A Appendix

A.1 Related Work

Language models and compression. Shannon's source coding theorem (Shannon, 1948) first formalized the duality between prediction and compression. The connection between language modeling and compression was studied as far back as Shannon (1950), which observed that more accurate models of English can compress text in fewer bits. Other works note the connection between Kolmogorov complexity (Kolmogorov, 1965) and Shannon information in detail (Grunwald & Vitanyi, 2004). Delétang et al. (2024) investigate using modern transformer-based language models as compressors. We use compression as a tool to measure memorization in models.

Language model capacity. (Arpit et al., 2017) formalize the idea of *effective capacity* of a model and its training procedure; they also observe that both representation capacity and training time have a strong impact on empirical model capacity. Several other works measure language model capacity in the number of facts or random labels that can be memorized by a network such as an RNN (Collins et al., 2017; Boo et al., 2019) or transformer (Roberts et al., 2020; Heinzerling & Inui, 2021; Allen-Zhu & Li, 2024), sometimes under quantization. A few research efforts (Yun et al., 2019; Curth et al., 2023; Mahdavi et al., 2024; Kajitsuka & Sato, 2024) have developed theoretical estimates for the capacity of different model architectures, although none have yet scaled to multi-layer modern transformers. We are the first to measure a clear upper-bound in model capacity.

Alternative definitions of memorization. Unintended memorization is deeply related to the many other definitions of memorization proposed in the literature. We provide a detailed comparison in and Section A.3.

Figure 8: Bits memorized across training. This particular model is a GPT-style transformer with $6.86M$ parameters and a capacity of 23.9 MB.

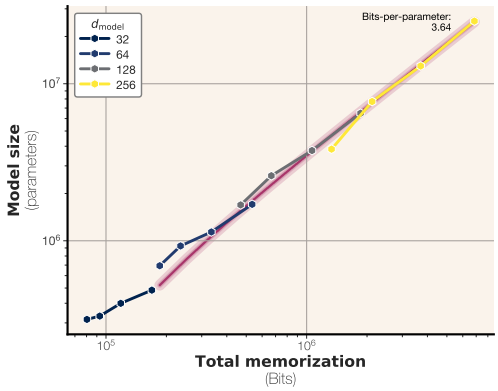

Figure 9: Capacity in bits-per-parameter for models trained on synthetic data. We estimate $\alpha = 3.64$ bits-per-parameter for GPT models trained in half precision.

A.2 Related Work: Definitions of Memorization

Prior definitions of memorization. Carlini et al. (2019) defined a string m as memorized by a language model θ if the second half of m can be generated greedily when prompting the model with the first half. Following this, Nasr et al. (2023) introduced *extractable memorization*, where model θ is said to memorize m if an adversarial prompt p can be found that generates m. Mireshghallah et al. (2022) and Schwarzschild et al. (2024) refined this definition by restricting p to a certain number of tokens, preventing it from containing the entire m. However, even this definition has limitations: for example, generating the sequence "cat cat cat ... cat" with the prompt "repeat cat 1000 times" does not necessarily indicate memorization. Carlini et al. (2019) use perplexity or likelihood, one measure of

the compressibility of a sequence, in an effort to distinguish highly memorized sequences from merely easy-to-compress ones. One additional definition of note is *counterfactual memorization* (Zhang et al., 2023), which measures the impact of a single datapoint on training; this can be seen as an instantiation of our definition where a different model of the same family is used as a reference model. Overall, all these works regarded memorization in terms that can be seen as forms of compression, although did not explicitly define it as such.

Finally, in an independent (but earlier) work of (Cohen et al., 2024), authors propose a theoretical definition for memorization also relying on Kolmogorov complexity. This notion is similar to us in using Kolmogorov complexity to measure information content but it does not separate unintended from intended memorization.

