
L²M: Mutual Information Scaling Law for Long-Context Language Modeling

Zhuo Chen^{12*} Oriol Mayné i Comas²³ Zhuotao Jin²⁴
Di Luo¹²⁴⁵ Marin Soljačić¹²

¹ NSF AI Institute for Artificial Intelligence and Fundamental Interactions

² Massachusetts Institute of Technology

³ Polytechnic University of Catalonia

⁴ Harvard University

⁵ University of California, Los Angeles

{chenzhuo, omayne, jinzhta, diluo, soljadic}@mit.edu

Abstract

We present a universal[†] theoretical framework for understanding *long-context language modeling* based on a *bipartite* mutual information scaling law that we rigorously verify in natural language. We demonstrate that bipartite mutual information captures multi-token interactions distinct from and scaling independently of conventional two-point mutual information, and show that this provides a more complete characterization of the dependencies needed for accurately modeling long sequences. Leveraging this scaling law, we formulate the **Long-context Language Modeling (L²M)** condition, which lower bounds the necessary scaling of a model’s history state—the latent variables responsible for storing past information—for effective long-context modeling. We validate the framework and its predictions on transformer and state-space models. Our work provides a principled foundation to understand long-context modeling and to design more efficient architectures with stronger long-context capabilities, with potential applications beyond natural language.

1 Introduction

Large language models (LLMs) have revolutionized natural language processing, achieving remarkable capabilities across a wide range of tasks [1–4]. Recent advances in large language models, including ChatGPT [1, 5], Claude, Gemini [6, 7], Grok, LLaMA [4, 8], DeepSeek [9, 10], and Qwen [11, 12] have achieved breakthroughs across diverse tasks, including code generation, mathematical problem solving, text summarization, and creative writing [13–16]. These models have become increasingly powerful and versatile, pushing the boundaries of what’s possible in natural language processing and marking significant steps toward artificial general intelligence [17–19].

In pushing these advances further, the ability to handle long contexts has become increasingly crucial. This ability is the key to document-level understanding, multi-turn dialogue, and complex reasoning. Models like GPT-o1/o3, Claude Opus, Gemini 2.5 pro, and DeepSeek-R1 often generate extensive chains of thought, spanning tens of thousands of tokens to solve complex problems [20, 21]. However, processing long contexts remains a significant challenge. Despite their success and expressiveness, transformer architectures suffer from an intrinsic quadratic computational cost in sequence length,

*Corresponding author

[†]Our framework applies to autoregressive language models, which encompass all widely-used LLMs such as GPT, Claude, Gemini, and LLaMA.

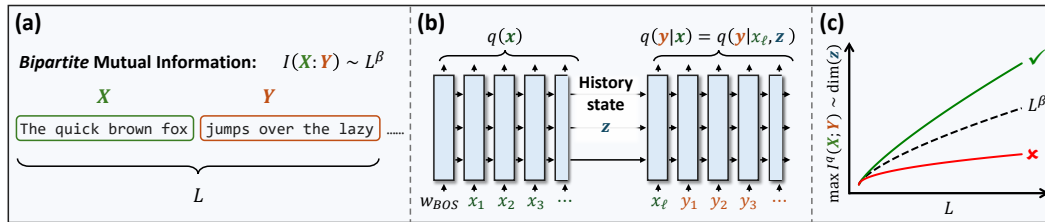


Figure 1: (a) The bipartite mutual information between two text segments scales as a power law (sub-volume law) with sequence length L . (b) In autoregressive models, conditional distributions are parameterized through the history state z , the latent variables that store past information. Examples of the history state include the recurrent states in state-space models or recurrent neural networks, and the key-value pairs in transformers. (c) The maximum bipartite mutual information a model can express scales with the dimensionality of its history state, $\dim(z)$. To model long contexts effectively, $\dim(z)$ must grow at least as fast as the power-law scaling of the true bipartite mutual information.

creating challenges for long sequence generation. Recent advances like DeepSeek have improved per-token efficiency [9], yet the fundamental quadratic cost persists.

Although various architectures have been proposed to address the quadratic scaling [22–31], these approaches still struggle with truly long sequences in practice. A fundamental gap persists in our theoretical understanding of what is necessary for capturing multi-token long-range dependencies in natural language. Despite efforts to characterize these dependencies through various statistical measures [32–35] a theory that can guide practical architecture design remains lacking.

In this work, we address the challenges of understanding long-context language modeling through the following contributions (Fig. 1).

1. We present a universal theoretical framework for autoregressive long-context language modeling based on bipartite mutual information.
2. We demonstrate a bipartite mutual information scaling law in natural language and provide reliable empirical validations of power-law scaling across diverse natural language datasets using state-of-the-art LLMs.
3. We derive the L^2M condition from this scaling law, lower bounding the necessary scaling of a model’s history state dimension for effective long-context modeling.
4. We validate our framework and its predictions across transformer and state-space model (SSM) architectures on both synthetic and natural language datasets of varying lengths.

Our theoretical framework offers crucial insights into understanding an LLM’s capability to model long sequences based on its architectural design. By identifying the minimum required growth rate of the history state, our work provides concrete guidance for designing efficient architectures that can effectively handle long contexts, avoiding the quadratic cost of transformers or the capacity limitations of fixed-state models, paving the way for future AI systems.

2 Related Works

Mutual Information Estimation and Application in Machine Learning

Mutual information estimation and optimization have been extensively studied in machine learning, with approaches including variational bounds [36], neural estimators [37], nearest-neighbor methods [38, 39], and various upper bounds [40]. It has found wide application in areas such as feature selection [41], representation learning [42], disentanglement [43], and generative modeling [44].

Statistical Properties of Natural Language

Natural language exhibits characteristic statistical scaling behaviors across different levels of analysis. Zipf’s law [45] describes how word frequencies decay with their rank, following a power-law distribution. Heaps’ law [45] characterizes vocabulary growth, showing that the number of unique words scales sublinearly with text length. Hilberg’s conjecture and its relaxed version posit specific scaling laws for entropy and bipartite mutual information in natural language, respectively [46].

Neural Scaling Laws

Power-law relationships between model performance, architecture, and computational requirements have been first empirically observed in neural networks [47–49], with theoretical understanding still being developed [50, 51], including recent information-theoretic approach [52]. These observations have guided the development of larger models at fixed context lengths, whereas our work examines a distinct but complementary question: what determines whether model architectures can maintain performance as context length increases?

Universal Prediction and Markov Modeling

Recent studies on transformers as universal predictors [53, 54] show that they can, in principle, model arbitrary variable-order Markov processes, establishing their theoretical universality in prediction. Our analysis focuses instead on the information-theoretic scaling that governs how much past information must be stored to reproduce the mutual-information growth observed in natural language.

Architectures for Efficient Long-Context Modeling

Various approaches have been proposed for processing long sequences. Architectural innovations targeting quadratic complexity include sparse attention [55, 26, 25, 56], recurrent mechanisms [57–59], and alternative formulations [22, 60, 23, 27–30, 61–63, 31]. Efficient attention implementations like Flash Attention [64–66], Lightning Attention [67], and Paged Attention [68] have improved per-token computational efficiency despite maintaining the underlying complexity scaling.

Long-Form Reasoning and Context Utilization

Chain-of-thought prompting [20] and scratchpad methods [69] demonstrate the importance of extended context for complex reasoning tasks, emphasizing the urgent need for effective long-range dependency modeling.

Information Theory and Physics-Inspired Approaches

Recent work has demonstrated how information-theoretic principles and physics-inspired approaches can guide machine learning [70–72], leading to novel architectures [73–80], training methods [81–83], and broad applications [84–87, 82, 88].

3 Preliminaries

Mutual Information

Mutual information $I(X; Y)$ quantifies the statistical dependence between random variables X and Y , defined as $I(X; Y) = D_{KL}(p_{XY} || p_X \otimes p_Y)$, where $D_{KL}(\cdot || \cdot)$ is the Kullback–Leibler (KL) divergence, and p_{XY} is the joint distribution of X and Y . For discrete random variables, mutual information permits equivalent formulations as

$$I(X; Y) = H(X) + H(Y) - H(XY) = H(X) - H(X|Y) = H(Y) - H(Y|X), \quad (1)$$

where $H(\cdot)$ is the (Shannon) entropy and $H(\cdot|\cdot)$ is the conditional entropy. This definition naturally extends to collections of random variables: $I(X_{1:m}; Y_{1:n})$, with $X_{i:j}$ denoting the sequence $(X_i, \dots, X_j)$. For notational convenience, we will use boldface notation $\mathbf{X} := X_{i:j}$ when the indices are clear from context. Similarly, we will drop the index of a single variable $X := X_i$ when convenient.

Autoregressive Neural Networks

Modern LLMs predominantly employ autoregressive neural architectures. An autoregressive neural network models a sequence of conditional probability distributions over tokens $\{q(w_i | w_{1:i-1}, w_{\text{BOS}})\}_{i=1}^L$, where w_{BOS} is the beginning-of-sequence token. Throughout this paper, we use q to denote model-generated probability distributions (and sometimes the model itself) and p to denote the true underlying distributions. Upper case letters denote random variables, and lower case letters denote specific values or realizations of these random variables. These conditional distributions jointly model the probability for a sequence of tokens given a prefix as

$$q(w_{\ell:L} | w_{1:\ell-1}, w_{\text{BOS}}) = \prod_{i=\ell}^L q(w_i | w_{1:i-1}, w_{\text{BOS}}). \quad (2)$$

When $\ell = 1$, this reduces to the distribution of unconditional generation $q(w_{1:L}|w_{\text{BOS}})$. During inference, tokens are sampled sequentially from these conditional distributions to generate text or respond to prompts.

For a complete list of notation and conventions used throughout this paper, we refer the reader to Appx. A.

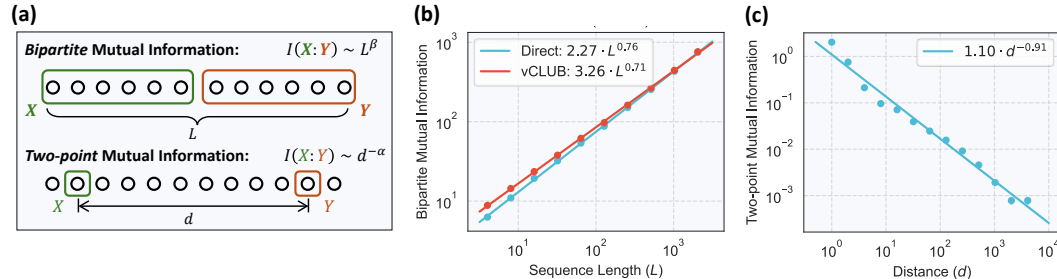


Figure 2: (a) Illustration of bipartite and two-point mutual information. The *bipartite* mutual information measures statistical dependence between two adjacent segments within a text block of length L , whereas the two-point mutual information measures the dependence between two tokens separated by a distance d . (b) Estimates of bipartite mutual information using LLaMA 3.1 405B model [89] on PG19 dataset [90] of pre-1919 books. (c) Estimates of two-point mutual information on PG19 dataset. See Appx. B.I, B.II, and B.VI for additional results.

4 Mutual Information Scaling Laws

4.1 Bipartite Mutual Information as Predictive Information

While classical scaling laws in natural language, such as Zipf's and Heaps' laws, primarily address token-level statistics, a deeper understanding of language modeling necessitates analyzing dependencies between entire text segments. A central challenge in modeling language effectively is to characterize how information is carried over from an existing block of text, X , to inform the generation of a subsequent block, Y . The *bipartite* mutual information between such adjacent blocks directly quantifies this inter-segment information transfer, emerging as a particularly revealing measure.

Definition 4.1 (*Bipartite Mutual Information* [Fig. 2(a)]). For a consecutive sequence of tokens (random variables) $W_{1:L}$ of length L , consider a bipartition of the tokens: $X_{1:\ell} := W_{1:\ell}$ and $Y_{1:L-\ell} := W_{\ell+1:L}$. The bipartite mutual information is the mutual information between the two parts $I_{\ell,L}^{\text{BP}} := I(X_{1:\ell}; Y_{1:L-\ell})$.

The role of bipartite mutual information in quantifying this predictive relationship is formally illuminated by decomposing the entropy of the subsequent block Y :

$$H(Y) = H(Y|X) + I(X; Y) = H(Y|X) + I^{\text{BP}}. \quad (3)$$

This decomposition shows that the total information in Y (its entropy $H(Y)$) consists of two distinct components: new information unique to Y given X (the conditional entropy $H(Y|X)$), and information that Y shares with X (the bipartite mutual information I^{BP}). Consequently, I^{BP} precisely measures the amount of information from the preceding block X that is predictive of the next block Y , and therefore, bipartite mutual information is also referred to as predictive information [91].

Despite its crucial role in quantifying predictive information, this form of mutual information in language has remained relatively underexplored. This research gap is primarily due to two factors: the absence of a comprehensive theory of natural language that would permit a direct calculation, and the substantial challenges in empirically measuring entropy and mutual information for high-dimensional distributions from samples.

Existing literature offers differing perspectives on its scaling properties. On one hand, analogies drawing from critical physical systems [92–97]—often based on two-point mutual information

scaling (discussed later)—suggest that bipartite mutual information should scale logarithmically with sequence length. On the other hand, research in computational linguistics has proposed that it follows power-law growth [98], a behavior often referred to as the sub-volume law (these terms are used interchangeably in this paper). Previous empirical efforts to measure such scaling have been constrained by methodological biases and the curse of dimensionality [46, 98, 99]. Although existing evidence tends to favor sub-volume law growth, these limitations have prevented a definitive characterization. In Sec. 4.3, we address these challenges by leveraging state-of-the-art LLMs as density estimators, establishing clear power-law scaling for bipartite mutual information across diverse datasets.

4.2 Two-point Mutual Information

Before presenting our main results concerning bipartite mutual information scaling, it is instructive to discuss two-point mutual information. This measure has conventionally been used to assess long-range dependencies in natural language, and its scaling properties are relatively well understood.

Definition 4.2 (*Two-point Mutual Information* [Fig. 2(a)]). The two-point mutual information measures the mutual information between two tokens (random variables) X and Y separated by a distance d : $I_d^{\text{TP}} = I(X; Y)$.

Specifically, two-point mutual information has been observed to follow a power-law decay, $I_d^{\text{TP}} \sim d^{-\alpha}$ [92–95]. This characteristic decay has prompted arguments that natural language shares structural properties with critical physical systems, which exhibit similar two-point correlation behavior [96, 97]. However, we contend that such analogies, while offering certain insights, can be misleading when assessing the full complexity of multi-token dependencies crucial for language modeling. The limitations of two-point mutual information in this regard, and why it provides an incomplete characterization for this task, will be detailed in Sec. 4.4 and Appx. B.VIII. Our present discussion of two-point mutual information serves primarily to contrast it with the bipartite measure that is central to our work.

