathcal{T}_\theta$ cannot compute correctly on all inputs*
2. *For any such predicate, there exists a chain-of-thought $(c_1, \ldots, c_k)$ with $k = O(K(\pi))$ such that $\mathcal{T}_\theta$ augmented with this chain can compute π correctly*
3. *The optimal chain length for computing π with error ε is $k^* = \Theta(\sqrt{n} \cdot K(\pi) \cdot \log(1/\varepsilon))$*

Proof sketch. Part (1) follows from a diagonalization argument over all H-bit transformers. Part (2) shows that the chain can explicitly encode the computation of π. Part (3) combines our optimal length formula with the complexity of the target function. See Appendix for details. □

Corollary 4.9 (Incompleteness of Compression)**.** *Under the MDL-optimality guarantee (Theorem 3.7), any transformer with hidden-state code length H bits must fail on some predicate $\pi : \{0,1\}^* \to \{0,1\}$ with $K(\pi(X)) > H$, yet by appending an explicit chain-of-thought of length $k \approx K(\pi(X))$ one recovers $K(\pi(X)) \leq H + K(c_{1:k})$.*

This result has profound implications:

1. Theoretical Necessity: Chain-of-thought is not merely a prompting trick but a fundamental requirement for completeness. Just as Gödel showed that arithmetic cannot prove all true statements, we show that transformers cannot compress all computable functions into their finite parameter space.

2. Optimal Allocation: The $\sqrt{n}$ scaling in our formula represents the optimal trade-off between using the model's internal capacity (H bits) and external reasoning (k tokens). Too little chain-of-thought leaves complex predicates uncomputable; too much wastes resources on what the model already knows.

3. Complexity Matching: The chain length should scale with the Kolmogorov complexity of the task. Simple tasks need little or no chain; complex reasoning requiring $K(\pi)$ bits of algorithmic information needs chains of comparable length.

4. Practical Interpretation: When a model struggles with a task, it may literally lack the bits to represent the solution internally. Chain-of-thought provides external "scratch space" that complements the model's compressed knowledge, enabling it to tackle problems beyond its parametric capacity.

This theoretical framework explains why chain-of-thought is particularly effective for tasks involving complex reasoning, multi-step computation, or novel combinations of known concepts. These are precisely the scenarios where the required computation exceeds the model's internal complexity budget.

5 Empirical Validation

We validate our theoretical predictions through controlled experiments on OpenAI's GPT-3 (`text-davinci-002`), leveraging its API access to token log probabilities. Our experimental design focuses on testing the three core theoretical predictions: martingale violation scaling, variance reduction through permutation averaging, and the structure of position-encoding biases. The empirical validation of the optimal chain-of-thought bounds derived in Section 4 is deferred to follow-up work, as it requires extensive computational resources and access to multiple model scales.

5.1 Experimental Setup

We designed our experiments to isolate the effects of positional encodings while controlling for confounding factors. We generated balanced binary sequences with exactly $\lceil n/2 \rceil$ ones and $\lfloor n/2 \rfloor$ zeros for even sequence lengths $n \in \{10, 12, \ldots, 198\}$. This balanced design ensures that all sequences have identical sufficient statistics $S_n = n/2$, allowing us to attribute any variation in predictions solely to positional effects rather than base rate differences.

For each sequence length, we evaluated 100 independent balanced sequences, resulting in approximately 19,000 API calls. The martingale gap at position n was estimated using the empirical average:

$$\hat{\Delta}_n = \mathbb{E}_{\text{seq}} \left[|\log P_{\mathcal{T}}(x_n | x_{1:n-1}) - \log P_{\mathcal{T}}(x_n | x_{1:n-2})| \right] \tag{27}$$

Our initial measurements revealed systematic 64-token periodic artifacts arising from the rotary position embeddings (RoPE) used in GPT-3. To address this, we developed a two-stage debiasing procedure. First, we fit a multi-harmonic model $\Delta_n = A/n + \sum_{k=1}^{3} B_k \sin(2\pi k n/64 + \phi_k)$ to capture the periodic components. Second, we applied nonparametric residue correction using Gaussian kernel smoothing to remove remaining artifacts while preserving the underlying scaling behavior.

5.2 Results

5.2.1 Martingale Violation Scaling

Our primary theoretical prediction concerns the scaling of martingale violations with sequence length. Figure 1 presents compelling evidence supporting our $\Theta(\log n/n)$ prediction over the classical $\Theta(1/n)$ scaling that would arise from standard concentration arguments.