Some of our findings also relate to the discovery of *double descent* in machine learning (Belkin et al., 2019; Nakkiran et al., 2019) and language modeling (Xia et al., 2023), as well as general discussions of memorization and generalization in deep learning (Zhang et al., 2017; Tänzer et al., 2022).

Here, we discuss other definitions of memorization.

A.3 Other notions of memorization

In this section we list multiple other notions of memorization and compare it with our definition. We specifically focus on why these notions do not satisfy all of our requirements.

- **Stability-based notions of memorization.** There are notions of privacy and memorization that deal with "stability" of the training algorithm to small changes in the training set. Most notably, differential privacy Dwork (2006) considers the worst-cast drift of the model distribution when a single data point changes. Another notion of memorization in Feldman (2020) is based on the change of the model prediction on a point x, when we add the labeled pair (x, y) to the training set of a classification/regression model. Both of these notions are crucially relying on the learning algorithm and how it behaves. Moreover, the definition of differential privacy is not ideal for our case because it is a worst-case definition and cannot be applied at sample/model level. While the notion of memorization in Feldman (2020) does not have this particular issue, it suffers from the fact that it only applies to classification models and mostly deals with the memorization of the association between the label (y) and input (x), and not the memorization of x itself. These issues make these notions not ideal for our case.

- **Extraction-based memorization.** There are multiple works in the literature (Carlini et al., 2019; Mireshghallah et al., 2022; Nasr et al., 2023; Zhang et al., 2023; Carlini et al., 2023b; Schwarzschild et al., 2024) that define memorization of samples in language models based on how easy it is to extract that sample. Specifically, when trying to understand the extent of memorization of a sample x in a model θ they measure some notion of complexity for the task of eliciting the model to output x. Although these notions are great in that they only take a model θ and a sample x, they still do not account for generalization. Considering our running example of the following training sample: "What is 2^{100}? (A: $1,267,650,600,228,229,401,496,703,205,376$)", this will be identified as highly memorized by almost all of the extraction based notions of memorization. Another issue with these definitions are that they are heavily dependent on the details of decoding algorithm. This is not ideal as we do not expect the memorization of a sample x in a model θ to depend on the detailed parameters we use to generate samples using θ.

  The work of Schwarzschild et al. (2024) in this category is the closest to ours. This work which is based on prompt-optimization, optimizes a short prompt p to make the model elicit x, then it calls the sample x memorized, if length of p is less than x. Although this definition is close to our definition in using compression, it still does not account for generalization of the model. Moreover, it focuses on a specific way of compression through prompting. We posit that compression through prompting

is an inferior compression scheme and can often lead to compression rates greater than 1.

- **Membership/attribute inference.** Membership inference Shokri et al. (2017) and attribute inference attacks Jayaraman & Evans (2022) have been used for empirically measuring the privacy of machine learning algorithms. These notions which usually aim at approximating the stability notions of memorization are suffering from the same shortcomings. They rely heavily on the learning algorithm and the data distribution. Moreover, they fail at providing a sample level notion of memorization. For example, the obtained accuracy for membership inference attack is only meaningful in the population level. This is because various attack may have different true positives for membership, and the union of all these true positive across different attack may cover the entire training set, rendering it unusable as a sample level notion of memorization.
- **Data copying in generative models.** There are some interesting notions of memorization designed specifically for generative modeling where a generative model may output a certain portion of training samples (Bhattacharjee et al., 2023; Carlini et al., 2023a). These notions are similar to extraction based definition of memorization but they are more lenient in that they only require extraction of part of the training data. However, they still suffer from the same challenges as of extraction based definitions.

A.4 Compression with language models beyond arithmetic coding

Shannon (1948) noted that the optimal compression method for a given source is one that assigns codes to symbols such that the average code length approaches the entropy of the source. Arithmetic coding (Pasco, 1977; Rissanen, 1976) is known to be one optimal way to compress text given a distribution over symbols; it was used in (Delétang et al., 2024) to compress text using modern language models.