4.3 Empirical Verification of Mutual Information Scaling Laws

Bipartite Mutual Information. Measuring bipartite mutual information presents significant challenges without access to the underlying probability distribution p . Traditional estimation methods face severe limitations in our setting: K-nearest neighbor estimators [39] and neural estimators like MINE [37] and InfoNCE [100] struggle with the high dimensionality of long text sequences, with errors that increase rapidly as sequence length grows. Additionally, neural estimators require substantial training on large amounts of data to learn representations of natural language distributions, especially for long sequences. Fortunately, recent advances in LLMs allow us to circumvent training our own density estimators by offering high-quality approximations q to these distributions (see Appx. B.V for additional discussions). As autoregressive models, LLMs enable efficient computation of conditional probabilities (Sec. 3) and their associated cross-entropies (negative log-likelihoods):

$$H(p_{Y|X}, q_{Y|X}) := -\mathbb{E}_{p_{XY}} \log q(Y|X), \quad (4)$$

where the expectation is taken over samples from the true underlying distribution p_{XY} . The cross-entropy provides an upper bound to the true entropy:

$$H(p_{Y|X}, q_{Y|X}) = D_{KL}(p_{Y|X} || q_{Y|X}) + H^p(Y|X) \geq H^p(Y|X), \quad (5)$$

where the conditional cross-entropy and KL divergence implicitly average over p_X , and H^p (or H^q) denotes the entropy computed with respect to distribution p (or q).

Using these properties, we can construct a direct estimator for bipartite mutual information:

$$I_{\ell;L}^{\text{BP,direct}} = \mathbb{E}_{p_{XY}} [\log q(Y|X) - \log q(Y)] = I^p(X; Y) + \varepsilon(p, q), \quad (6)$$

where $I^p(X; Y)$ denotes mutual information with respect to p , and $\varepsilon(p, q) = D_{KL}(p_{Y|X} || q_{Y|X}) - D_{KL}(p_{Y|X} || q_{Y|X})$. While this estimator no longer provides a bound, it preserves the key property that $\varepsilon(p, q) \rightarrow 0$ as $q \rightarrow p$.

We note that this estimation method faces a specific challenge with modern LLMs: they model $q(w_i | w_{1:i-1}, w_{\text{BOS}})$ rather than $q(w_i | w_{1:i-1})$, where w_{BOS} denotes the BOS token. When sampling

from the dataset, we can ensure $\mathbf{X}$ starts at sentence beginnings, making $q(\mathbf{Y}|\mathbf{X}, w_{BOS}) \equiv q(\mathbf{Y}|\mathbf{X})$. However, $\mathbf{Y}$ may start mid-sentence, creating a mismatch where $q(\mathbf{Y}) \neq q(\mathbf{Y}|w_{BOS})$. This introduces errors in estimating $H(p_{\mathbf{Y}}, q_{\mathbf{Y}})$. We address this using n -gram corrections for the first two tokens, which are the primary source of this bias (see Appx. B.IV).

To circumvent issues with estimating $q(\mathbf{Y})$, we also employ the vCLUB estimator [40]:

$$I_{\ell;L}^{\text{BP,vCLUB}} = \mathbb{E}_{p_{\mathbf{XY}}} \log q(\mathbf{Y}|\mathbf{X}) - \mathbb{E}_{p_{\mathbf{X}} \otimes p_{\mathbf{Y}}} \log q(\mathbf{Y}|\mathbf{X}), \quad (7)$$

where the second term can be calculated by shuffling the second halves of samples in the dataset. Analysis in [40] shows that vCLUB provides an upper bound on the true bipartite mutual information when q closely approximates p . Even when q deviates moderately from p , though the upper bound property may not hold, vCLUB continues to provide reliable estimates of the true bipartite mutual information.

Our empirical analysis in Fig. 2(b) focuses on equal-length partitions of $\mathbf{X}$ and $\mathbf{Y}$ ($\ell = L/2$), where the bipartite mutual information tends to maximize for fixed L 's. Nevertheless, the same analysis can be carried out using other partitions where similar results can be obtained (with the results in Appx. B.II). Using both the bias-corrected direct estimator [Eq.(6)] and vCLUB estimator [Eq. (7)], we measure scaling on the PG19 dataset* [90] (a collection of books before 1919), employing the LLaMA 3.1 405B model [89] as density estimator q . All measurements robustly demonstrate a clear power-law scaling that extends across thousands of tokens. Additional measurements on WIKIPEDIA [101] and using additional LLMs, along with varying ℓ/L ratios, can be found in Appx. B.I and B.II. We note that both estimators likely underestimate the true exponent β (see Appx. B.III for discussions).

Two-point Mutual Information. For completeness, we also measure two-point mutual information scaling on the same datasets, confirming the expected power-law decay [Fig. 2 (c)]. Detailed methodologies for these measurements are provided in Appx. B.VI and B.VII.

4.4 Failures of Two-point Mutual Information

As previously noted, while two-point mutual information is easier to measure and more frequently studied in existing literature, it often fails to adequately capture the long-range multi-token dependencies crucial for natural language modeling. When modeling language, our primary concern is the accurate prediction of future tokens given a preceding context, i.e., $q(w_{\ell:L}|w_{1:\ell-1}, w_{BOS})$. Effective modeling of this conditional distribution necessitates a clear understanding of the multi-token dependencies between the context $w_{1:\ell-1}$ and the subsequent tokens $w_{\ell:L}$. It is important to recognize that this multi-token dependency cannot always be accurately represented by a simple aggregation of pairwise (two-point) interactions; such an approach can be insufficient or even misleading in certain contexts. The following examples illustrate these potential limitations, and we provide more formal derivations in Appx. B.VIII.

Consider a simple distribution where all tokens must be identical: $p(x_1, x_2, \dots, x_L) = \mathbb{1}(x_1 = x_2 = \dots = x_L)/M$, where $\mathbb{1}(\cdot)$ is the indicator function that evaluates to 1 when the condition is satisfied and 0 otherwise, and M is the vocabulary size. This distribution permits a Markov chain construction, as $p(x_1) = 1/M$ and $p(x_i|x_{1:i-1}) = \mathbb{1}(x_i = x_{i-1})$, thus possessing a simple token-to-token dependency structure. Despite this inherent simplicity, the two-point mutual information suggests a misleadingly strong “long-range” dependency: it maintains a large, constant value of $I_d^{\text{TP}} = \log M$ regardless of the distance d , significantly larger than the decaying two-point mutual information typically observed in natural languages. In contrast, bipartite mutual information correctly reflects this simple dependency structure, with $I_{\ell;L}^{\text{BP}} = \log M$ remaining constant for any choice of ℓ and L . This indicates that any two segments share exactly the same amount of information ($\log M$), which is no more than the information shared between just two adjacent tokens, accurately capturing the limited nature of the dependency.

For a more realistic setting, we refer to Appx. C for a discussion of two families of multivariate Gaussian distributions of varying lengths (details of their construction are in Appx C.II). Notably, both families exhibit identical power-law decay in their two-point mutual information when measured between variables at maximum separation. However, their bipartite mutual information scaling differs dramatically: one scales as L^β , akin to natural language, while the other scales as $\log L$, similar to

*We avoid the BOOKS3 dataset due to copyright infringement concerns.

that observed in critical physical systems. This disparity further underscores that two-point mutual information alone may be insufficient to distinguish between systems with fundamentally different long-range correlational structures.

5 Long-Context Language Modeling (L^2M) Condition

Having established bipartite mutual information as a crucial tool for measuring long-range dependencies, we analyze how a model’s capacity to handle long contexts fundamentally depends on its ability to store past information, using bipartite mutual information scaling as our theoretical framework. Intuitively, to model natural language effectively, a model must be able to capture all dependencies between past and future tokens. Since these dependencies (measured by bipartite mutual information) grow with sequence length, the model’s state capacity for storing past information (the history state) must necessarily grow as well. We formalize this intuition through the L^2M condition and explore its implications in detail throughout this section.

5.1 Theoretical Derivations

To analyze how models handle long-range dependencies, we first formalize the notion of *history state*.

Definition 5.1. Consider a sequence of tokens $w_{1:L}$. Denote $x_{1:\ell} := w_{1:\ell}$ and $y_{1:L-\ell} := w_{\ell+1:L}$. Autoregressive neural networks parameterize conditional probabilities by first encoding the input tokens $x_{1:\ell-1}$ into a set of latent intermediate variables $z_\ell = f(x_{1:\ell-1})$ before outputting the conditional probabilities as $q(y_{1:L-\ell}|x_{1:\ell}) := q(y_{1:L-\ell}|x_\ell, z_\ell)$.^{*} We define the *history state* as the smallest set of such latent intermediate variables that fully characterizes the model’s output conditional probability. [Fig. 1(b)].

As illuminating examples, the history state corresponds to the recurrent state in RNNs and SSMs after processing token $w_{\ell-1}$, and to the key-value pairs up to token $w_{\ell-1}$ for transformers (see Appx. D). Generally, the history state z_ℓ is the smallest hidden state responsible for caching all historical information.

The following theorem shows that this history state upper bounds a model’s capacity to capture bipartite mutual information:

Theorem 5.2. *The bipartite mutual information that a model can capture is bounded by the size of its history state:*

$$I_{L/2:L}^{\text{BP},q} \leq C \cdot \dim(z_{L/2}) + \log(M) \quad (8)$$

where C is some constant and M denotes the vocabulary size.

Proof. This theorem admits multiple independent proofs under different mild and practical assumptions. See Appx. E for details. $\square$

We now use this bound to analyze when architectures can maintain performance as sequence length increases. Consider a series of natural language datasets $\{W_{1:L}\}_{L=1}^\infty$ of different lengths, which can be thought of as truncations of an ideal infinite-length dataset.

Definition 5.3. A model q is *MI-capable* if the maximum bipartite mutual information it can express satisfies $\max_{\theta} I_{L/2:L}^{\text{BP},q\theta} \geq I_{L/2:L}^{\text{BP}}$ for any sequence length L , where the maximum is taken over all model parameters θ .

Since a model’s ability to capture mutual information is bounded by its history state dimension, we immediately obtain[†]:

^{*}We separate x_ℓ from z_ℓ to accurately reflect its distinct role as the current input token in autoregressive models, though including it in z_ℓ would not affect the main results of this paper.

[†]See Appx. A for conventions on asymptotic notations.

Theorem 5.4 (L²M Condition for Single Models). *For a model to be MI-capable across all sequence lengths, its history states $\mathbf{z}_{L/2}^q$ must satisfy $\dim(\mathbf{z}_{L/2}^q) \gtrsim I_{L/2;L}^{\text{BP}} \sim L^\beta$.*

Proof. We prove by contrapositive. By Thm. 5.2, if $\dim(\mathbf{z}_{L/2}) \prec I_{L/2;L}^{\text{BP}}$, then $\max_{\theta} I_{L/2;L}^{\text{BP},q\theta} \prec I_{L/2;L}^{\text{BP}}$, implying there exists some L where $\max_{\theta} I_{L/2;L}^{\text{BP},q\theta} < I_{L/2;L}^{\text{BP}}$, violating MI-capability. $\square$

For some architectures, a single fixed-size model may not satisfy this condition across all sequence lengths. In such cases, we can extend our framework to families of models where model size grows with sequence length. Consider a series of models $\{q_L\}_{L=1}^\infty$ of the same architecture, where model size may increase with L .

Definition 5.5. A series of models $\{q_L\}_{L=1}^\infty$ is *MI-capable* if the maximum bipartite mutual information each model can express satisfies $\max_{\theta_L} I_{L/2;L}^{\text{BP},q_L,\theta_L} \geq I_{L/2;L}^{\text{BP}}$ for its corresponding sequence length L , where the maximum is taken over all parameters θ_L of model q_L .

Theorem 5.6 (L²M Condition for Model Series). *For a series of models $\{q_L\}_{L=1}^\infty$ to be MI-capable, the history states $\mathbf{z}_{L/2}^{q_L}$ of each model must satisfy: $\dim(\mathbf{z}_{L/2}^{q_L}) \gtrsim I_{L/2;L}^{\text{BP}} \sim L^\beta$.*

Note that an MI-capable single model trivially induces an MI-capable series when applied to all sequence lengths, though the converse is not true.

5.2 Implications to Common LLM Architectures

We can now apply our framework to analyze whether different architectures satisfy the L²M condition and thus can capture long-range dependencies as sequence length grows.

In transformer-based models (excluding sparse attention and linear attention variants), the history state consists of stored key-value pairs for all previous tokens. Even with fixed model size, these key-value pairs grow linearly with sequence length: $\dim(\mathbf{z}_{L/2}^q) \sim L \gtrsim L^\beta$. This means a single transformer model naturally satisfies the L²M (single model) condition across all sequence lengths, notwithstanding the quadratic computational cost.

In contrast, SSMS, RNNs, and linear attention models, despite being celebrated for their “infinite” context length and linear complexity, cannot satisfy the L²M condition with a single fixed-size model. Their history state dimension remains constant regardless of sequence length, and our theory demonstrates that this constant-size state cannot capture the growing mutual information. However, these architectures can achieve MI-capability (model-series) through a series of models $\{q_L\}_{L=1}^\infty$ where model size, and thus history state dimension, increases with sequence length. This requirement effectively offsets their computational efficiency advantage when modeling long sequences.*

For other architectures, such as sparse attention models and log-linear models, we can similarly analyze their history state scaling to determine whether they satisfy the L²M condition as single models or require a series of growing models. Crucially, any architecture must exhibit power-law growth in its history state dimension with sequence length in order to truly satisfy the single-model L²M condition.

We note that the L²M condition addresses a model’s capacity to capture long-range dependencies, not its overall language modeling capability. It is a necessary but not sufficient condition: architectures that fail to satisfy it will have inherent limitations at longer sequences, while satisfying it does not guarantee effective language modeling. As discussed in Sec. 2, the L²M condition is also distinct from neural scaling laws, which typically study how model performance scales with model size, dataset size, and compute budget at a *fixed* sequence length.

*And a new model must be trained for each sequence length, which can be prohibitively expensive.

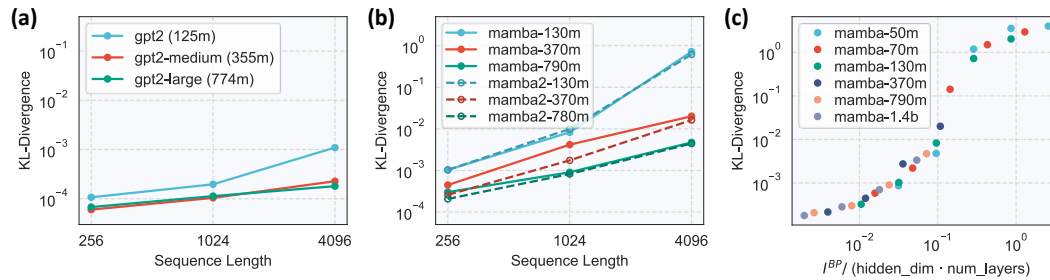


Figure 3: Evaluation of KL-divergence across model architectures trained on synthetic data that satisfies the bipartite mutual information scaling. (a, b) Average KL-divergence per token for models trained on different sequence lengths. (c) Average KL-divergence per token as a function of the ratio between bipartite mutual information and Mamba recurrent state sizes.