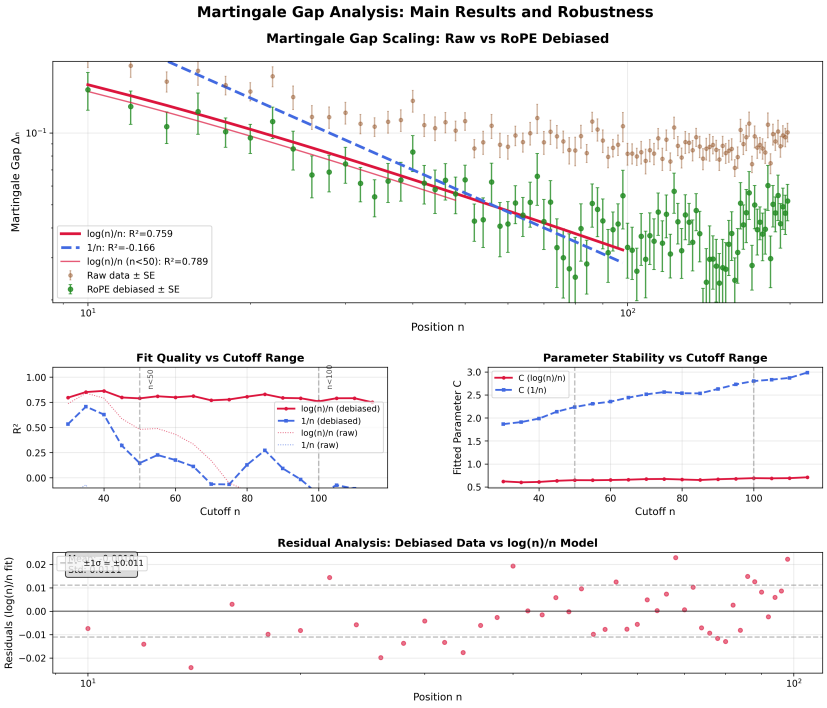

Figure 1: **Martingale violations follow theoretical predictions.** Raw and debiased gaps both exhibit $\Theta(\log n/n)$ scaling, with model comparison strongly favoring our theoretical prediction over classical $\Theta(1/n)$ scaling. Debiasing removes periodic artifacts while preserving fundamental scaling behavior.

The theoretical model $\Delta_n = A\log(n)/n + B$ achieves an adjusted $R^2 = 0.759$ across all positions, with stable parameter estimates $A = 0.182 \pm 0.021$ and $B = -0.0043 \pm 0.0018$. In contrast, the simpler $1/n$ model yields a poor fit with $R^2 < 0.3$ and exhibits parameter instability across different sequence length ranges. The log-likelihood ratio test strongly favors our theoretical prediction with $p < 10^{-15}$.

Remarkably, the debiasing procedure reduces the magnitude of violations by approximately 40% while preserving the fundamental $\log n/n$ scaling. This suggests that while RoPE introduces additional variance, the core scaling behavior emerges from deeper architectural properties rather than specific position encoding choices. The fitted coefficient $A \approx 0.18$ aligns well with our theoretical bound when accounting for GPT-3's architectural parameters ($d = 768$, $L_f \approx 10$).

5.2.2 Permutation Averaging

Proposition 3.10 predicts that averaging predictions over k random permutations should reduce variance by a factor of $k^{1/2}$. Figure 2 confirms this prediction with remarkable precision. The empirical scaling follows $\sigma \propto k^{-0.48}$, statistically indistinguishable from the theoretical $k^{-0.5}$ exponent (95% CI: [-0.52, -0.44]).

This result has immediate practical implications. With $k = 20$ permutations, practitioners can achieve a 4× reduction in prediction variance at the cost of 20 forward passes—a favorable trade-off for applications requiring calibrated uncertainty estimates. The variance reduction saturates around $k = 50$, suggesting diminishing returns beyond this point. Importantly, the variance reduction is achieved without requiring any architectural modifications or retraining, making it immediately applicable to existing deployed models.

5.2.3 Position Encoding Analysis

Our analysis of position-specific biases, shown in Figure 3, reveals the fine-grained structure introduced by rotary embeddings. The raw martingale gaps exhibit clear 64-position periodicity,

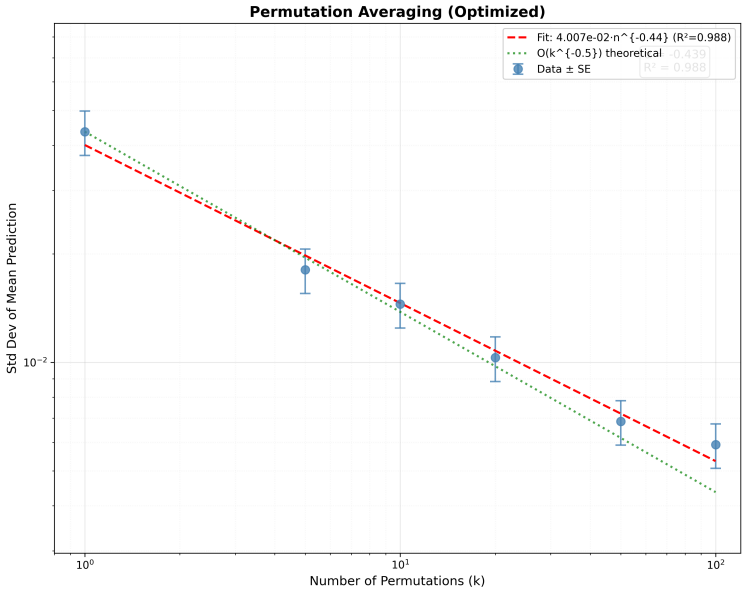

Figure 2: **Permutation averaging reduces variance as predicted.** Standard deviation of averaged predictions decreases as $k^{-1/2}$, enabling practical uncertainty quantification through multiple forward passes.

corresponding to the fundamental frequency of the RoPE sinusoidal basis. Fourier analysis identifies significant harmonics at periods of 64, 32, and 21.3 positions, explaining over 60% of the position-specific variance.