Although arithmetic coding is known to be optimal for samples generated from the random process of choice, it may still be sub-optimal for cases where the compressed samples are correlated with the choice of random process. Specifically, in language modeling, the training data is highly correlated with the model itself and hence we might need to treat them differently. For instance, we know from previous work that the models behavior on training data points is different from random samples. A large portion of training data can be generated using greedy decoding (Carlini et al., 2023b; Liu et al., 2025) which is a behavior not expected for randomly sampled data. To this end, we design a new compression technique, a generalization of arithmetic coding.

Ensemble compression. Sampling from language models involve two key parameters k for top_k selection and t for temperature. We design a compression method that sets these parameters adaptively. For instance, for cases where we know we can decode the next 100 tokens in a greedy fashion, we set $k = 1$ to reduce the bit length of arithmetic code. Changing the setup of the coding scheme itself requires a new token to be injected and wastes some number of bits, but it could still be beneficial for the code length. Our compression program uses dynamic programming to find the optimal code with injection of these new tokens in the middle of the text. Notably, our algorithm runs in time $O(n * T)$, where n is the number of tokens and T is the number of possible setups (combination of t and k) that we allow.

A.5 How reliable are our linear estimates of capacity?

Instead of scaling the number of examples in a dataset, we scale model sequence length to adjust the size of a dataset. We use the following measurement for expected memorization of a model:

$$\text{mem}(X, L(X)) \approx \min(\textit{capacity}(L), H(X))$$

	d_{emb}	n_{layer}	$\lvert\theta\rvert$	$\lvert D\rvert$	Predicted F1	Observed F1
GPT2-XL	1600	48	1,556,075,200	170,654,583	0.55	54.61 ± 1.3
				76,795,021	0.75	71.08 ± 0.4
				18,851,574	0.95	95.85 ± 0.8
GPT2-Medium	768	12	123,702,528	13,566,442	0.55	53.44 ± 1.1
				6,104,935	0.75	65.69 ± 0.6
				1,498,634	0.95	97.98 ± 0.3

Table 2: Dataset sizes that our scaling law predicts will produce a given membership inference F1, along with empirical values.

S	Params.	Memorized	Expected	Error
4	6.59×10^5	1.73×10^5	1.80×10^5	4.19
8	6.60×10^5	3.54×10^5	3.60×10^5	1.80
16	6.61×10^5	7.15×10^5	7.21×10^5	0.84
32	6.63×10^5	1.44×10^6	1.44×10^6	0.41
64	6.67×10^5	2.29×10^6	2.36×10^6	2.97
128	6.75×10^5	2.36×10^6	2.39×10^6	1.24
256	6.92×10^5	2.44×10^6	2.45×10^6	0.44

Table 3: Model capacity estimates across sequence length S, along with error (%).

V	Params.	Memorized	Expected	Error
128	4.21×10^5	1.49×10^6	1.49×10^6	0.36
512	4.71×10^5	1.71×10^6	1.67×10^6	2.78
1024	5.36×10^5	1.95×10^6	1.90×10^6	2.70
2048	6.67×10^5	2.39×10^6	2.36×10^6	1.11
4096	9.29×10^5	3.13×10^6	3.15×10^6	0.47

Table 4: Model capacity estimates across vocab size V, along with error (%).

we substitute our previous estimate of $\alpha = 3.642$ and ensure to adjust the parameter count for increases due to resizing the model's embedding matrices. We fix the number of training samples to 4096 and train a model with 2 layers and a hidden size of 128. Results are illustrated in Figure 10 and Table 3. Our predictions of total memorization are accurate, with an average error rate of 1.7% while scaling S and 1.8% when scaling V.

A.6 Additional memorization results

Our findings indicate that memorization of text data neatly plateaus near the model capacity just as in the synthetic data case. When the dataset size increases by a factor of N, the model divides its memorization between datapoints by an equal amount; the sum of memorization is measured to be constant, presumably at the upper bound of the model's capacity.