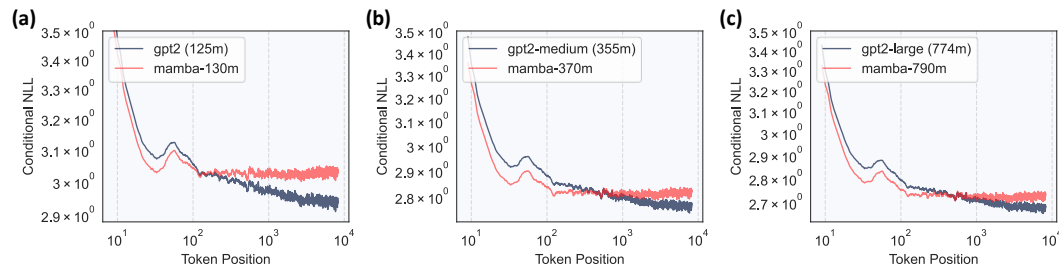


Figure 4: Position-wise conditional negative log likelihood (NLL) evaluation for models trained on 8192-token sequences on the PG19 dataset [90].

6 Empirical Verification

Sub-volume-law Gaussian Test. We first validate our theory using a synthetic dataset comprising a family of multivariate Gaussian distributions (see Appx. C for details). This distribution family closely mimics the scaling of both bipartite and two-point mutual information observed in natural language, while crucially allowing for the efficient calculation of conditional probabilities and KL divergences—calculations that would be intractable with real-world natural language datasets. Furthermore, the synthetic data enables an isolated assessment of a model’s ability to handle long sequences, without interference from its capacity to understand the semantic meanings of natural language.

In Fig. 3, we present the average per-token KL divergence (defined in Appx. F.IV) for GPT2, Mamba, and Mamba2 models, serving as representative transformer and SSM architectures. Panels (a) and (b) show that GPT2 maintains consistent KL divergence across different sequence lengths. In contrast, smaller Mamba and Mamba2 models exhibit increasing difficulty with longer contexts, necessitating substantially larger model sizes to achieve comparable performance at a sequence length of 4096. Panel (c) offers direct confirmation of our theoretical framework: it plots KL divergence against the ratio of bipartite mutual information to the recurrent state size for Mamba models of varying configurations. For this, we varied sequence lengths from 64 to 16,384 and model sizes from 50M to 1.4B parameters. The KL divergence values from these diverse configurations remarkably collapse onto a single curve, demonstrating that model performance depends only on the ratio $I^{\text{BP}} / \dim(\mathbf{z})$. This finding precisely confirms our theory that for effective long-context modeling, a model’s history state size must scale at least as fast as the bipartite mutual information present in the data.

These findings have important implications for modeling very long sequences. Extrapolating from the measured scaling in Fig. 2 (which likely underestimates the true exponent), the bipartite mutual information for a sequence of one million tokens could exceed 60,000 nats. Our results in Fig. 3(c) suggest that maintaining low KL divergence at such bipartite mutual information levels would require recurrent state dimensions approaching one million.

PG19 Test. We then extend our analysis to the PG19 dataset [90], a high-quality collection of pre-1919 books exhibiting long contextual dependencies.

In Fig. 4, we show the position-wise conditional negative log likelihood (NLL) of models trained on the PG19 dataset [90] with 8192-token sequences, where calculating KL-divergence is not feasible.

Note that, unlike conditional KL divergence, conditional NLL naturally decreases with token position (see Appx. F for details). Two key patterns emerge from this experiment: First, Mamba models typically outperform GPT2 models of comparable size at early token positions, but this advantage diminishes and eventually reverses at later positions. Most notably, Mamba's NLL tends to plateau beyond certain positions unless the model size is increased, while GPT2's NLL continues to improve. Second, the performance gap between Mamba and GPT2 narrows with increasing model size. Both observations align with our theoretical predictions: since Mamba's history state size remains fixed regardless of sequence position, its performance inevitably degrades beyond a certain token position unless model size increases. As model size grows, the history state size also increases, eventually becoming sufficient to capture the mutual information present in 8192-token sequences.

We note that Mamba's linear computational complexity can make larger Mamba models practically more efficient than smaller transformers. Our results should not be interpreted as suggesting Mamba's architectural inferiority. Rather, they demonstrate how different architectures handle long sequences differently, and that a model's capacity for capturing long-range dependencies aligns with our theoretical L^2M framework, regardless of the architecture.

Additional experimental results can be found in Appx. G.

7 Discussion

The L^2M condition establishes a fundamental relationship between the information structure of data and architectural requirements. This relationship manifests differently across architectures: transformers with linearly growing key-value caches naturally satisfy the condition as single models (given our measured sublinear mutual information scaling with $\beta < 1$), though at quadratic computational cost, while SSMs and similar fixed-state architectures require model size to scale with sequence length to achieve comparable mutual information capability.

Interestingly, transformers appear to *over-provision* their history state relative to the measured mutual information scaling: their linear growth exceeds the sublinear (L^β with $\beta < 1$) scaling we observe. This observation provides a clear goal for future architecture design. Although it remains unclear whether the over-provisioning is necessary for other aspects of language modeling beyond pure information storage, the gap between the linear growth of transformers and the L^β requirement suggests a concrete target: architectures that precisely match the required sublinear scaling could potentially achieve substantially improved efficiency while maintaining the capacity to capture long-range dependencies.

Our framework applies to autoregressive language models, which encompass the vast majority of widely-used LLMs. While diffusion-based language models represent an alternative generative paradigm, they typically still operate autoregressively at a higher level of granularity, making our framework applicable in practice. Extending our framework to hybrid architectures that combine different mechanisms represents an important research direction that could unify our understanding of how diverse architectural choices affect long-context capabilities. Applying our framework to other sequential domains, such as biological sequences like proteins or DNA, or computer code, also presents a particularly promising direction, as different mutual information scaling behaviors in these domains could provide a principled explanation for the observed differences in model requirements across domains.

Additional discussions on limitations and broader impacts can be found in Appx. H and I.

8 Conclusion

We establish a bipartite mutual information scaling law that characterizes long-range dependencies in natural language and introduce the L^2M condition, which lower bounds the necessary scaling of a model's history state for effective long-context modeling. By identifying the minimum required growth rate of the history state, our work provides a principled foundation for understanding how different architectures handle long contexts. This framework establishes concrete, information-theoretical, and data-driven targets that could guide the design of architectures balancing computational efficiency with the capacity to capture long-range dependencies in natural language and potentially beyond.

Acknowledgements

The authors acknowledge support from the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions). Z.C. acknowledges support from the Mathworks Fellowship. Z.C. thanks Rumen Dangovski for helpful discussions. Z.C. and O.M. thank Amazon Web Services account team, including Brian McCarthy and Jared Novotny, for technical support. The research was sponsored by the United States Air Force Research Laboratory and the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The computations in this paper were partly run on the FASRC cluster supported by the FAS Division of Science Research Computing Group at Harvard University. This research used the DeltaAI advanced computing and data resource, which is supported by the National Science Foundation (award OAC 2320345) and the State of Illinois, through allocation CIS240904 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296, and through the National Artificial Intelligence Research Resource (NAIRR) Pilot NAIRR250043.

References

- [1] Brown, T. *et al.* Language Models are Few-Shot Learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H. (eds.) *Advances in Neural Information Processing Systems*, vol. 33, 1877–1901 (Curran Associates, Inc., 2020). URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- [2] Chowdhery, A. *et al.* PaLM: Scaling Language Modeling with Pathways (2022). URL <https://arxiv.org/abs/2204.02311>. 2204.02311.
- [3] Touvron, H. *et al.* LLaMA: Open and Efficient Foundation Language Models (2023). URL <https://arxiv.org/abs/2302.13971>. 2302.13971.
- [4] Touvron, H. *et al.* Llama 2: Open Foundation and Fine-Tuned Chat Models (2023). URL <https://arxiv.org/abs/2307.09288>. 2307.09288.
- [5] OpenAI *et al.* GPT-4 Technical Report (2024). URL <https://arxiv.org/abs/2303.08774>. 2303.08774.
- [6] Team, G. *et al.* Gemini: A family of highly capable multimodal models (2025). URL <https://arxiv.org/abs/2312.11805>. 2312.11805.
- [7] Comanici, G. *et al.* Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities (2025). URL <https://arxiv.org/abs/2507.06261>. 2507.06261.
- [8] Grattafiori, A. *et al.* The Llama 3 Herd of Models (2024). URL <https://arxiv.org/abs/2407.21783>. 2407.21783.
- [9] DeepSeek-AI *et al.* DeepSeek-V3 Technical Report (2024). URL <https://arxiv.org/abs/2412.19437>. 2412.19437.
- [10] Guo, D. *et al.* Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature* **645**, 633–638 (2025). URL <https://doi.org/10.1038/s41586-025-09422-z>.
- [11] Qwen *et al.* Qwen2.5 technical report (2025). URL <https://arxiv.org/abs/2412.15115>. 2412.15115.
- [12] Yang, A. *et al.* Qwen3 technical report (2025). URL <https://arxiv.org/abs/2505.09388>. 2505.09388.
- [13] Stiennon, N. *et al.* Learning to summarize from human feedback (2020).
- [14] Yuan, A., Coenen, A., Reif, E. & Ippolito, D. Wordcraft: Story Writing With Large Language Models (2022).

- [15] Gou, Z. *et al.* ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving (2023).
- [16] Wang, Y. *et al.* CodeT5+: Open Code Large Language Models for Code Understanding and Generation (2023).
- [17] Bubeck, S. *et al.* Sparks of Artificial General Intelligence: Early experiments with GPT-4 (2023).
- [18] Ge, Y. *et al.* OpenAGI: When LLM Meets Domain Experts (2023).
- [19] Kosinski, M. Theory of Mind May Have Spontaneously Emerged in Large Language Models (2023).
- [20] Wei, J. *et al.* Chain of Thought Prompting Elicits Reasoning in Large Language Models. In Oh, A. H., Agarwal, A., Belgrave, D. & Cho, K. (eds.) *Advances in Neural Information Processing Systems* (2022). URL https://openreview.net/forum?id=_VjQ1MeSB_J.
- [21] Wang, J., Meng, F., Liang, Y. & Zhou, J. DRT-o1: Optimized Deep Reasoning Translation via Long Chain-of-Thought (2024). URL <https://arxiv.org/abs/2412.17498>. 2412.17498.
- [22] Katharopoulos, A., Vyas, A., Pappas, N. & Fleuret, F. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention (2020). URL <https://arxiv.org/abs/2006.16236>. 2006.16236.
- [23] Gu, A., Goel, K. & Re, C. Efficiently Modeling Long Sequences with Structured State Spaces. In *International Conference on Learning Representations* (2022). URL <https://openreview.net/forum?id=uYLFoz1v1AC>.
- [24] Zhu, C. *et al.* Long-Short Transformer: Efficient Transformers for Language and Vision (2021). URL <https://arxiv.org/abs/2107.02192>. 2107.02192.
- [25] Beltagy, I., Peters, M. E. & Cohan, A. Longformer: The Long-Document Transformer (2020). URL <https://arxiv.org/abs/2004.05150>. 2004.05150.
- [26] Ding, J. *et al.* LongNet: Scaling Transformers to 1,000,000,000 Tokens (2023). URL <https://arxiv.org/abs/2307.02486>. 2307.02486.
- [27] Gu, A. & Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces (2024). URL <https://openreview.net/forum?id=AL1fq05o7H>.
- [28] Dao, T. & Gu, A. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality (2024). URL <https://arxiv.org/abs/2405.21060>. 2405.21060.
- [29] Peng, B. *et al.* RWKV: Reinventing RNNs for the Transformer Era. In *The 2023 Conference on Empirical Methods in Natural Language Processing* (2023). URL <https://openreview.net/forum?id=7SaXczaBpG>.
- [30] Sun, Y. *et al.* Learning to (Learn at Test Time): RNNs with Expressive Hidden States (2024). URL <https://arxiv.org/abs/2407.04620>. 2407.04620.
- [31] Guo, H. *et al.* Log-linear attention (2025). URL <https://arxiv.org/abs/2506.04761>. 2506.04761.
- [32] Ebeling, W. & Pöschel, T. Entropy and Long-Range Correlations in Literary English (1994).
- [33] Debowski, L. Excess entropy in natural language: present state and perspectives (2011).
- [34] Bentz, C., Alikaniotis, D., Cysouw, M. & i Cancho, R. F. The Entropy of Words - Learnability and Expressivity across More than 1000 Languages (2017).

- [35] Futrell, R., Qian, P., Gibson, E., Fedorenko, E. & Blank, I. Syntactic dependencies correspond to word pairs with high mutual information. In Gerdes, K. & Kahane, S. (eds.) *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, 3–13 (Association for Computational Linguistics, Paris, France, 2019). URL <https://aclanthology.org/W19-7703/>.
- [36] Poole, B., Ozair, S., van den Oord, A., Alemi, A. A. & Tucker, G. On Variational Bounds of Mutual Information (2019). URL <https://arxiv.org/abs/1905.06922>. 1905.06922.
- [37] Belghazi, M. I. *et al.* MINE: Mutual Information Neural Estimation (2021). URL <https://arxiv.org/abs/1801.04062>. 1801.04062.
- [38] Kraskov, A., Stögbauer, H. & Grassberger, P. Estimating mutual information. *Phys. Rev. E* **69**, 066138 (2004). URL <https://link.aps.org/doi/10.1103/PhysRevE.69.066138>.
- [39] Gao, S., Steeg, G. V. & Galstyan, A. Efficient Estimation of Mutual Information for Strongly Dependent Variables (2015). URL <https://arxiv.org/abs/1411.2003>. 1411.2003.
- [40] Cheng, P. *et al.* CLUB: A Contrastive Log-ratio Upper Bound of Mutual Information (2020). URL <https://arxiv.org/abs/2006.12013>. 2006.12013.
- [41] Brown, G., Pocock, A., Zhao, M.-J. & Luján, M. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *Journal of Machine Learning Research* **13**, 27–66 (2012). URL <http://jmlr.org/papers/v13/brown12a.html>.
- [42] Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S. & Lucic, M. On Mutual Information Maximization for Representation Learning. In *International Conference on Learning Representations* (2020). URL <https://openreview.net/forum?id=rkxoh24FPH>.
- [43] Chen, X. *et al.* InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I. & Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 29 (Curran Associates, Inc., 2016). URL https://proceedings.neurips.cc/paper_files/paper/2016/file/7c9d0b1f96aebd7b5eca8c3edaa19ebb-Paper.pdf.
- [44] Hjelm, R. D. *et al.* Learning deep representations by mutual information estimation and maximization (2019). URL <https://arxiv.org/abs/1808.06670>. 1808.06670.
- [45] Gelbukh, A. & Sidorov, G. Zipf and Heaps Laws’ Coefficients Depend on Language. In *Conference on Intelligent Text Processing and Computational Linguistics* (2001). URL <https://api.semanticscholar.org/CorpusID:20344161>.
- [46] Hilberg, W. Der bekannte Grenzwert der redundanzfreien Information in Texten - eine Fehlinterpretation der Shannonschen Experimente? *Frequenz* **44**, 243 – 248 (1990). URL <https://api.semanticscholar.org/CorpusID:124878549>.
- [47] Hoffmann, J. *et al.* Training compute-optimal large language models (2022). URL <https://arxiv.org/abs/2203.15556>. 2203.15556.
- [48] Kaplan, J. *et al.* Scaling Laws for Neural Language Models (2020). URL <https://arxiv.org/abs/2001.08361>. 2001.08361.
- [49] Biderman, S. *et al.* Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling (2023). URL <https://arxiv.org/abs/2304.01373>. 2304.01373.
- [50] Bahri, Y., Dyer, E., Kaplan, J., Lee, J. & Sharma, U. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences* **121**, e2311878121 (2024). URL <https://www.pnas.org/doi/abs/10.1073/pnas.2311878121>. <https://www.pnas.org/doi/pdf/10.1073/pnas.2311878121>.
- [51] Bordelon, B., Atanasov, A. & Pehlevan, C. A Dynamical Model of Neural Scaling Laws (2024). URL <https://arxiv.org/abs/2402.01092>. 2402.01092.