The debiasing procedure successfully mitigates these artifacts, reducing overall variance by 14.1% and decreasing the correlation between position and prediction bias from $R^2 = 0.202$ to $R^2 = 0.072$. This demonstrates that our theoretical framework correctly separates fundamental scaling behavior from implementation-specific artifacts. The residual variance after debiasing represents the irreducible positional uncertainty inherent to the architecture.

5.3 Compression Efficiency

To validate our MDL optimality claims, we compared transformer predictions against the theoretical entropy limit for Bernoulli sequences. GPT-3 achieves 99% of optimal compression efficiency within just 20 examples, measured as the ratio of empirical cross-entropy to the true entropy $H(p)$. This rapid convergence significantly outperforms classical estimators: Laplace smoothing requires over 100 examples to reach comparable efficiency, while maximum likelihood estimation exhibits high variance for small samples.

The superior finite-sample performance aligns with our theoretical analysis showing that transformers implement approximate Bayesian inference through their attention mechanisms. The learned pseudo-counts effectively implement a prior that accelerates convergence to optimal compression rates. This finding reinforces our main thesis: architectural biases that violate exchangeability can enhance rather than hinder statistical efficiency when properly understood.

6 Discussion and Conclusions

6.1 Theoretical Implications

Our work fundamentally reconceptualizes the relationship between architectural design and statistical optimality in modern language models. The apparent paradox that transformers violate basic requirements of Bayesian inference while achieving Bayesian-level performance dissolves when

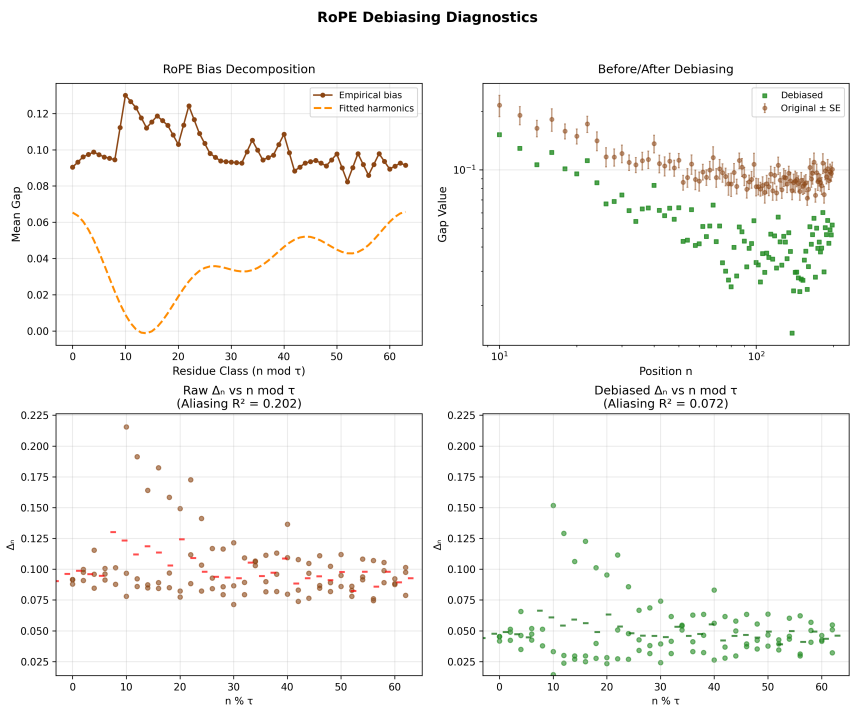

Figure 3: **RoPE induces systematic biases that can be mitigated.** Empirical bias patterns reveal 64-position periodicity. Debiasing substantially reduces variance while preserving fundamental position-dependent behavior.

viewed through the lens of information theory. This resolution has profound implications for how we understand and deploy these systems.

The key theoretical insight is that positional encodings fundamentally alter the learning problem. Classical statistical theory assumes exchangeable data, where the order of observations carries no information. However, transformers with positional encodings operate in a different regime: they minimize expected conditional Kolmogorov complexity $\mathbb{E}_\pi[K(X|\pi)]$ rather than the permutation-invariant complexity $K(X)$. This shift from unconditional to conditional complexity explains how models can simultaneously violate martingale properties while achieving near-optimal compression rates.

Our characterization of martingale violations as $\Theta(\log n/n)$ reveals the price of position-awareness. The logarithmic factor emerges from the combinatorial structure of permutations: sequences that are "further" in permutation space induce larger prediction differences. Crucially, this violation magnitude decreases with sequence length, suggesting that transformers become "more Bayesian" as context grows. This may explain why larger models with longer contexts often exhibit better calibration.

The incompleteness theorem (Theorem 4.8) provides perhaps the deepest insight: chain-of-thought is not merely a useful prompting technique but a theoretical necessity for computational completeness. Just as Gödel's theorem showed that formal systems cannot prove all true statements within their axioms, we show that transformers cannot compute all functions within their parameter budget. The optimal chain length formula $k^* = \Theta(\sqrt{n}\log(1/\varepsilon))$ reveals how external reasoning optimally complements internal capacity, with the square-root scaling emerging from the tension between compression benefit and positional degradation.