When the dataset is small enough for each model to fit – that is, below the capacity of the smallest model – we observe very similar performance between the models. For larger data sizes we notice an interesting trend: unintended memorization increases with dataset size for to a point, presumably as a model fills its capacity with the available information, and then decreases, as the model replaces sample-level information with more useful, generalizable knowledge. A given model generalizes the most (and memorizes the least information about any individual sample) when the dataset is maximally large.

A.7 Comparison of distributions memorized

Distribution-level analysis. Text sequences have very different properties than uniform synthetic bitstrings. We explore how two models of equal capacity spread their memorization across datapoints. We plot a histogram (Figure 14) of train and test compression rates of training data from both synthetic random bitstrings and text. Random training data follows a very normal distribution with a small amount of overlap between train and test compression rates. Text loss is lower on average but more spread out, with low loss on some training points and a long tail of higher losses. There is much more overlap between the train and test loss distributions, which explains why membership inference is more difficult for text data.

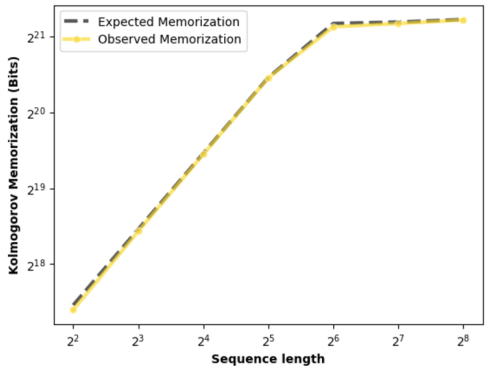

Figure 10: Model memorization across sequence lengths for a fixed-length dataset. Our predictions of total memorization are accurate, with an average error rate of 1.7%.

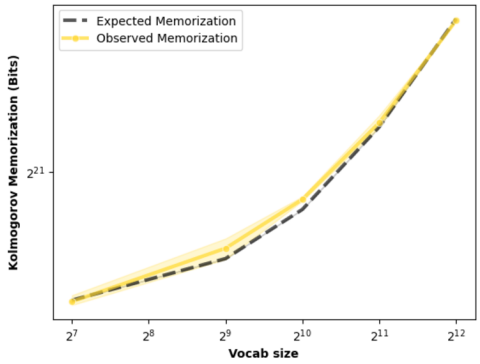

Figure 11: Model memorization across vocabulary size for a fixed-length dataset. Our predictions of total memorization are accurate, with an average error rate of 1.8%. Note that, we do not observe a capacity plateau, since increasing V also increases parameters.

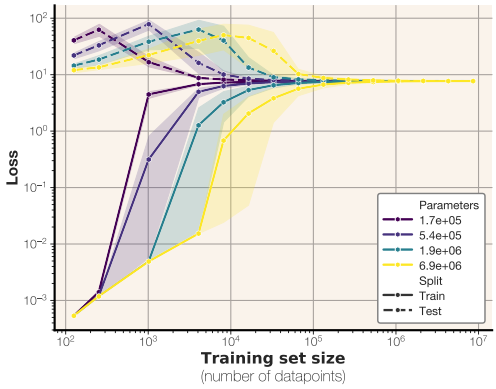

Figure 12: Train and test losses for different-sized language models trained on synthetic data.

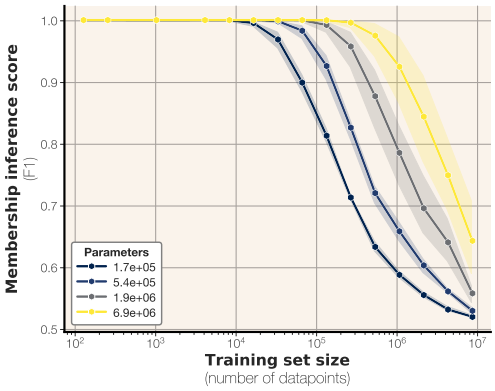

Figure 13: Membership inference attack performance decreases with dataset scale. In the case of uniform synthetic data, membership inference performance never falls below 0.54.