- [52] Nayak, A. K. & Varshney, L. R. An information theory of compute-optimal size scaling, emergence, and plateaus in language models (2024). URL <https://arxiv.org/abs/2410.01243>. 2410.01243.
- [53] Basu, S., Choraria, M. & Varshney, L. R. Transformers are universal predictors (2023). URL <https://arxiv.org/abs/2307.07843>. 2307.07843.
- [54] Zhou, R., Tian, C. & Diggavi, S. Transformers learn variable-order markov chains in-context (2024). URL <https://arxiv.org/abs/2410.05493>. 2410.05493.
- [55] Child, R., Gray, S., Radford, A. & Sutskever, I. Generating Long Sequences with Sparse Transformers (2019). URL <https://arxiv.org/abs/1904.10509>. 1904.10509.
- [56] Zaheer, M. *et al.* Big Bird: Transformers for Longer Sequences. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H. (eds.) *Advances in Neural Information Processing Systems*, vol. 33, 17283–17297 (Curran Associates, Inc., 2020). URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf.
- [57] Dai, Z. *et al.* Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context (2019). URL <https://arxiv.org/abs/1901.02860>. 1901.02860.
- [58] Rae, J. W., Potapenko, A., Jayakumar, S. M. & Lillicrap, T. P. Compressive Transformers for Long-Range Sequence Modelling (2019). URL <https://arxiv.org/abs/1911.05507>. 1911.05507.
- [59] Sukhbaatar, S., Grave, E., Bojanowski, P. & Joulin, A. Adaptive Attention Span in Transformers (2019). URL <https://arxiv.org/abs/1905.07799>. 1905.07799.
- [60] He, Z., Qin, Z., Prakriya, N., Sun, Y. & Cong, J. HMT: Hierarchical Memory Transformer for Long Context Language Processing (2024). URL <https://arxiv.org/abs/2405.06067>. 2405.06067.
- [61] Gu, A., Goel, K. & Ré, C. Efficiently Modeling Long Sequences with Structured State Spaces (2022). URL <https://arxiv.org/abs/2111.00396>. 2111.00396.
- [62] Beck, M. *et al.* xLSTM: Extended Long Short-Term Memory (2024). URL <https://arxiv.org/abs/2405.04517>. 2405.04517.
- [63] De, S. *et al.* Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models (2024). URL <https://arxiv.org/abs/2402.19427>. 2402.19427.
- [64] Dao, T., Fu, D. Y., Ermon, S., Rudra, A. & Ré, C. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness (2022). URL <https://arxiv.org/abs/2205.14135>. 2205.14135.
- [65] Dao, T. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning (2023). URL <https://arxiv.org/abs/2307.08691>. 2307.08691.
- [66] Shah, J. *et al.* FlashAttention-3: Fast and Accurate Attention with Asynchrony and Low-precision (2024). URL <https://arxiv.org/abs/2407.08608>. 2407.08608.
- [67] Qin, Z. *et al.* Lightning Attention-2: A Free Lunch for Handling Unlimited Sequence Lengths in Large Language Models (2024). URL <https://arxiv.org/abs/2401.04658>. 2401.04658.
- [68] Kwon, W. *et al.* Efficient Memory Management for Large Language Model Serving with PagedAttention (2023). URL <https://arxiv.org/abs/2309.06180>. 2309.06180.
- [69] Nye, M. *et al.* Show Your Work: Scratchpads for Intermediate Computation with Language Models (2021). URL <https://arxiv.org/abs/2112.00114>. 2112.00114.
- [70] Tishby, N. & Zaslavsky, N. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, 1–5 (2015).

- [71] Goldfeld, Z. & Polyanskiy, Y. The Information Bottleneck Problem and its Applications in Machine Learning. *IEEE Journal on Selected Areas in Information Theory* **1**, 19–38 (2020).
- [72] Chen, Z. & Luo, D. Entangling Intelligence: AI-Quantum Crossovers and Perspectives. In *2024 IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, 516–519 (IEEE, 2024). URL <https://ieeexplore.ieee.org/document/10835527>.
- [73] Luo, D. & Clark, B. K. Backflow Transformations via Neural Networks for Quantum Many-Body Wave Functions. *Phys. Rev. Lett.* **122**, 226401 (2019). URL <https://link.aps.org/doi/10.1103/PhysRevLett.122.226401>.
- [74] Luo, D. *et al.* Gauge-invariant and anyonic-symmetric autoregressive neural network for quantum lattice models. *Physical Review Research* **5**, 013216 (2023). URL <https://journals.aps.org/prresearch/abstract/10.1103/PhysRevResearch.5.013216>.
- [75] Wang, J. *et al.* Spacetime neural network for high dimensional quantum dynamics. *arXiv preprint arXiv:2108.02200* (2021). URL <https://arxiv.org/abs/2108.02200>.
- [76] Chen, Z., Luo, D., Hu, K. & Clark, B. K. Simulating 2+ 1d lattice quantum electrodynamics at finite density with neural flow wavefunctions. *arXiv preprint arXiv:2212.06835* (2022). URL <https://arxiv.org/abs/2212.06835>.
- [77] Lami, G., Carleo, G. & Collura, M. Matrix product states with backflow correlations. *Phys. Rev. B* **106**, L081111 (2022). URL <https://link.aps.org/doi/10.1103/PhysRevB.106.L081111>.
- [78] Wu, D., Rossi, R., Vicentini, F. & Carleo, G. From tensor-network quantum states to tensorial recurrent neural networks. *Physical Review Research* **5** (2023). URL <http://dx.doi.org/10.1103/PhysRevResearch.5.L032001>.
- [79] Chen, Z., Newhouse, L., Chen, E., Luo, D. & Soljacic, M. ANTNet: Bridging autoregressive neural networks and tensor networks for quantum many-body simulation. *Advances in neural information processing systems* **36**, 450–476 (2023). URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/01772a8b0420baec00c4d59fe2fbace6-Abstract-Conference.html.
- [80] Dugan, O. *et al.* OccamLLM: Fast and Exact Language Model Arithmetic in a Single Step. *Advances in Neural Information Processing Systems* **37**, 35665–35699 (2024). URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/3eceb70f47690051d6769739fbf6294b-Abstract-Conference.html.
- [81] Stokes, J., Izaac, J., Killoran, N. & Carleo, G. Quantum Natural Gradient. *Quantum* **4**, 269 (2020). URL <https://doi.org/10.22331/q-2020-05-25-269>.
- [82] Chen, Z., McCarran, J., Vizcaino, E., Soljacic, M. & Luo, D. TENG: Time-Evolving Natural Gradient for Solving PDEs With Deep Neural Nets Toward Machine Precision. In *International Conference on Machine Learning*, 7143–7162 (PMLR, 2024). URL <https://proceedings.mlr.press/v235/chen24ad.html>.
- [83] Chen, Z. *et al.* QuanTA: Efficient high-rank fine-tuning of llms with quantum-informed tensor adaptation. *Advances in Neural Information Processing Systems* **37**, 92210–92245 (2024). URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a7c17115db36193f6b83b71b0fe1d416-Abstract-Conference.html.
- [84] Carleo, G. *et al.* Machine learning and the physical sciences. *Rev. Mod. Phys.* **91**, 045002 (2019). URL <https://link.aps.org/doi/10.1103/RevModPhys.91.045002>.
- [85] Lee, C.-H. *et al.* Deep Learning Enabled Strain Mapping of Single-Atom Defects in Two-Dimensional Transition Metal Dichalcogenides with Sub-Picometer Precision. *Nano Letters* **20**, 3369–3377 (2020). URL <https://doi.org/10.1021/acs.nanolett.0c00269>.

- [86] Luo, D., Chen, Z., Carrasquilla, J. & Clark, B. K. Autoregressive neural network for simulating open quantum systems via a probabilistic formulation. *Physical review letters* **128**, 090501 (2022). URL <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.128.090501>.
- [87] Moro, V. *et al.* Multimodal foundation models for material property prediction and discovery. *Newton* **1** (2025). URL [https://www.cell.com/newton/fulltext/S2950-6360\(25\)00008-8](https://www.cell.com/newton/fulltext/S2950-6360(25)00008-8).
- [88] Choi, S. *et al.* Photonic probabilistic machine learning using quantum vacuum noise. *Nature Communications* **15**, 7760 (2024). URL <https://www.nature.com/articles/s41467-024-51509-0>.
- [89] Meta. URL <https://ai.meta.com/blog/meta-llama-3-1/>.
- [90] Rae, J. W., Potapenko, A., Jayakumar, S. M., Hillier, C. & Lillicrap, T. P. Compressive Transformers for Long-Range Sequence Modelling. In *International Conference on Learning Representations* (2020). URL <https://openreview.net/forum?id=SylKikSYDH>.
- [91] Bialek, W. & Tishby, N. Predictive Information (1999). URL <https://arxiv.org/abs/cond-mat/9902341>.
- [92] Ebeling, W. & Pöschel, T. Entropy and Long-Range Correlations in Literary English. *Europhysics Letters* **26**, 241 (1994). URL <https://dx.doi.org/10.1209/0295-5075/26/4/001>.
- [93] Ebeling, W. & Neiman, A. Long-range correlations between letters and sentences in texts. *Physica A: Statistical Mechanics and its Applications* **215**, 233–241 (1995). URL <https://www.sciencedirect.com/science/article/pii/0378437195000253>.
- [94] MONTEMURRO, M. A. & PURY, P. A. LONG-RANGE FRACTAL CORRELATIONS IN LITERARY CORPORA. *Fractals* **10**, 451–461 (2002). URL <https://doi.org/10.1142/S0218348X02001257>. <https://doi.org/10.1142/S0218348X02001257>.
- [95] Shen, H. Mutual Information Scaling and Expressive Power of Sequence Models (2019). URL <https://arxiv.org/abs/1905.04271>.
- [96] Lin, H. W. & Tegmark, M. Critical Behavior in Physics and Probabilistic Formal Languages. *Entropy* **19** (2017). URL <https://www.mdpi.com/1099-4300/19/7/299>.
- [97] Stanley, H. E. Power laws and universality. *Nature* **378**, 554–555 (1995).
- [98] Łukasz Debowski. The Relaxed Hilberg Conjecture: A Review and New Experimental Support. *Journal of Quantitative Linguistics* **22**, 311–337 (2015). URL <https://doi.org/10.1080/09296174.2015.1106268>. <https://doi.org/10.1080/09296174.2015.1106268>.
- [99] Lu, S., Kanász-Nagy, M., Kukuljan, I. & Cirac, J. I. Tensor networks and efficient descriptions of classical data (2024). URL <https://arxiv.org/abs/2103.06872>.
- [100] van den Oord, A., Li, Y. & Vinyals, O. Representation learning with contrastive predictive coding (2019). URL <https://arxiv.org/abs/1807.03748>.
- [101] Foundation, W. Wikimedia downloads. URL <https://dumps.wikimedia.org>.
- [102] Grassberger, P. Entropy Estimates from Insufficient Samplings (2008). URL <https://arxiv.org/abs/physics/0307138>.
- [103] Inc., W. R. Mathematica, Version 14.2. URL <https://www.wolfram.com/mathematica>. Champaign, IL, 2024.
- [104] Donsker, M. D. & Varadhan, S. S. Asymptotic evaluation of certain markov process expectations for large time, i. *Communications on pure and applied mathematics* **28**, 1–47 (1975).

- [105] Elhage, N. *et al.* Toy Models of Superposition (2022). URL <https://arxiv.org/abs/2209.10652>. 2209.10652.
- [106] Park, K., Choe, Y. J. & Veitch, V. The Linear Representation Hypothesis and the Geometry of Large Language Models (2023).
- [107] Jiang, Y., Rajendran, G., Ravikumar, P., Aragam, B. & Veitch, V. On the Origins of Linear Representations in Large Language Models (2024).
- [108] Kabatiansky, G. A. & Levenshtein, V. I. On bounds for packings on a sphere and in space. *Problemy Peredachi Informatsii* **14**, 3–25 (1978). URL <http://mi.mathnet.ru/ppi1518>. English translation: *Problems Inform. Transmission*, vol. 14, no. 1, pp. 1–17, 1978.
- [109] Cohn, H. & Zhao, Y. Sphere packing bounds via spherical codes. *Duke Mathematical Journal* **163** (2014). URL <http://dx.doi.org/10.1215/00127094-2738857>.
- [110] Kawabata, T. & Dembo, A. The rate-distortion dimension of sets and measures. *IEEE Transactions on Information Theory* **40**, 1564–1572 (1994).
- [111] Black, S. *et al.* GPT-NeoX-20B: An Open-Source Autoregressive Language Model (2022). URL <https://arxiv.org/abs/2204.06745>. 2204.06745.
- [112] Wolf, T. *et al.* HuggingFace’s Transformers: State-of-the-art Natural Language Processing (2020). URL <https://arxiv.org/abs/1910.03771>. 1910.03771.
- [113] Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library (2019). URL <https://arxiv.org/abs/1912.01703>. 1912.01703.
- [114] Li, J. *et al.* FlexAttention for Efficient High-Resolution Vision-Language Models (2024). URL <https://arxiv.org/abs/2407.20228>. 2407.20228.

A Notations and Conventions

Basic notations

- Random variables: Uppercase letters (e.g., X, Y, W) denote random variables.
- Realizations: Lowercase letters (e.g., x, y, w) denote specific values or realizations of random variables.
- Sequences: $X_{i:j}$ denotes the sequence $(X_i, X_{i+1}, \dots, X_j)$.
- Bold notation: $\mathbf{X} := X_{i:j}$ when indices are clear from context.
- Single variable shorthand: $X := X_i$ when the index is clear from context.
- Logarithms: While the choice of base does not affect the scaling laws (only the multiplicative constants), all logarithms are natural logarithms (base e) unless otherwise specified.

Information-theoretic quantities

- $H(\cdot)$: Shannon entropy.
- $H(\cdot|\cdot)$: Conditional (Shannon) entropy.
- $I(\cdot;\cdot)$: (Shannon) mutual information.
- $D_{\text{KL}}(\cdot||\cdot)$: Kullback–Leibler divergence.
- $H^p(\cdot)$: Entropy computed with respect to distribution p .
- $H^q(\cdot)$: Entropy computed with respect to distribution q .
- $H(p, q)$: Cross-entropy between distributions p and q .

Asymptotic notations

- $f(n) \sim g(n)$: f and g have the same asymptotic growth rate, i.e., $f(n) = \Theta(g(n))$.
- $f(n) \succ g(n)$: f grows strictly faster than g asymptotically, i.e., $f(n) = \omega(g(n))$.
- $f(n) \succeq g(n)$: f grows at least as fast as g asymptotically, i.e., $f(n) = \Omega(g(n))$.
- $f(n) \prec g(n)$: f grows strictly slower than g asymptotically, i.e., $f(n) = o(g(n))$.
- $f(n) \preceq g(n)$: f grows at most as fast as g asymptotically, i.e., $f(n) = O(g(n))$.