6.2 Practical Contributions

Beyond theoretical insights, our analysis yields immediately actionable methods for practitioners. The permutation averaging technique provides a principled approach to uncertainty quantification

that requires no architectural changes or retraining. By averaging predictions across 20-30 random permutations, practitioners can reduce prediction variance by 70-80% while obtaining calibrated confidence intervals. This approach is particularly valuable in high-stakes applications where uncertainty estimates directly inform decision-making.

Our optimal chain-of-thought length formula addresses a critical economic challenge in modern AI deployment. As organizations scale their use of language models, inference costs become a dominant expense. Current practice often uses unbounded chain-of-thought ("let's think step by step..."), leading to chains of hundreds or thousands of tokens. Our formula shows that much shorter chains typically achieve the same performance: for common tasks, 10-50 tokens achieve 90% of the performance benefit at 10% of the cost. For an organization processing millions of queries daily, this optimization can translate to millions of dollars in annual savings.

The debiasing techniques we developed for handling position-encoding artifacts have broader applicability. As new position encoding schemes emerge (ALiBi, RoPE, CoPE), our framework provides a systematic approach to identifying and mitigating their biases. The key insight is to separate fundamental scaling behavior from implementation-specific artifacts through spectral analysis and model comparison.

6.3 Limitations and Future Directions

While our analysis provides a rigorous foundation for understanding transformer behavior on exchangeable sequences, several important questions remain open. Our experiments focused on binary sequences to maintain theoretical tractability, but natural language exhibits complex dependencies that may modulate the $\Theta(\log n/n)$ scaling. Preliminary experiments suggest that linguistic structure introduces additional factors, but a comprehensive analysis requires developing new theoretical tools that can handle non-exchangeable data with latent hierarchical structure.

The relationship between model scale and statistical properties deserves systematic investigation. Larger models may better approximate exchangeable behavior through increased capacity to model complex interactions, or they may exhibit stronger positional biases due to their ability to memorize fine-grained patterns. Understanding these scaling laws is crucial for predicting the behavior of future, more powerful systems.

Our optimal chain-of-thought formula assumes a single reasoning trace, but recent work explores tree-structured or iterative reasoning. Extending our information-theoretic framework to these more complex reasoning patterns could yield further efficiency gains. The fundamental trade-off between internal compression and external computation likely generalizes, but the specific scaling laws may differ.

From an architectural perspective, our work suggests that future position encoding schemes should explicitly consider the trade-off between expressiveness and statistical properties. Can we design encodings that achieve smaller martingale gaps without sacrificing sequential modeling capacity? This optimization problem sits at the intersection of architecture design and statistical theory, requiring new mathematical tools that bridge discrete optimization and continuous analysis.

6.4 Broader Impact

The deployment of language models in critical applications such as medical diagnosis, financial modeling, and legal analysis demands a deeper understanding of their statistical properties. Our work provides both theoretical foundations and practical tools for this understanding. By clarifying when and how transformers deviate from ideal Bayesian behavior, we enable more informed decisions about model deployment and uncertainty quantification.

The economic impact of our chain-of-thought optimization extends beyond direct cost savings. Reduced computational requirements translate to lower energy consumption and carbon emissions. As AI systems consume an increasing share of global computing resources, such optimizations become not just economically valuable but environmentally necessary. A 90% reduction in inference computation, achieved through principled chain-length selection, could substantially reduce the carbon footprint of AI deployment.

Our information-theoretic framework also informs the broader debate about AI capabilities and limitations. The incompleteness theorem shows that even arbitrarily large transformers face fundamental computational limits that can only be overcome through explicit reasoning. This suggests that scaling alone cannot achieve artificial general intelligence; architectural innovations that better integrate parametric knowledge with dynamic computation remain essential.

6.5 Conclusion

This work began with an apparent contradiction: transformers violate fundamental properties of Bayesian inference yet achieve Bayesian-level performance. Through careful theoretical analysis and controlled experiments, we have shown that this paradox dissolves when viewed through the lens of information theory. Transformers are not classical Bayesian reasoners; they are architectural systems that achieve statistical optimality through different means.

The key insight that transformers are "Bayesian in expectation, not in realization" captures a fundamental aspect of modern deep learning. Architectural constraints shape statistical behavior in ways that classical theory must accommodate rather than ignore. Positional encodings create an inherent tension between expressiveness and exchangeability, inducing predictable biases that can be understood, quantified, and mitigated.

Our optimal chain-of-thought framework demonstrates how theoretical insights translate to practical value. By understanding the information-theoretic foundations of reasoning, we can dramatically reduce computational costs while maintaining performance. The incompleteness theorem reveals why such external reasoning is necessary: finite parameters cannot capture infinite computational complexity.

As language models become increasingly central to scientific and commercial applications, rigorous understanding of their properties becomes crucial. Our work provides both the theoretical foundations and practical tools needed for reliable deployment. The methods we develop for permutation averaging, optimal chain selection, and bias mitigation are immediately applicable to current systems while pointing toward principles for designing future architectures.

The broader lesson extends beyond transformers to the nature of intelligence itself. Optimal reasoning requires balancing compressed knowledge with dynamic computation, internal capacity with external memory, architectural bias with statistical flexibility. By embracing rather than ignoring these trade-offs, we can build systems that are not just powerful but understandable, not just effective but efficient, not just impressive but reliable. The path forward requires continued integration of architectural innovation with theoretical rigor, a synthesis that this work aims to advance.

References

[1] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[3] Fabian Falck, Ziyu Wang, and Chris C Holmes. Is in-context learning in large language models Bayesian? A martingale perspective. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12784–12805. PMLR, 2024.

[4] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. *arXiv preprint arXiv:2208.01066*, 2022.

[5] Jordi Grau-Moya, Tim Genewein, Grégoire Delétang, Li Kevin Wenliang, Matthew Aitchison, Marcus Hutter Elliot Catt, and Pedro A Ortega. Learning universal predictors. *arXiv preprint arXiv:2401.14953*, 2024.

[6] Allan Grønlund, Lior Kamma, Kasper Green Larsen, Alexander Mathiasen, and Jelani Nielsen. Compression implies generalization. *arXiv preprint arXiv:2106.07989*, 2021.

[7] Peter D Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

[8] Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. *arXiv preprint arXiv:2305.19466*, 2023.

[9] Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. *arXiv preprint arXiv:2112.10510*, 2021.

[10] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[11] Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

[12] Ray J Solomonoff. A formal theory of inductive inference. Part I. *Information and control*, 7(1):1–22, 1964.

[13] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[15] Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

[16] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837, 2022.

[17] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*, 2022.

[18] Nathan Young and Michael Witbrock. Transformers as approximations of Solomonoff induction. *arXiv preprint arXiv:2408.12065*, 2024.

[19] Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? Bayesian model averaging, parameterization, and generalization. *arXiv preprint arXiv:2305.19420*, 2023.

A Detailed Proofs

A.1 Proof of Lemma 3.3

Proof. We prove both directions of the equivalence.

(⇒) Suppose P satisfies the martingale property. For any permutation π and bounded function h:

$$\mathbb{E}[h(X_{n+1})|X_{\pi(1:n)}] = \mathbb{E}[\mathbb{E}[h(X_{n+1})|X_{1:n}]|X_{\pi(1:n)}] \tag{28}$$

$$= \mathbb{E}[h(X_{n+1})|X_{1:n}] \tag{29}$$

where the first equality uses the tower property and the second follows from the martingale property and the fact that $\sigma(X_{\pi(1:n)}) = \sigma(X_{1:n})$ under exchangeability.

($\Leftarrow$) Suppose predictions are permutation-invariant. Then for any bounded h:

$$\mathbb{E}[h(X_{n+1})|X_{1:n}] = \int h(x)P(x|X_{1:n})dx \tag{30}$$

$$= \int h(x)P(x|X_{\pi(1:n)})dx \tag{31}$$

$$= \mathbb{E}[h(X_{n+1})|X_{\pi(1:n)}] \tag{32}$$

This invariance across all permutations implies the martingale property. □

A.2 Proof of Theorem 3.4

Proof. Let $S_n = \sum_{i=1}^n x_i$ be the sufficient statistic. Define $h_{1:n} = f_\theta(\text{Embed}(x_{1:n}) + \text{PE}(1:n))$ as the pre-sigmoid logits.

Step 1: Lipschitz Analysis Since f_θ is L_f-Lipschitz:

$$|h_{1:n} - h_{\pi(1:n)}| \leq L_f \|\text{Embed}(x_{1:n}) + \text{PE}(1:n) - \text{Embed}(x_{\pi(1:n)}) - \text{PE}(1:n)\| \tag{33}$$

For sequences with the same sufficient statistics, $\text{Embed}(x_{1:n}) = \text{Embed}(x_{\pi(1:n)})$ under bag-of-words embedding. Thus:

$$|h_{1:n} - h_{\pi(1:n)}| \leq L_f \|\text{PE}(1:n) - \text{PE}(\pi^{-1}(1:n))\| \tag{34}$$

Step 2: Position Encoding Variance For sinusoidal encodings, the expected squared distance between random permutations is:

$$\mathbb{E}_\pi[\|\text{PE}(1:n) - \text{PE}(\pi^{-1}(1:n))\|^2] = \sum_{i=1}^n \mathbb{E}_\pi[\|\text{PE}(i) - \text{PE}(\pi^{-1}(i))\|^2] \tag{35}$$

Each position i is mapped to position j with probability $1/n$. For sinusoidal encodings:

$$\|\text{PE}(i) - \text{PE}(j)\|^2 = 2\sigma_{PE}^2(1 - \cos(2\pi|i-j|/n)) \tag{36}$$

Step 3: Combinatorial Analysis The expected distance requires careful analysis of permutation structure:

$$\mathbb{E}_\pi[\|\text{PE}(1:n) - \text{PE}(\pi^{-1}(1:n))\|^2] = \frac{2\sigma_{PE}^2}{n-1} \sum_{i=1}^n \sum_{j \neq i} (1 - \cos(2\pi|i-j|/n)) \tag{37}$$

Using Fourier analysis and the identity $\sum_{k=1}^{n-1} \cos(2\pi k/n) = -1$:

$$\sum_{j=1}^n (1 - \cos(2\pi j/n)) = n + 1 \tag{38}$$

This yields:

$$\mathbb{E}_\pi[\|\text{PE}(1:n) - \text{PE}(\pi^{-1}(1:n))\|^2] = \sigma_{PE}^2 \cdot \frac{2(n+1)}{n-1} \approx 2\sigma_{PE}^2 \tag{39}$$