Distributions and expectations

- p : True underlying probability distribution (of natural language).
- q : Model-generated probability distribution (sometimes refers to the model itself).
- $\mathbb{E}_p[\cdot]$: Expectation with respect to distribution p .
- $p_X \otimes p_Y$: Product distribution of marginals p_X and p_Y .

Model-specific notations

- w_{BOS} : Beginning-of-sequence token.
- M : Vocabulary size.
- L : Sequence length.
- ℓ : Position of sequence split for bipartite mutual information.
- $\dim(\mathbf{z})$: Dimensionality of the history state $\mathbf{z}$.
- θ : Model parameters.

Special notations

- I^{BP} : Bipartite mutual information.
- I^{TP} : Two-point mutual information.
- $\mathbb{1}(\cdot)$: Indicator function (equals 1 when condition is true, 0 otherwise).

The notations are used consistently throughout the main text and appendices unless otherwise specified in local contexts.

B Additional Details on Mutual Information Scalings

B.I Bipartite Mutual Information Scaling with Additional LLMs on Additional Datasets

In the main text, we use the LLaMA 3.1 405B model as the density estimator and measured the bipartite mutual information scaling on PG19 dataset. In this section, we provide additional estimations of the bipartite mutual information scaling using the DeepSeek V3 Base model and on WIKIPEDIA dataset. We note that because we are merely measuring the conditional probabilities of the input tokens without interactions with the agent, we believe the non-instruction-finetuned model better suits our tasks.

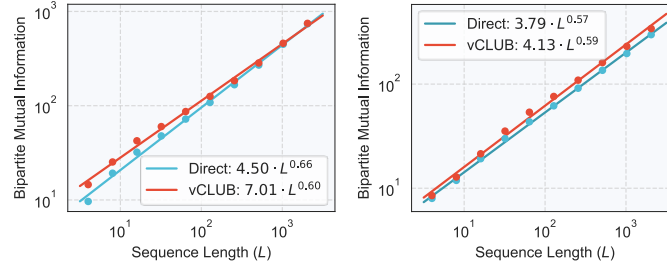


Figure B.1: Bipartite mutual information estimation using (left) LLaMA 3.1 405B on the WIKIPEDIA dataset and (right) Deepseek V3 Base model on the PG19 dataset. All direct measurements include the bias correction described in Appx. B.IV.

In Fig. B.1, we report the results on WIKIPEDIA dataset using LLaMA 3.1 405B model as the density estimator and on PG19 dataset using Deepseek V3 Base model as the density estimator. We find that in both cases, clear sub-volume growth behavior is observed. We note that the measured exponent should be taken with a grain of salt and likely underestimates the true mutual information scaling due to reasons explained in Appx. B.III.

B.II Bipartite Mutual Information Scaling Under Various Ratios of ℓ/L

In the main text, we focused on the bipartite mutual information with equal splits. However, the bipartite mutual information scaling is not limited to equal bipartition. In this section, we provide additional results for various ratios of ℓ/L .

In Fig. B.2, we provide estimation of the bipartite mutual information scaling for $\ell/L = 3$ and $\ell/L = 4$. All results show clear power-law relations, and are consistent with Fig. 2 in the main text. These results can be used to support the L^2M condition with similar arguments as in the main text.

B.III Why The Estimated Exponent β Is Likely An Underestimation?

In the main text, we mentioned that our measured exponent β using LLMs likely underestimates the true β . Here, we discuss the reasons.

For the direct estimator,

$$I_{\ell;L}^{\text{BP,direct}} = H(p_{\mathbf{Y}}, q_{\mathbf{Y}}) - H(p_{\mathbf{Y}|\mathbf{X}}, q_{\mathbf{Y}|\mathbf{X}}), \quad (\text{B.1})$$

both terms (without the minus sign) overestimates the true (conditional) entropy, but for different extent and at different scales.

At small L , the first term suffers from the bias from the BOS token as discussed in Appx. B.IV. The second term, despite also an overestimation, does not suffer from the BOS token issue. Therefore, at small L , the direct estimator tends to overestimate the true entropy.

At large L , the bias from the BOS token is less severe. However, modeling $p(\mathbf{Y}|\mathbf{X})$ requires the model to correctly capture all the dependencies between $\mathbf{X}$ and $\mathbf{Y}$, making it significantly harder than modeling $p(\mathbf{Y})$ alone. Therefore, $q(\mathbf{Y}|\mathbf{X})$ is likely a worse estimation of the true distribution than $q(\mathbf{Y})$, resulting in more overestimation in the second term, and an underestimation of the bipartite mutual information.

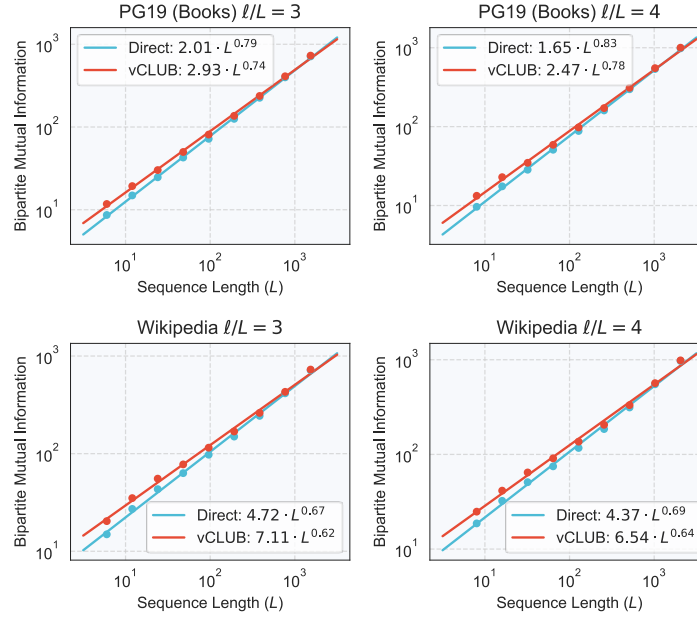


Figure B.2: Bipartite mutual information estimation using different ratios of ℓ/L . All results suggest the existence of power-law scaling, with various fitted exponents.

This means that the direct estimator tends to overestimate the true bipartite mutual information at small L and underestimate it at large L , resulting in an underestimation of the fitted exponent.

The vCLUB estimator, as pointed out in [40], is an upper bound to the true mutual information if q is close to p , but fails to maintain the property when the KL-divergence between them increases. Therefore, it is likely that this estimator also overestimates the true bipartite mutual information at small L and underestimates it at large L , resulting in a similar underestimation of the fitted exponent as our direct estimator. As our fitted exponent for the vCLUB estimator is smaller than that of the direct estimator, we conclude that the vCLUB estimator has a larger bias in this case, and it is reasonable to believe that the true exponent is even larger.

B.IV Direct Estimation of Bipartite Mutual Information Using LLMs

In the main text, our direct estimator for the bipartite mutual information is

$$I_{\ell;L}^{\text{BP,direct}} = \mathbb{E}_{p_{\mathbf{X}\mathbf{Y}}} [\log q(\mathbf{Y}|\mathbf{X}) - \log q(\mathbf{Y})] = H(p_{\mathbf{Y}}, q_{\mathbf{Y}}) - H(p_{\mathbf{Y}|\mathbf{X}}, q_{\mathbf{Y}|\mathbf{X}}) = I^p(\mathbf{X}; \mathbf{Y}) + \varepsilon(p, q). \quad (\text{B.2})$$

where as usual, $\mathbf{X} := X_{1:\ell} := W_{1:\ell}$ and $\mathbf{Y} := Y_{1:L-\ell} := W_{\ell+1:L}$ with $W_{1:L}$ being a sequence of tokens. However, as discussed in the main text, the $H(p_{\mathbf{Y}}, q_{\mathbf{Y}})$ term suffers from an additional bias—we cannot guarantee that $\mathbf{Y}$ starts at the beginning of a sentence, but LLMs model distributions conditioned on BOS token. To mitigate this issue, we use n -gram calculations to correct the entropy of the first two tokens as explained below.

We first rewrite the (marginal) cross entropy as

$$H(p_{\mathbf{Y}}, q_{\mathbf{Y}}) = -\mathbb{E}_p[\log q(\mathbf{Y})] = -\sum_{i=1}^{L-\ell} \mathbb{E}_p[\log q(Y_i|Y_{1:i-1})], \quad (\text{B.3})$$

where as usual, the expectation over the conditional variable is omitted but implied in the cross entropy calculation.

In modern LLMs, we can only compute $q(y_i|y_{1:i-1}, w_{\text{BOS}}) \neq q(y_i|y_{1:i-1})$, resulting in an additional error in the bipartite mutual information estimation. In practice, this difference becomes less pronounced for larger i , because it matters less if the sequence starts at the beginning of a sequence or not if there are many $y_{1:i-1}$ prior tokens to conditional on. Therefore, we focusing on reducing the

bias for small i . In addition, if i is small, we can iterate over the dataset and construct a histogram for the i -gram distribution $p(y_{1:i})$.

We denote the count for each i -tuple of tokens with $n_{y_{1:i}}$ and the total number of samples with N . Then, the entropy of the distribution can be estimated naively as

$$\hat{H}^{\text{naive}}(Y_{1:i}) = - \sum_{y_{1:i}} \frac{n_{y_{1:i}}}{N} \log \frac{n_{y_{1:i}}}{N} = \log N - \frac{1}{N} \sum_{y_{1:i}} n_{y_{1:i}} \log n_{y_{1:i}}, \quad (\text{B.4})$$

where the summation runs over all possible combination of tokens $y_{1:i} := (y_1, y_2, \dots, y_i)$.

However, this estimation is severely biased and underestimates the true entropy, due to the concavity of logarithm function. In [102], a bias-corrected estimator is proposed by replacing the logarithm function with a new function

$$\hat{H}^G(Y_{1:i}) = \log N - \frac{1}{N} \sum_{y_{1:i}} n_{y_{1:i}} G(n_{y_{1:i}}), \quad (\text{B.5})$$

where

$$G(n) = \psi(n) + \frac{(-1)^n}{2} \left(\psi\left(\frac{n+1}{2}\right) - \psi\left(\frac{n}{2}\right) \right), \quad (\text{B.6})$$

with $\psi(\cdot)$ the digamma function. We note that Ref. [102] was not able to obtain the closed form expression for $G(\cdot)$, which we derived with the help of Wolfram Mathematica [103].

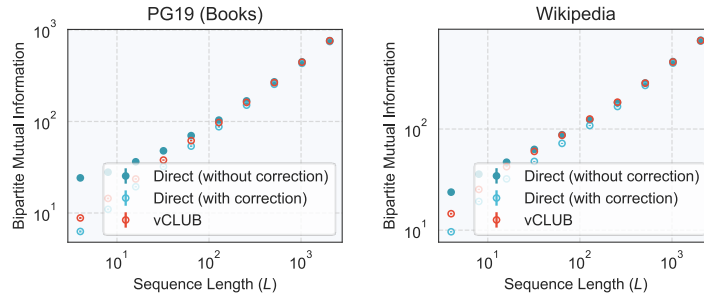


Figure B.3: Effect of bias correction method in the direct estimator. The bias only affects the estimation at small sequence lengths, and all methods converge at large sequence lengths.

This bias-corrected estimator still underestimates the true entropy, but much less compared to the original naïve estimator. In the main text, we estimate the (marginal) cross entropy with 2-gram correction in the following way. Breaking up the cross entropy as

$$H(p_{\mathbf{Y}}, q_{\mathbf{Y}}) = - \sum_{i=3}^{L-\ell} \mathbb{E}_p[\log q(Y_i|Y_{1:i-1})] + H(p_{Y_1 Y_2}, q_{Y_1 Y_2}). \quad (\text{B.7})$$

For the first term, we use LLM generated $q(y_i|y_{1:i-1}, w_{BOS})$ as approximation. For the second term, we mitigate the bias from LLM estimation by combining it with Eq. (B.5) as $H(p_{Y_1 Y_2}, q_{Y_1 Y_2|w_{BOS}})/5 + 4\hat{H}_p^G(Y_1 Y_2)/5$. In Fig. B.3, we also present the result without this correction and show that this bias correction mostly affects the estimation at small lengths L , and does not alter the general scaling behavior. In addition, since the result from this bias-corrected direct estimator agrees with the vCLUB [40] estimator, we believe this correction is reasonable.

B.V Additional Discussion on Mutual Information Estimation Methods

In the main text, we briefly discussed the limitations of traditional mutual information estimation methods for our high-dimensional, long-sequence setting. Here we provide additional technical details on why these methods are challenging to apply to our settings.

Neural Estimators: MINE and InfoNCE. Neural estimators like MINE [37] and InfoNCE [100] train deep neural networks as critics to estimate mutual information. Both methods can fundamentally be viewed as training unnormalized density estimators or density ratio estimators.

MINE uses the Donsker–Varadhan representation of KL divergence [104] and trains a critic $T_\theta(x, y)$ to maximize:

$$\mathbb{E}_{p(x,y)}[T_\theta(x, y)] - \log \mathbb{E}_{p(x)p(y)}[e^{T_\theta(x,y)}] \quad (\text{B.8})$$

The optimal critic approximates the log density ratio $\log \frac{p(x,y)}{p(x)p(y)}$. However, this objective suffers from high variance and numerical instability when mutual information is large, which is especially challenging given the high-dimensional nature of long sequences we analyze.

InfoNCE uses noise-contrastive estimation with multiple negative samples:

$$I(X; Y) \geq \mathbb{E} \left[\log \frac{e^{f(x_i, y_i)}}{\frac{1}{K} \sum_{j=1}^K e^{f(x_i, y_j)}} \right] \quad (\text{B.9})$$

where the expectation is over K independent samples from the joint distribution. The bound is upper bounded by $\log K$, which means for our setting where mutual information can be on the order of thousands, this would require prohibitively large batch sizes to obtain accurate estimates.

Both methods require training critics from scratch to learn representations of natural language distributions, which could require datasets and computational resources comparable to training LLMs themselves.

Other variational bounds [36] face similar challenges.

K-Nearest Neighbor Estimators. K-nearest neighbor (K-NN) estimators [39] estimate mutual information based on distances between samples in joint and marginal spaces. While asymptotically unbiased and training-free, they also face challenges for text.

Text consists of discrete tokens that must be embedded into continuous spaces for K-NN estimation. Modern token embeddings have dimensions in the thousands, and for sequences of thousands of tokens, the combined dimensionality can make K-NN estimation impractical as the number of samples required for reliable K-NN estimates grows exponentially with dimension.

Connection to Our LLM-Based Approach. Our approach leverages pre-trained LLMs as density estimators, providing $q(y|x)$ directly through conditional probabilities and approximating $q(y)$ efficiently. This avoids training critics from scratch and the curse of dimensionality from distance-based estimation. We believe this is well-suited for our use case of analyzing long natural language sequences.

B.VI Estimation of Two-Point Mutual Information

In the main text, we included the results for two-point mutual information for completeness. In this section, we explain how the results are obtained.

Two-point mutual information estimation is more straightforward compared to bipartite mutual information, requiring only 1- and 2-gram statistics without LLM approximations. We estimate this quantity using entropy calculations for individual tokens and token pairs separated by distance d . Following [102], we employ their bias-reduced entropy estimator:

$$\hat{H}^G(X) = \hat{H}^G(Y) = \log N - \frac{1}{N} \sum_{m=1}^M n_m G(n_m), \quad (\text{B.10})$$

where N is the total number of tokens, M is the vocabulary size, n_m is the number of tokens whose token ID equals m , and $G(\cdot)$ is defined as

$$G(n) = \psi(n) + \frac{(-1)^n}{2} \left(\psi\left(\frac{n+1}{2}\right) - \psi\left(\frac{n}{2}\right) \right) \quad (\text{B.11})$$

with $\psi(\cdot)$ the digamma function.

The entropy of pairs of tokens is estimated analogously, with the summation running over all ordered pairs of tokens (m_i, m_j) , resulting in the total number of terms quadratic in the vocabulary size. The mutual information is then estimated as

$$\hat{I}_d^{\text{TP}}(X; Y) = \hat{H}^G(X) + \hat{H}^G(Y) - \hat{H}^G(XY). \quad (\text{B.12})$$

We note that this mutual information estimator exhibits systematic bias. The entropy estimator has a negative bias that increases (in absolute value) with dimension of the sample space $|\Omega|$. Since $|\Omega_X| = |\Omega_Y| = M$ where as $|\Omega_{XY}| = M^2$, the bias in $\hat{H}(XY)$ exceeds that in $\hat{H}(X) = \hat{H}(Y)$, leading to a positive bias in $\hat{I}_d$. This bias becomes particularly problematic at large distances d , where $H(XY) \approx H(X) + H(Y)$ and I_d approaches zero. To mitigate this issue, we perform additional bias correction by fitting the systematic bias from the data (see Appx. B.VII for details).

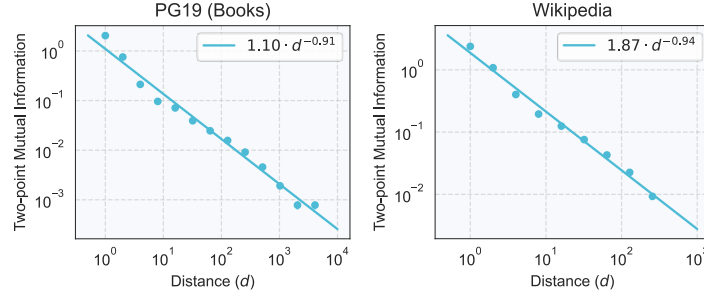


Figure B.4: Two-point mutual information scaling on PG19 and WIKIPEDIA datasets.

We apply this methodology to measure two-point mutual information on both the PG19 dataset and WIKIPEDIA, confirming power-law decay in both cases [Fig. B.4].

B.VII Bias Correction for Two-point Mutual Information

As discussed previously, the estimation of two-point mutual information can be calculated directly using the n -gram approximation [Eq. (B.5)], and compute the two-point mutual information as

$$\hat{I}_d^{\text{TP}}(X; Y) = \hat{H}^G(X) + \hat{H}^G(Y) - \hat{H}^G(XY). \quad (\text{B.13})$$

As discussed in Appx. B.IV, this entropy estimator has a negative bias, whose magnitude depends on the ratio $|\Omega|/N$, with $|\Omega|$ the size of the corresponding sample space. Since the sample space for the joint distribution is larger, it has a larger negative bias, resulting in a positive bias in $\hat{I}$. When d is small, this bias is relatively small compared to the mutual information itself. However, as d becomes larger, X and Y become less correlated, and $H(XY) \rightarrow H(X) + H(Y)$. In this case, the estimator can be dominated by this bias, and fitting for the power-law exponent becomes impossible.

To mitigate this issue, we propose a bias-corrected estimator.

$$\hat{I}_d^{\text{TP,corrected}}(X; Y) = \hat{H}^G(X) + \hat{H}^G(Y) - \hat{H}^G(XY) - C, \quad (\text{B.14})$$

where C is an unknown positive constant that does not depend on the distance d , which accounts for the bias of the original estimator.

To obtain this bias correction term and fit the power-law exponent, we minimize the following loss function

$$\sum_d (\log(\hat{I}_d^{\text{TP}} - C) - (\log A - \alpha \log d))^2, \quad (\text{B.15})$$

which is just $\hat{I}_d^{\text{TP}} = Ad^{-\alpha} + C$ fitted in log-log space. Then, we take the fitted C as the systematic bias and α as the fitted power-law exponent.

In Fig. B.5, we compare the corrected and uncorrected two-point mutual information as a function of d (only the corrected version is shown in the main text). Without the bias correction, the data appear to have larger long-range dependencies, but after the bias correction, all points lie on a straight line in a log-log plot. The bias correction constant is much smaller than the entropies involved in the calculation, even the smallest two-token entropy measured is 12.5, at least two orders of magnitude larger than the fitted bias correction. In addition, the fact that a single variable added to the fitting function can fit the data so well suggests the bias correction is reasonable and highly effective.

We note that on WIKIPEDIA, we were only able to measure the two-point mutual information up to $d = 256$, due to limited long-context length data in WIKIPEDIA.

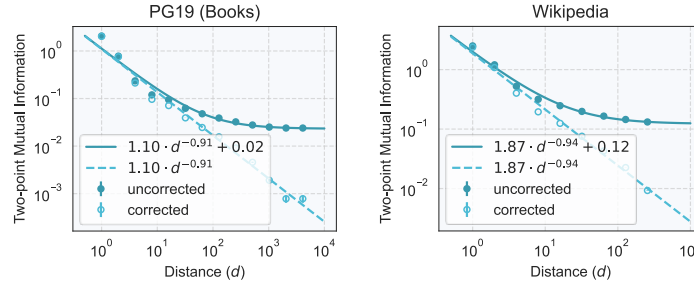


Figure B.5: Effect of bias correction for two-point mutual information. The bias causes a plateau at large distances.

B.VIII Failures of Two-point Mutual Information

In this section, we further demonstrate the relation between two-point and bipartite mutual information, and why two-point mutual information cannot properly capture the full multi-token dependencies needed for modeling natural language.

In existing literature, the scaling of two-point mutual information has been used to demonstrate the existence of long-range dependencies in natural language. In particular, it is believed that short-range dependencies are characterized by an exponential decay in two-point mutual information, as seen typically in finite-state Markov chains, and the existence of power-law decay in two-point mutual information in natural language indicates non-trivial long-range dependencies. Although this perspective is correct if one defines the existence of long-range dependence as the existence of non-exponential-decay two-point mutual information, this definition does not properly account for the mutual information between token pairs when other tokens are present. This can be made more clear by considering the following decomposition of bipartite mutual information.

For a sequence of tokens $W_{1:L}$ with $X_{1:\ell} = W_{1:\ell}$ and $Y_{1:L-\ell} = W_{\ell+1:L}$, the bipartite mutual information reads

$$I_{\ell;L}^{\text{BP}} = I(X_{1:\ell}; Y_{1:L-\ell}). \quad (\text{B.16})$$

Standard information theory allows mutual information to be decomposed as

$$I(XZ; Y) = I(X; Y) + I(Z; Y|X), \quad (\text{B.17})$$

where $I(Z; Y|X)$ is the conditional mutual information between Z and Y given X . Using this relation repeatedly, the bipartite mutual information can be decomposed as

$$\begin{aligned} I_{\ell;L}^{\text{BP}} &= I(X_{1:\ell}; Y_{1:L-\ell}) \\ &= I(X_1; Y_1) + I(X_2; Y_1|X_1) + I(X_1; Y_2|Y_1) + I(X_2; Y_2|X_1 Y_1) + \dots \\ &= \sum_{i=1}^{\ell} \sum_{j=1}^{L-\ell} I(X_i; Y_j | X_{1:i-1} Y_{1:j-1}) \\ &\neq \sum_{i=1}^{\ell} \sum_{j=1}^{L-\ell} I(X_i; Y_j) = \sum_{i=1}^{\ell} \sum_{j=1}^{L-\ell} I_{j-i+\ell}^{\text{TP}}. \end{aligned} \quad (\text{B.18})$$

In fact, it is in general not even clear whether the conditional mutual information $I(X_i; Y_j | X_{1:i-1} Y_{1:j-1})$ is greater or less than the marginal mutual information $I(X_i; Y_j)$. Therefore, as demonstrated here, when considering dependencies between blocks of text, a simple aggregation of the two-point mutual information gives a very incomplete picture. Due to this reason, weakly correlated systems, such as the example mentioned in Sec. 4.4 could exhibit seeming strong long-range two-point dependencies, and systems with very different bipartite mutual information, could share very similar two-point information scaling, as we will show later in Appx. C.

C Multivariate Gaussian Distributions

In the main text, we considered two families of multivariate Gaussian distributions of different sequence lengths to demonstrate the distinction between bipartite and two-point mutual information scalings. In particular, one is designed to mimic natural language, both in terms of the sub-volume law growth of the bipartite mutual information and the power-law decay of two-point mutual information. This family of distributions is also used to empirically verify our theory on L^2M condition for different LLM architectures. The other is designed to have the same two-point mutual information scaling, but very different bipartite mutual information scaling, showcasing that one can have distributions with the same two-point mutual information scaling, but drastically different bipartite mutual information scalings.

C.I Mutual Information Scalings

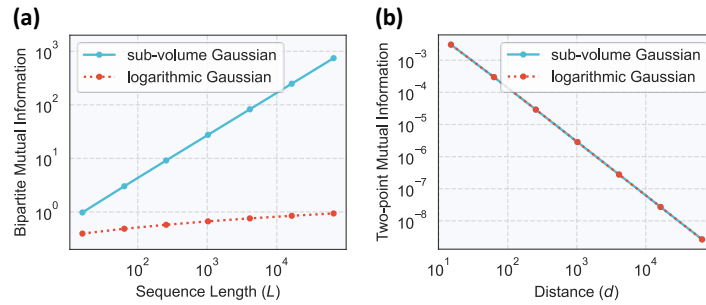


Figure C.6: Bipartite and two-point mutual information of the two families of Gaussian distributions.

Before showing the construction details, we first present both the bipartite and two-point mutual information scalings of the two families of Gaussian distributions. As shown in Fig. C.6, they have drastically different bipartite mutual information scaling—one has a power-law relation and the other logarithmic—but their antipodal two-point mutual information scaling is identical. This further demonstrates that two-point mutual information alone gives incomplete information of the multi-token long-range dependencies present in sequence data, and the simple aggregation of two-point mutual information does not tell the full picture of the bipartite mutual information. In fact, one can construct distributions with the same power-law decay in two-point mutual information, but has a constant bipartite mutual information scaling as well.

C.II Construction

Let's start by considering the family of distributions with sub-volume law growth. The distributions are constructed in a hierarchical manner.

We start at the first layer, with four independent standard Gaussian random variables (X_1, X_2, X_3, X_4) . Then, define the change-of-coordinate matrix

$$\mathcal{M} = \begin{pmatrix} \gamma & \gamma & \gamma & \rho \\ -\gamma & \gamma & -\gamma & \rho \\ -\gamma & -\gamma & \gamma & \rho \\ \gamma & -\gamma & -\gamma & \rho \end{pmatrix}, \quad (\text{C.19})$$

where we choose $\gamma = \sqrt{5}/4$ and $\rho = 1/4$. The output of the first layer is defined as

$$\mathbf{Y} = \mathcal{M}\mathbf{X}, \quad (\text{C.20})$$

where the random variables are now correlated. It is easy to verify that this operation only changes the off-diagonal elements in the covariance matrix, and leaves the diagonal elements unaffected.

For the second layer and up, we first stack three independently sampled copies from the previous layer and attach additional independent standard Gaussian random variables as the fourth elements as

$$\mathcal{X} = \begin{pmatrix} Y_{1,1} & Y_{1,2} & Y_{1,3} & W_1 \\ Y_{2,1} & Y_{2,2} & Y_{2,3} & W_2 \\ Y_{3,1} & Y_{3,2} & Y_{3,3} & W_3 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \quad (\text{C.21})$$

where $Y_{i,j}$ refers to the i th output from previous layer of the j th copy, and W_i refers to the i th newly sampled standard Gaussian random variable.

Note that at this point, all rows are independent from each other; therefore we apply the change of coordinate matrix $\mathcal{M}$ at each row to correlate them. The matrix is then flattened to obtain $\mathbf{Z}$

$$\mathbf{Z} = (Z_{1,1}, Z_{1,2}, Z_{1,3}, Z_{1,4}, Z_{2,1}, Z_{2,2}, Z_{2,3}, Z_{2,4}, Z_{3,1}, Z_{3,2}, Z_{3,3}, Z_{3,4}, \dots), \quad (\text{C.22})$$

where the subscripts denote the variables' original position in the matrix. Before outputting from this layer, we perform an addition operation to each pair of random variables $(Z_{i,4}, Z_{i+1,1})$, by applying a coordinate transformation that modifies their correlations as

$$\text{corr}(Z_{i,4}, Z_{i+1,1}) \rightarrow \frac{2}{5}(\text{corr}(Z_{i,3}, Z_{i,4}) + \text{corr}(Z_{i+1,1}, Z_{i+1,2})) + \frac{1}{5}. \quad (\text{C.23})$$

This operation may seem arbitrary, but it is crucial to introduce correlations that give a linear ordering of the random variables. Without this operation, the distribution simply forms a tree structure.

Now, we can truncate the construction at different layers l and form a family of distributions with different sequence lengths $L = 4^l$. In Fig. C.6 of the main text, we consider up to 8 layers, and in Fig. 3 and 4, we consider $l = 4, 5$ and, 6.

The second family of distributions is constructed analogously. The only difference is that we replace Eq. (C.21) with a single copy of $\mathbf{Y}$ and three independent copies of $\mathbf{W}$ as

$$\mathcal{X} = \begin{pmatrix} Y_1 & W_{1,1} & W_{1,2} & W_{1,3} \\ Y_2 & W_{2,1} & W_{2,2} & W_{2,3} \\ Y_3 & W_{3,1} & W_{3,2} & W_{3,3} \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \quad (\text{C.24})$$

C.III Properties

These two series of distributions have many nice properties, in addition to their bipartite and two-point mutual information scalings. In addition, these constructions directly defines the multi-variate probability distribution due to their Gaussian nature. This allows for exact calculations of conditional probability distributions for training LLMs, as well as direct computation of the bipartite and two-point mutual information without LLM approximations.

D Model State for Storing Past Information

In Definition 5.1 in the main text, we give a concrete definition of “model state for storing past information” as history state, and claim that it is the past key-value pairs for transformers and recurrent state for SSMs and RNNs. Here, we explain them in more detail.

D.I Transformers

In transformers, only the attention block mixes information among different tokens, therefore we only need to analyze the behavior of the attention block. We will be assuming the existence of the causal mask, as our theory mainly applied to autoregressive LLMs. Denoting the input and output of the attention layer as $\mathbf{x}$ and $\mathbf{y}$ (notice they are no longer two parts of a sequence), the self-attention mechanism is defined as

$$\mathbf{y} = \text{softmax}((W_q \mathbf{x})(W_k \mathbf{x})^T)W_v \mathbf{x}, \quad (\text{D.25})$$

where W_q , W_k and W_v are the weight matrices. For simplicity, we drop the usual $\sqrt{h_{\text{dim}}}$ normalization and the output weight matrix, as they are irrelevant to our discussion.