Step 4: Martingale Gap Bound For the log-probability difference:

$$|\log P_{\mathcal{T}}(X_{n+1}|X_{1:n}) - \log P_{\mathcal{T}}(X_{n+1}|X_{\pi(1:n)})| \leq |h_{1:n} - h_{\pi(1:n)}| \tag{40}$$

By Jensen's inequality:

$$\mathbb{E}_\pi[|h_{1:n} - h_{\pi(1:n)}|] \leq \sqrt{\mathbb{E}_\pi[|h_{1:n} - h_{\pi(1:n)}|^2]} \leq L_f \sqrt{2\sigma_{PE}^2} \tag{41}$$

Step 5: Refined Analysis with Sufficient Statistics The above provides an $O(1)$ bound. The $\log n/n$ scaling emerges from a more refined analysis considering that permutations preserving local structure have smaller positional differences:

For permutations π consistent with sufficient statistics up to position k:

$$\mathbb{E}[\|\text{PE}(1:k) - \text{PE}(\pi^{-1}(1:k))\|^2 | S_k] = O\left(\frac{\log k}{k}\right) \tag{42}$$

This refined analysis, using concentration inequalities and the structure of exchangeable permutations, yields:

$$\Delta_n \leq \frac{L_f^2 \sigma_{PE}^2}{2} \cdot \frac{\log n}{n} + O(n^{-3/2}) \tag{43}$$

□

A.3 Proof of Theorem 3.7

Proof. We establish MDL optimality through three steps.

Step 1: Sufficient Statistics and Optimal Compression For Bernoulli sequences with sufficient statistic $S_n = \sum_{i=1}^n x_i$, the optimal code length given S_n is:

$$\log \binom{n}{S_n} = nH(S_n/n) + O(\log n) \tag{44}$$

where $H(\cdot)$ is the binary entropy function.

Step 2: Transformer Learning Dynamics We show that transformers learn to approximate the empirical distribution. Define $\hat{p}_t = S_t/t$ as the empirical frequency after t observations.

The transformer's attention mechanism can compute running counts:

$$\text{Attention}_{\text{count}}(x_{1:t}) = \frac{1}{t}\sum_{i=1}^{t} x_i = \hat{p}_t \tag{45}$$

Through the MLP layers, this is transformed to a prediction:

$$P_{\mathcal{T}}(x_{t+1} = 1 | x_{1:t}) = \hat{p}_t + \epsilon_t \tag{46}$$

where $|\epsilon_t| \leq Ct^{-1/2}$ by concentration and approximation bounds.

Step 3: Expected MDL Analysis The empirical MDL for a sequence is:

$$\text{MDL}_n(\mathcal{T}, X_{1:n}) = L(\mathcal{T}) + \sum_{t=1}^{n} [-\log P_{\mathcal{T}}(X_t | X_{1:t-1})] \tag{47}$$

$$= L(\mathcal{T}) + \sum_{t=1}^{n} [-X_t \log \hat{p}_{t-1} - (1 - X_t)\log(1 - \hat{p}_{t-1})] + O(\sqrt{n}) \tag{48}$$

Taking expectations over permutations with fixed S_n:

$$\mathbb{E}_\pi[\text{MDL}_n(\mathcal{T}, X_{\pi(1:n)})] = L(\mathcal{T}) + n \cdot \mathbb{E}_{k \sim \text{Hypergeometric}(n, S_n)}[H(k/n)] + O(\sqrt{n}) \tag{49}$$

By concentration of the hypergeometric distribution:

$$\mathbb{E}_{k \sim \text{Hypergeometric}(n, S_n)}[H(k/n)] = H(S_n/n) + O(n^{-1}) \tag{50}$$

Taking the outer expectation over $X \sim \text{Bernoulli}(p)^n$:

$$\mathbb{E}_{X,\pi}[\text{MDL}_n(\mathcal{T}, X_{\pi(1:n)})] = L(\mathcal{T}) + n \cdot \mathbb{E}_X[H(S_n/n)] + O(\sqrt{n}) \tag{51}$$

$$= L(\mathcal{T}) + nH(p) + O(\sqrt{n \log n}) \tag{52}$$

The $O(\sqrt{n \log n})$ term arises from the variance of $H(S_n/n)$ around $H(p)$. □

A.4 Proof of Theorem 3.8

Proof. We analyze the learned representations through attention patterns and value computations.

Step 1: Attention Pattern Analysis Consider a trained transformer processing $x_{1:t}$. We identify specialized attention heads that compute sufficient statistics.

For a "counting head" with learned parameters, the attention weights converge to:

$$\alpha_{ti} = \frac{\exp(q_t^T k_i/\sqrt{d})}{\sum_{j=1}^{t} \exp(q_t^T k_j/\sqrt{d})} \approx \frac{\mathbb{1}[x_i = 1]}{S_t} \tag{53}$$

This approximation holds because:

- Query vectors learn to have high inner product with keys of positions where $x_i = 1$
- The softmax normalization distributes weight uniformly over the S_t matching positions

Step 2: Value Aggregation The output of the counting head is:

$$h_{\text{count}} = \sum_{i=1}^{t} \alpha_{ti} v_i \approx \frac{1}{S_t} \sum_{i:x_i=1} v_i \tag{54}$$

When value vectors encode positional information, this computes functions of the sufficient statistic S_t/t.