Separating the calculation for each token, the mechanism can be rewritten as

$$y_i = \frac{\sum_{j=1}^i e^{(W_q x_i)(W_k x_j)} W_v x_j}{\sum_{j'=1}^i e^{(W_q x_i)(W_k x_{j'})}} = \frac{e^{(W_q x_i)(W_k x_i)} W_v x_i + \sum_{j=1}^{i-1} e^{(W_q x_i)k_j} v_j}{e^{(W_q x_i)(W_k x_i)} + \sum_{j'=1}^{i-1} e^{(W_q x_i)k_{j'}}}, \quad (\text{D.26})$$

where $k_j = W_k x_j$ and $v_j = W_v x_j$ are keys and values which we sum over past tokens. Clearly, the attention output only depends on the current token x_i and the past key-value pairs $k_{1:i-1}$ and $v_{1:i-1}$. This argument extends to all y_k with $k \geq i$, where all y_k 's dependency on $x_{1:i-1}$ is via $k_{1:i-1}$ and $v_{1:i-1}$. Therefore, key-value pairs form the history state, and their size grows linearly with input sequence length. We note that Eq. (D.26) also describes how key-value caching works.

D.II State Space Models and RNNs

State space models (SSMs) and RNNs, on the other hand, are easier to analyze. These models in general all have some recurrent state with a fixed size, and some mechanism to update the state when a new token is observed. The output depends only on the previous recurrent state, and the current token. They can in general be written in the following way.

$$\begin{aligned} h_i &= f(h_{i-1}, x_i), \\ y_i &= g(h_{i-1}, x_i), \end{aligned} \quad (\text{D.27})$$

for some update function f and output function g . It is obvious that the history state is exactly this recurrent state (or the collection of recurrent states of different layers), which does not grow with the input sequence.

We note that this discussion also applies to linear attention models, whose key-value pairs can be merged into a recurrent state with fixed size, due to the replacement of softmax function. Test time training (TTT) models can also be included in this discussion. They can be viewed as RNNs with inner model parameters as recurrent state, and test time training process as update function.

D.III Other Architectures

For other models, such as sparse transformers or some compression-based models, the analysis has to be performed separately. Nevertheless, the L^2M framework is general: after identifying the history state, one can always compare its scaling with the bipartite mutual information scaling to see whether the model is capable of capturing the long-range dependencies in the data.

E Proofs of Theorem 5.2

We provide three proofs of Theorem 5.2 under different assumptions, demonstrating the universality and robustness of the result. Importantly, all three sets of assumptions are extremely mild and directly reflect realistic conditions in modern neural networks, whether through the discrete nature of floating-point arithmetic, empirically observed geometric properties of neural representations, or basic continuity requirements.

E.I Proof Under Discreteness Assumption

The discreteness assumption is already quite reasonable in practice. Modern neural networks use floating-point representations, which are inherently discrete with finite precision. Moreover, neural networks have been shown to retain strong performance even under aggressive quantization, demonstrating that discrete representations with limited precision are sufficient to capture the essential information. This discreteness assumption thus provides a natural and practical starting point for our proof.

Theorem E.1 (Theorem 5.2, Discrete Version). *Assume the history state z_ℓ takes discrete values. Then a model's capacity to capture bipartite mutual information is bounded by the size of its history state as*

$$I_{\ell;L}^{\text{BP},q} \leq C \cdot \dim(z_\ell) + \log(M) \quad (\text{E.28})$$

where C is some constant and M denotes the vocabulary size.

Proof. By the data processing inequality: $I^q(X_{1:\ell}; Y_{1:L-\ell}) \leq I^q(\mathbf{Z}_\ell X_\ell; Y_{1:L-\ell})$. This is upper bounded by the entropy: $H^q(\mathbf{Z}_\ell X_\ell)$, which is further upper bounded by $H^q(\mathbf{Z}_\ell) + H^q(X_\ell) \leq C \cdot \dim(\mathbf{z}_\ell) + \log(M)$, where the last inequality follows from the bound on entropies of discrete variables. $\square$

E.II Proof Under Almost Orthogonal Directions (AOD) Assumption

The discreteness assumption can be relaxed if we instead assume the following observed fact about neural networks: neural networks store distinct information in almost orthogonal directions (AODs) of the hidden state [105–107].

Theorem E.2 (Theorem 5.2, AOD Version). *Assume neural networks store distinct information in almost orthogonal directions (AODs) of the hidden state. Then a model’s capacity to capture bipartite mutual information is bounded as*

$$I_{\ell;L}^{\text{BP},q} \leq C \cdot \dim(\mathbf{z}_\ell) + \log(M) \quad (\text{E.29})$$

where C is some constant and M denotes the vocabulary size.

Proof. An autoregressive neural network’s dependency on past tokens is through the intermediate variable $\mathbf{z}_\ell = \mathbf{f}(x_{1:\ell-1})$ such that $q(\mathbf{y}|\mathbf{x}) := q(\mathbf{y}|x_\ell, \mathbf{z}_\ell)$. This can be viewed as the process $\mathbf{X} \rightarrow (\mathbf{Z}_\ell, X_\ell) \rightarrow \mathbf{Y}$. According to the data processing inequality,

$$I^q(X_{1:\ell}; Y_{1:L-\ell}) \leq I^q(\mathbf{Z}_\ell, X_\ell; Y_{1:L-\ell}) \leq \mathcal{H}^q(\mathbf{Z}_\ell, X_\ell) \leq \mathcal{H}^q(\mathbf{Z}_\ell) + H^q(X_\ell), \quad (\text{E.30})$$

where we use $\mathcal{H}$ to denote a generalized notion of entropy which measures the amount of information that can be stored in $\mathbf{Z}_\ell$. We only care about the scaling of $\mathcal{H}$, so its exact definition is irrelevant to our discussion.

Under the AOD assumption, neural networks store distinct information in almost orthogonal directions of the hidden state. Therefore, $\mathcal{H}$ should scale at most logarithmically with respect to the number of AODs as the state size increases. According to the Kabatjanskii–Levenstein bound [108, 109], given an error tolerance ε , the number of AODs is upper bounded by $\exp(f(\varepsilon) \cdot \dim(\mathbf{z}_\ell))$ for some function f that depends purely on the error threshold. Therefore, the generalized entropy scales as $\mathcal{H}^q(\mathbf{Z}_\ell) \lesssim \log \exp(f(\varepsilon) \cdot \dim(\mathbf{z}_\ell)) \sim \dim(\mathbf{z}_\ell)$. Since $H^q(X_\ell) \leq \log(M)$ where M is the vocabulary size, we conclude

$$I_{\ell;L}^{\text{BP},q} \leq C \cdot \dim(\mathbf{z}_\ell) + \log(M). \quad (\text{E.31})$$

$\square$

E.III Proof Under Lipschitz Continuity Assumption

The theorem can also be proved assuming only certain Lipschitz continuity conditions on the neural network.

Theorem E.3 (Theorem 5.2, Lipschitz Version). *Assume the history state mapping $\mathbf{f} : x_{1:\ell-1} \mapsto \mathbf{z}_\ell$ satisfies $\|\mathbf{f}(x_{1:\ell-1}) - \mathbf{f}(x'_{1:\ell-1})\|_2 \leq K_f \mathbb{1}(x_{1:\ell-1} \neq x'_{1:\ell-1})$ and the neural network is entropy-Lipschitz, satisfying $|H^q(\mathbf{Y}|\mathbf{z}_\ell) - H^q(\mathbf{Y}|\mathbf{z}'_\ell)| \leq K_H \|\mathbf{z}_\ell - \mathbf{z}'_\ell\|_2$. Then a model’s capacity to capture bipartite mutual information is bounded as*

$$I_{\ell;L}^{\text{BP},q} \leq C \cdot \dim(\mathbf{z}_\ell) + \log(M) \quad (\text{E.32})$$

where C is some constant and M denotes the vocabulary size.

Proof. We start with the data processing inequality and rewrite the bound as

$$\begin{aligned} I^q(X_{1:\ell}; Y_{1:L-\ell}) &\leq I^q(\mathbf{Z}_\ell, X_\ell; Y_{1:L-\ell}) \\ &= I^q(\mathbf{Z}_\ell; Y_{1:L-\ell}) + I^q(X_\ell; Y_{1:L-\ell}|\mathbf{Z}_\ell) \\ &\leq I^q(\mathbf{Z}_\ell; Y_{1:L-\ell}) + \log(M), \end{aligned} \quad (\text{E.33})$$

where the last inequality uses $I^q(X_\ell; Y_{1:L-\ell}|\mathbf{Z}_\ell) \leq H^q(X_\ell) \leq \log(M)$, with M being the vocabulary size.

The history state is a function of the input tokens $\mathbf{z}_\ell = \mathbf{f}(x_{1:\ell-1})$, with $x_{1:\ell-1} \in \{1, 2, \dots, M\}^{\ell-1}$. Under our assumption on $\mathbf{f}$, $\mathbf{z}_\ell$ lives in a d -dimensional ball of radius K_f , where $d = \dim(\mathbf{z}_\ell)$.

Consider a quantization $\mathbf{Q}(\mathbf{z}_\ell)$ that maps each $\mathbf{z}_\ell$ to the nearest point in an ε -covering of this ball. Then

$$I^q(\mathbf{Z}_\ell; \mathbf{Y}) = I^q(\mathbf{Q}(\mathbf{Z}_\ell); \mathbf{Y}) + H^q(\mathbf{Y}|\mathbf{Q}(\mathbf{Z}_\ell)) - H^q(\mathbf{Y}|\mathbf{Z}_\ell). \quad (\text{E.34})$$

By the entropy-Lipschitz assumption, $H^q(\mathbf{Y}|\mathbf{Q}(\mathbf{Z}_\ell)) - H^q(\mathbf{Y}|\mathbf{Z}_\ell) \leq K_H \varepsilon$. Since $\mathbf{Q}(\mathbf{Z}_\ell)$ is discrete and takes at most $(2K_f/\varepsilon)^d$ values (by a covering number argument), we have $I^q(\mathbf{Q}(\mathbf{Z}_\ell); \mathbf{Y}) \leq H^q(\mathbf{Q}(\mathbf{Z}_\ell)) \leq d \log(2K_f/\varepsilon)$.

Therefore, $I^q(\mathbf{Z}_\ell; \mathbf{Y}) \leq d \log(2K_f/\varepsilon) + K_H \varepsilon \leq C \cdot d$ for some constant C (by choosing ε appropriately). This concludes

$$I_{\ell;L}^{\text{BP},q} \leq C \cdot d + \log(M) = C \cdot \dim(\mathbf{z}_\ell) + \log(M). \quad (\text{E.35})$$

□

E.IV Discussion

We believe this theorem is more universal and can be proved in additional ways, such as by connecting it to channel capacity and potentially showing $I_{\ell;L}^{\text{BP},q} \leq d \log(1 + \text{SNR}) + \log(M)$. We also believe the theorem can be established with more relaxed assumptions, similar to how information dimension is proved to be the upper bound of lossless compression of continuous random variables [110]. However, additional proofs are beyond the scope of this work, and the three proofs provided should already be applicable in any practical settings.

F Additional Details on Experimental Setup

In this section, we provide detailed information about our experimental setup, including dataset construction, model configurations, training procedures, and evaluation metrics.

F.I Synthetic Gaussian Distribution Dataset

For experiments on the multivariate Gaussian distribution, we use the sub-volume Gaussian distributions described in Appx. C, which exhibits power-law bipartite mutual information scaling with an exponent of 0.79. To fully stress the LLMs, we stack 64 copies of the distribution and group the 64 Gaussian variables at each position to form a single token. More specifically, an example sample looks like

$$\begin{aligned} \mathbf{W} &= (W_1, W_2, \dots, W_L) \\ &:= ((Z_{1,1}, Z_{1,2}, \dots, Z_{1,64}), (Z_{2,1}, Z_{2,2}, \dots, Z_{2,64}), \dots, (Z_{L,1}, Z_{L,2}, \dots, Z_{L,64})), \end{aligned} \quad (\text{F.36})$$

where the two subscripts (i, j) refer to the i th random variable from the j th copy. In this way, the bipartite mutual information matches better with natural language, not only in scaling, but also in magnitude (multiplicative constant). We additionally prepend an all-zero token W_0 to each sample to mimic the effect of the BOS token.

In order to process continuous random variables, we replace the embedding layers of GPT2 and Mamba(2) models with two-layer MLPs. For output, since all the conditional distributions are also Gaussian, we use a different two-layer MLP to output the 64 conditional means $\mu_{q_{Z_{i,j}}|Z_{0:i-1,j}}$ and standard deviations $\sigma_{q_{Z_{i,j}}|Z_{0:i-1,j}}$.

As discussed in Appx. C.III, due to the analytical construction, the Gaussian distribution permits efficient calculation of conditional probabilities. Therefore, instead of simply training the neural networks with negative log likelihood on samples alone, we use the average conditional KL-divergence estimated as

$$\begin{aligned} D_{KL}(p||q_\theta) &= \\ \mathbb{E}_{p_Z} \left[\frac{1}{L} \sum_{i=1}^L \frac{1}{64} \sum_{j=1}^{64} \left(\log \frac{\sigma_{q_{Z_{i,j}}|Z_{0:i-1,j}}}{\sigma_{p_{Z_{i,j}}|Z_{0:i-1,j}}} + \frac{\sigma_{p_{Z_{i,j}}|Z_{0:i-1,j}}^2 + (\mu_{q_{Z_{i,j}}|Z_{0:i-1,j}} - \mu_{p_{Z_{i,j}}|Z_{0:i-1,j}})^2}{2\sigma_{q_{Z_{i,j}}|Z_{0:i-1,j}}^2} - \frac{1}{2} \right) \right] \end{aligned} \quad (\text{F.37})$$

to reduce sampling variance.

F.II Natural Language Dataset (PG19)

For the PG19 dataset, we train on standard average negative log likelihood. We first split the dataset into samples with a length of approximately 1.2 times the target length, ensuring each sample starts at the beginning of a sentence. We then train the models for 5 epochs (approximately 450k iterations) with a batch size of 16,384 tokens. To maintain consistency across different models, we always use the same tokenizer from GPT-Neo-X [111].

F.III Training Configuration

For the Gaussian distribution training, during each iteration, we use a batch size of 4 (4 times sequence length number of tokens) with freshly generated samples, meaning we never reuse any sample. We therefore have effectively a single epoch, thanks to the “infinite” dataset size. We train all neural networks using the AdamW optimizer and a cosine decay scheduler with warmup. We use a peak learning rate of 5×10^{-5} , a weight decay of 0.01, 2000 warmup steps, and 500,000 training steps in total. The results reported are at the end of training.

For the PG19 dataset experiments, we use similar hyperparameters: AdamW optimizer with a cosine decay scheduler with warmup, peak learning rate of 5×10^{-5} , weight decay of 0.01, 2000 warmup steps, and 500,000 steps in total. The results reported are at the end of training using a separate evaluation dataset containing 10,000 samples.