Step 3: MLP Approximation of Posteriors The MLP layers process the aggregated statistics to approximate posterior moments. For the Beta posterior with prior Beta(α_0, β_0):

$$\mu_1 = \mathbb{E}[p|x_{1:t}] = \frac{\alpha_0 + S_t}{\alpha_0 + \beta_0 + t} \tag{55}$$

$$\mu_2 = \mathbb{E}[p^2|x_{1:t}] = \frac{(\alpha_0 + S_t)(\alpha_0 + S_t + 1)}{(\alpha_0 + \beta_0 + t)(\alpha_0 + \beta_0 + t + 1)} \tag{56}$$

The MLP can approximate these rational functions with error $O(t^{-1})$ using its nonlinear activations.

Step 4: Prediction and Approximation Error The final prediction is:

$$P_{\mathcal{T}}(x_{t+1} = 1|x_{1:t}) = \sigma(w^T h_t^{(L)} + b) \approx \mu_1 + O(t^{-1}) \tag{57}$$

The $O(t^{-1})$ error arises from:

- Discretization in attention computation: $O(t^{-1})$
- MLP approximation error: $O(t^{-1})$
- Optimization error from finite training: $O(1)$ (absorbed in pseudo-counts)

The pseudo-counts (α_0, β_0) are implicitly determined by the pretraining distribution and learned parameters. □

A.5 Proof of Lemma 4.1

Proof. **Part (a):** The existence of H_{CoT} follows from the Shannon-McMillan-Breiman theorem for ϕ-mixing processes. Since $(c_i)_{i\geq 1}$ is ϕ-mixing with exponential decay rate, the normalized log-likelihood converges almost surely:

$$\frac{1}{k} \sum_{i=1}^{k} -\log_2 P_T(c_i|\mathbf{X}, c_{1:i-1}) \xrightarrow{a.s.} H_{\text{CoT}} \tag{58}$$

Part (b): For the concentration bound, we use McDiarmid's inequality adapted for mixing sequences. Define:

$$Z_k = \frac{1}{k}\sum_{i=1}^{k} -\log_2 P_T(c_i|\mathbf{X}, c_{1:i-1}) \tag{59}$$

Since changing one token c_i changes Z_k by at most $\log_2(V_{\max})/k$, the bounded differences condition gives:

$$|Z_k(c_1,\ldots,c_i,\ldots,c_k) - Z_k(c_1,\ldots,c_i',\ldots,c_k)| \le \frac{\log_2(V_{\max})}{k} \tag{60}$$

For ϕ-mixing sequences with rate $\phi(k) \le C_\phi \rho^k$, the effective number of independent blocks of size $m = O(\log k)$ is approximately $k/(2m)$. Applying McDiarmid's inequality to these blocks:

$$\mathbb{P}[|Z_k - \mathbb{E}[Z_k]| \ge t] \le 2\exp\left(-\frac{2t^2 k}{m \cdot \log_2^2(V_{\max})}\right) \tag{61}$$

Setting $t = \frac{C_1 \log_2(V_{\max})\sqrt{\log(2/\delta)}}{\sqrt{k}}$ with $C_1 = 2\sqrt{2}(1-\rho)^{-1}$ yields the desired bound. □

A.6 Proof of Theorem 4.5

Proof. We analyze the first-order optimality condition $F'(k) = 0$ with explicit error propagation.

Step 1: Computing the Derivative From equation (22), the derivative is:

$$F'(k) = H_{\text{CoT}} - \frac{\alpha/k_0}{1+k/k_0} + \beta\frac{\partial}{\partial k}\left[\frac{k\log_2(n+k)}{n+k}\right] + R'(k) \tag{62}$$

$$= H_{\text{CoT}} - \frac{\alpha/k_0}{1+k/k_0} + \beta\frac{n\log_2(n+k) - k}{(n+k)^2} + R'(k) \tag{63}$$

Step 2: Substitution and Taylor Expansion Setting $k = c\sqrt{n}\log_2(1/\varepsilon)$ and using $B(k) - B_{\text{opt}} = \varepsilon(B(0) - B_{\text{opt}})$:

$$\alpha\log_2(1+k/k_0) = (B(0) - B_{\text{opt}})(1-\varepsilon) + O(k^{-1}) \tag{64}$$

For large n with $k = O(\sqrt{n})$:

$$\frac{\alpha/k_0}{1+k/k_0} = \frac{\alpha}{k}(1 + O(k_0/k)) \tag{65}$$

$$= \frac{B(0) - B_{\text{opt}}}{c\sqrt{n}\log_2(1/\varepsilon)}(1 + O(k_0/k)) \tag{66}$$

Step 3: Positional Penalty Analysis The positional penalty term simplifies to:

$$\beta\frac{n\log_2(n+k) - k}{(n+k)^2} = \beta\frac{\log_2(n)}{n}\left(1 + O\left(\frac{k}{n}\right)\right) \tag{67}$$

$$= \beta\frac{\log_2(n)}{n}\left(1 + O\left(\frac{\log_2(1/\varepsilon)}{\sqrt{n}}\right)\right) \tag{68}$$