F.IV Evaluation Metrics

In this paper, we report results on the position-wise conditional KL-divergence

$$D_{KL,i} = D_{KL}(p_{W_i|W_{1:i-1}} || q_{W_i|W_{1:i-1}}) = \mathbb{E}_p [\log p(W_i|W_{1:i-1}) - \log q(W_i|W_{1:i-1})], \quad (\text{F.38})$$

average KL-divergence

$$D_{KL}^{\text{avg}} = \frac{1}{L} \sum_{i=1}^L D_{KL,i}, \quad (\text{F.39})$$

and position-wise conditional NLL

$$\text{NLL}_i = -\mathbb{E}_p [\log q(W_i|W_{1:i-1})]. \quad (\text{F.40})$$

One can also define an average NLL as

$$\text{NLL}^{\text{avg}} = \frac{1}{L} \sum_{i=1}^L \text{NLL}_i, \quad (\text{F.41})$$

which we use in Appx. G.

F.IV.1 Understanding the Behavior of KL-divergence and NLL

It is important to understand how KL-divergence and NLL behave differently as token position increases. Using the relationship between cross-entropy and KL-divergence from Eq. (5), we can decompose the conditional NLL as

$$\text{NLL}_i = D_{KL,i} + H^p(W_i|W_{1:i-1}), \quad (\text{F.42})$$

where $H^p(W_i|W_{1:i-1}) = -\mathbb{E}_p [\log p(W_i|W_{1:i-1})]$ is the conditional entropy of the true distribution at position i .

This decomposition reveals why KL-divergence and NLL exhibit opposite trends with token position. As token position increases, the conditional entropy $H^p(W_i|W_{1:i-1})$ typically decreases because more context is available, making the next token more predictable. In natural language, this reflects that with more preceding text, there is less uncertainty about what comes next. Meanwhile, the conditional KL-divergence $D_{KL,i}$ often increases with position, because learning all long-range dependencies becomes more challenging, resulting in worse model estimations compared to the true conditional distribution.

For models with sufficient capacity relative to sequence length, the decrease in conditional entropy dominates, causing NLL to decrease with position despite increasing KL-divergence. However, for models with insufficient capacity (such as fixed-state models at long sequence lengths), the KL-divergence can increase rapidly enough that NLL plateaus or even increases at later positions. This behavior is precisely what we observe in our experiments with Mamba models in Fig. 4, where Mamba’s NLL plateaus at later positions while transformers’ NLL continues to improve.

The same reasoning applies to average quantities: NLL^{avg} typically decreases with sequence length L as the average conditional entropy decreases, while D_{KL}^{avg} may increase if the model’s capacity becomes insufficient relative to the growing sequence length.

When reporting conditional NLL, we smooth the curves using a small window around nearby tokens to reduce noise in the results.

F.V Model Configurations

In Tables F.1 and F.2, we include the model configurations and sequence lengths for all experiments performed in this paper.

Table F.1: Models and configurations for synthetic dataset experiments.

Model	num_hidden_layers	hidden_size	seq_len
GPT2	12	768	256,1024,4096
GPT2-medium	24	1024	256,1024,4096
GPT2-large	36	1280	256,1024,4096
Mamba-50m	12	512	64,256,1024,4096,16384
Mamba-70m	24	512	64,256,1024,4096,16384
Mamba-130m	24	768	64,256,1024,4096,16384
Mamba-370m	48	1024	64,256,1024,4096
Mamba-790m	48	1536	64,256,1024,4096
Mamba-1.4b	48	2048	64,256,1024,4096
Mamba2-130m	24	768	256,1024,4096
Mamba2-370m	48	1024	256,1024,4096
Mamba2-790m	48	1536	256,1024,4096

Table F.2: Models and configurations for PG19 experiments.

Model	num_hidden_layers	hidden_size	seq_len
GPT2	12	768	4096,8192
GPT2-medium	24	1024	4096,8192
GPT2-large	36	1280	4096,8192
Mamba-130m	24	768	4096,8192
Mamba-370m	48	1024	4096,8192
Mamba-790m	48	1536	4096,8192

F.VI Computational Resources and Implementation Details

Our experiments are performed primarily on H100 GPUs, with varying VRAM sizes between 80GB and 96GB. Some experiments use A100 GPUs with 80GB VRAM instead. We use the vLLM library [68] when running inference to estimate the mutual information scaling. For both LLaMA 3.1 405B and DeepSeek V3, we run the FP8 version using 8 H100 GPUs (with 96GB VRAM each). The model weights and configurations are downloaded from HuggingFace [112].

When training GPT and Mamba(2) models on the Gaussian distribution, we use our custom library developed in PyTorch [113]. When training GPT and Mamba models on the PG19 dataset, we use the trainer from the HuggingFace transformers library. All models are initialized from scratch, with model configurations taken from HuggingFace. All training experiments are performed on individual H100 and A100 GPUs with FP32 precision to avoid possible training failures. Although

training with FP16 would make the experiments run faster, it should not affect the actual results. We note that for Mamba2, we use the official implementation instead of the HuggingFace version. For the GPT2 experiments on PG19, we re-implement the attention mechanism with FlexAttention [114] to save memory, as the official FlashAttention [64–66] does not support FP32 precision.

F.VII Code Availability

The code for reproducing our mutual information estimation and the PG19 results is available at https://github.com/LSquaredM/mutual_info_scaling_law.

G Additional Experimental Results

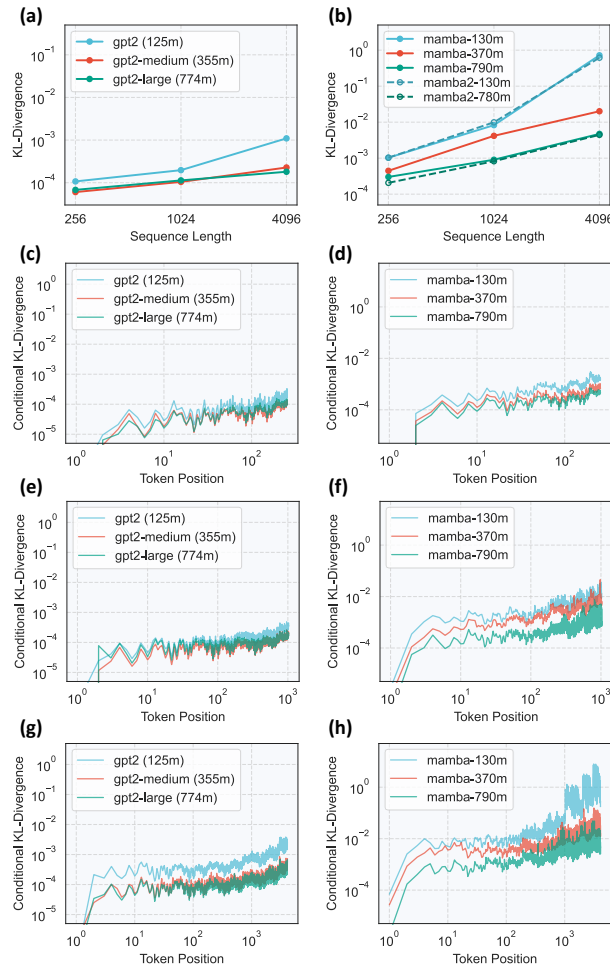


Figure G.7: Evaluation of KL-divergence across model architectures trained on sub-volume Gaussian distributions. (a, b) Average KL-divergence per token for models trained on different sequence lengths [same as Fig. 3 (a, b)]. (c, d) Position-wise conditional KL-divergence for models trained on sequence length 256. (e, f) Position-wise conditional KL-divergence for models trained on sequence length 1024. (g, h) Position-wise conditional KL-divergence for models trained on sequence length 4096 [same as Fig. 3 (c, d)]. Lower values indicate better performance.

In this section, we show additional experimental results. In Fig. G.7, we include positional-wise conditional KL-divergences of models trained on sub-volume Gaussian distributions with sequence length 256 (c, d), 1024 (e, f), and 4096 (g, h). As clearly demonstrated in the figure, for short sequence lengths, Mamba maintains similar performances to GPT2; Mamba models of different sizes also appear to have a smaller performance gap. However, as we go to longer sequence lengths,

smaller Mamba models starts to fail, while GPT2 always maintain relatively stable performances, consistent with our theory.

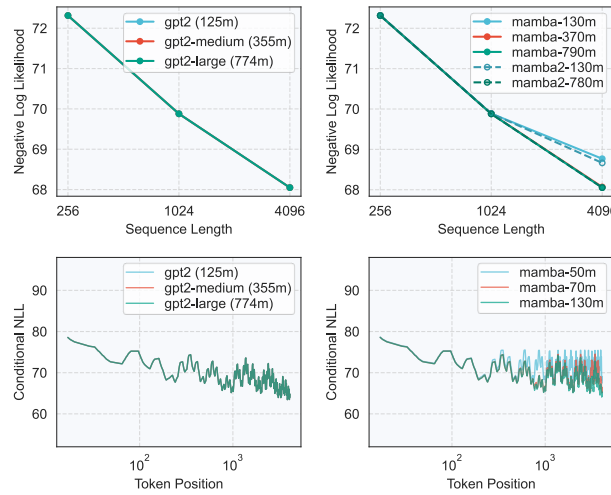


Figure G.8: Negative log likelihood (NLL) across model architectures trained on sub-volume Gaussian distributions (a, b) Average NLL per token for models trained on different sequence lengths. (c, d) Position-wise conditional NLL for models trained on sequence length 4096. Lower values indicate better performance.

In Fig. G.8, we show the negative log likelihood (NLL) of models trained on sub-volume Gaussian distributions. We note that, because NLL combines the KL-divergence with the intrinsic entropy of the underlying distribution (the average and position-wise conditional of which decays as sequence lengths), the differences between model performances are less visible. It's worth noting that, since Gaussian random variables are continuous, NLL values can differ by an arbitrary additive constant by rescaling the distribution. Therefore, the exact values of conditional NLL do not carry intrinsic meaning, though relative comparisons (which is exactly the same as the KL-divergence) between models remain valid.

In Fig. G.9, we show the position-wise conditional negative log likelihood (NLL) of models trained on the PG19 dataset [90] with 4096-token sequences. The results here is consistent with the 8192-token-sequence results in the main text.

H Limitations

Our theoretical framework specifically examines models' capacity to capture long-range dependencies through the lens of bipartite mutual information and does not address other aspects of language modeling, such as reasoning capabilities or world knowledge. The L^2M condition establishes necessary but not sufficient conditions for effective long-context modeling. Understanding how this theoretical capacity translates to actual downstream task performance remains an important open question. The relationship likely depends on additional factors including optimization dynamics, architectural inductive biases, and task-specific requirements. Systematic evaluation across diverse long-context benchmarks represents a crucial next step to clarify these relationships and identify any gaps between theoretical capability and practical performance.

While our empirical validation on synthetic Gaussian distributions with controlled mutual information scaling provides clean verification of the theoretical predictions, it may not capture all complexities present in natural language. Our theory focuses on autoregressive language models, which remains broadly applicable as even diffusion-based approaches typically employ autoregressive generation for extended sequences in practice. Nonetheless, exploring whether similar information-theoretic principles govern fundamentally different generative paradigms or multimodal models represents an interesting direction for future work.

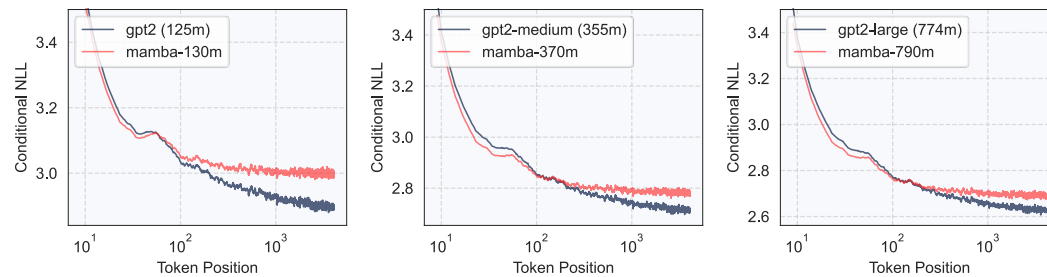


Figure G.9: Position-wise conditional negative log likelihood (NLL) evaluation for models trained on 4096-token sequences on the PG19 dataset [90].

The methodology we employ, using LLMs as density estimators for mutual information measurement, represents a practical approach given the severe challenges of high-dimensional estimation in long sequences. Alternative methods like K-NN and neural estimators face fundamental difficulties with dimensionality and sequence length. While our approach yields consistent power-law behavior across different models and estimators, both methods likely underestimate the true exponent, and developing more accurate estimation techniques remains an important challenge.

Our evaluations rely primarily on open-source models; further verification using state-of-the-art closed-source models would provide additional validation.

I Broader Impact

This work advances our theoretical understanding of how language models process long-range dependencies, with implications for the design and deployment of more efficient LLM architectures. By establishing the L^2M condition, we provide a principled framework for evaluating an architecture's fundamental capacity for long-context modeling. This could lead to more efficient models that maintain effectiveness while reducing computational resources, potentially decreasing the environmental footprint of training and inference. Our findings may influence the development of specialized architectures for tasks requiring long-context understanding, such as legal document analysis, scientific research, and complex reasoning.

However, improved long-context modeling could also amplify existing challenges in LLMs, including the propagation of bias over longer contexts and enhanced capabilities for generating persuasive misinformation. Research applying the L^2M framework should consider these ethical dimensions, particularly how improvements in long-range dependency modeling might affect model safety, fairness, and the verifiability of model outputs.

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: All claims are supported by either theoretical or experimental results.

Guidelines:

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

2. Limitations

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

Answer: [Yes] .

Justification: The limitations are discussed in Appx. H.

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [Yes] .

Justification: All the proofs are included in either the main text or in Appendix, together with the necessary assumptions.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [Yes] .

Justification: All the experimental details are described in the appendix, with main code provided at https://github.com/LSquaredM/mutual_info_scaling_law.

Guidelines:

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

5. Open access to data and code

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

Answer: [Yes] .

Justification: The main code is available at https://github.com/LSquaredM/mutual_info_scaling_law.

Guidelines:

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

6. Experimental setting/details

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

Answer: [Yes] .

Justification: All the details are described in the appendix.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [No] .

Justification: For the measurement of mutual information scaling, the error bar is too small to see on the figure, so we omit it when drawing the figures. For the experimental results, it is too costly to train multiple neural networks for each setup. Nevertheless, our experimental results are very clean and consistent with our theory predictions.

Guidelines:

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

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

8. Experiments compute resources

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

Answer: [Yes] .

Justification: All the compute resources used are described in the appendix.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes] .

Justification: Everything conforms with the Code of Ethics.

Guidelines:

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

10. Broader impacts

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

Answer: [Yes] .

Justification: The broader impact is discussed in Appx. I.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

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

Answer: [NA] .

Justification: Not applicable.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes] .

Justification: We cite all the datasets and models used in the paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [\[Yes\]](#) .

Justification: The main code used in the paper is posted at https://github.com/LSquaredM/mutual_info_scaling_law. The synthetic dataset generation process is fully described in the appendix.

Guidelines:

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

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

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

Answer: [\[NA\]](#) .

Justification: Not applicable.

Guidelines:

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

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

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

Answer: [\[NA\]](#) .

Justification: Not applicable.

Guidelines:

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

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

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

Answer: [NA] .

Justification: Not applicable.

Guidelines:

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