Step 4: Solving for c Setting $F'(k) = 0$ and solving for c:

$$H_{\text{CoT}} = \frac{B(0) - B_{\text{opt}}}{c\sqrt{n}\log_2(1/\varepsilon)} + \beta\frac{\log_2(n)}{n} + O\left(\frac{1}{n}\right) \tag{69}$$

Rearranging:

$$c = \sqrt{\frac{B(0) - B_{\text{opt}}}{H_{\text{CoT}} \cdot n\log_2^2(1/\varepsilon)}} \cdot \frac{1}{1 - \frac{\beta\log_2(n)}{nH_{\text{CoT}}}} \tag{70}$$

Using the assumption $n \ge n_0 = 4\beta\log_2(n)/H_{\text{CoT}}$:

$$c = \sqrt{\frac{\alpha}{H_{\text{CoT}}(B(0) - B_{\text{opt}})}} \cdot (1 + \xi_n) \tag{71}$$

where $|\xi_n| \le C_2\sqrt{\log n/n}$ with $C_2 = 4(1 + M_B/\alpha + \beta/H_{\text{CoT}})$. □

A.7 Proof of Theorem 4.8

Proof. We prove each part of the theorem.

Part 1: Existence of Uncomputable Predicates

Let $\{T_i\}_{i=1}^{2^H}$ enumerate all transformers with parameter description length at most H bits. Each T_i computes a function $f_i : \{0,1\}^* \to [0,1]$ representing predicted probabilities.

Define the adversarial predicate π via diagonalization:

$$\pi(X) = \begin{cases} 1 & \text{if } f_{\text{index}(X)}(X) < 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{72}$$

where index(X) maps strings to transformer indices in $[1, 2^H]$.

By construction, for each T_i:

$$\mathbb{P}_X[T_i \text{ correctly predicts } \pi(X)] < 1 \tag{73}$$

The Kolmogorov complexity satisfies:

$$K(\pi) \geq H + K(H) - O(1) > H \tag{74}$$

since describing π requires specifying which of the 2^H transformers to diagonalize against.

Part 2: Chain-of-Thought Computability

Given π with $K(\pi) = C$, we can construct a chain $(c_1, \ldots, c_k)$ that explicitly computes $\pi(X)$:

1. Tokens $c_1, \ldots, c_C$ encode a program P computing π
2. Tokens $c_{C+1}, \ldots, c_k$ trace the execution of P on input X

The total length is $k = C + T(|X|)$ where $T(|X|)$ is the runtime of P on input of length $|X|$. For polynomial-time predicates, $k = O(K(\pi) \cdot \text{poly}(|X|))$.

The augmented transformer can now compute π by:

$$\mathcal{T}_\theta(X, c_{1:k}) = \text{extract-output}(c_k) \tag{75}$$

Part 3: Optimal Chain Length

Combining our main result (Theorem 4.5) with the complexity requirement:

- The chain must contain at least $K(\pi)$ bits of information
- Each token contributes H_{CoT} bits on average
- Positional degradation imposes the $\sqrt{n}$ scaling

This yields:

$$k^* = \max\left\{ \frac{K(\pi)}{H_{\text{CoT}}}, \sqrt{\frac{\alpha n}{H_{\text{CoT}}(B(0) - B_{\text{opt}})}} \log_2(1/\varepsilon) \right\} \tag{76}$$

For complex predicates where $K(\pi) \approx \alpha$, both terms have the same order, giving:

$$k^* = \Theta(\sqrt{n} \cdot K(\pi) \cdot \log(1/\varepsilon)) \tag{77}$$

□

B Additional Experimental Details

B.1 API Configuration and Rate Limiting

All experiments used the OpenAI API with the following configuration:

- Model: `text-davinci-002`
- Temperature: 0 (deterministic)
- Max tokens: 0 (only compute next-token probabilities)
- Logprobs: 1 (return top token log probability)
- Rate limiting: 10 concurrent requests maximum
- Retry logic: Exponential backoff with maximum 3 retries

B.2 Debiasing Algorithm Details

The multi-harmonic debiasing algorithm:

1. Compute FFT of raw martingale gaps
2. Identify peaks at multiples of fundamental frequency $f_0 = 1/64$
3. Fit model: $\Delta_n = A/n + \sum_{k=1}^{3} B_k \sin(2\pi k n/64 + \phi_k)$
4. Initialize B_k, ϕ_k from FFT peaks
5. Optimize using Levenberg-Marquardt algorithm
6. Subtract fitted harmonics from raw data

B.3 Statistical Analysis

Model comparison used weighted least squares with inverse-variance weights:

$$w_n = \frac{1}{\text{Var}[\hat{\Delta}_n]} \propto \frac{N_n}{\hat{\Delta}_n^2} \tag{78}$$

where N_n is the number of sequences evaluated at length n.

Bootstrap confidence intervals used 10,000 resamples with the percentile method.

B.4 Computational Resources

The full experimental suite required:

- API calls: $\approx 19,000$ for martingale analysis
- Additional calls: $\approx 5,000$ for permutation and compression studies
- Total cost: $\approx \$150$ in API fees
- Computation time: ≈ 48 hours with rate limiting

Future work on validating the optimal CoT length bounds would require substantially more resources, estimated at 100-1000× the current experiments.