Which datapoints are most memorized? Our distribution-level analysis indicates that unlike in the random-bitstring case, models trained on a large amount of text are able to memorize a small number of datapoints. Prior work has indicated that a large amount of this memorization can be due to duplicated training points (Lee et al., 2022) but our dataset is fully deduplicated so this cannot be an explanation in our case.

To quantitatively evaluate the number of rare words per document, we measure the TF-IDF of each training document, plotted vs. unintended memorization in Figure 15. We use the following equation for TF-IDF:

$$\text{TF-IDF}(d; \mathcal{D}) = \frac{1}{|d|} \sum_{w \in d} \log \frac{|D|}{tf(w, \mathcal{D})}$$

where $tf(d, \mathcal{D})$ indicates the total number of times word w appears in dataset $\mathcal{D}$. Intuitively, a higher TF-IDF score for document d indicates that d contains more words that are rare in $\mathcal{D}$.

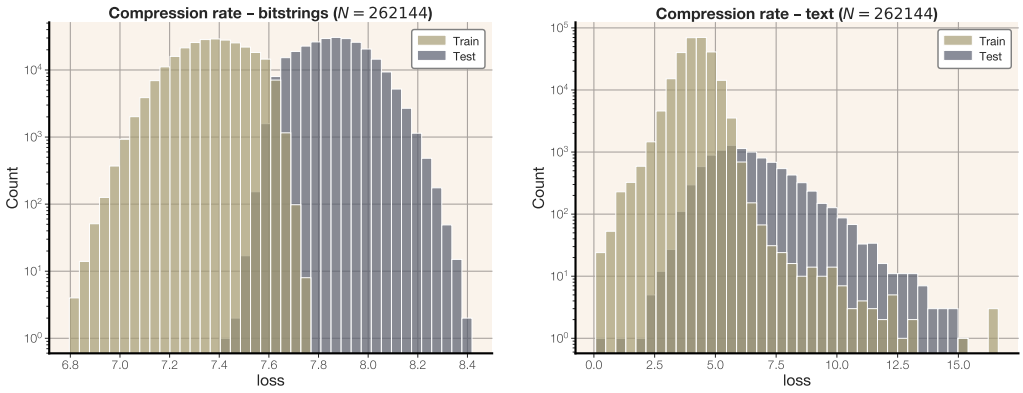

Figure 14: Distribution of compression rates for equal-sized transformers ($n_{\text{layer}} = 4$, $d_{\text{model}} = 128$) trained on 2^{14} sequences of equal-length random bitstrings (left) and text (right).

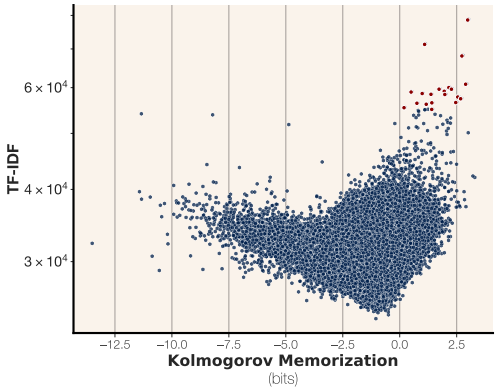

Figure 15: Unintended memorization vs. TF-IDF for all training points of a $20M$ param model trained past its capacity on 2^{16} sequences of English text. The training documents with rarest words are typically the most memorized.

We clearly observe for samples with positive unintended memorization there is a strong correlation between trainset TF-IDF and memorization: examples with more rare words are more memorized. In particular, the sample with highest TF-IDF out of the whole training dataset (a sequence of Japanese words) has the third-highest measured memorization; even though this is just one out of 260,000 training samples, the model can regurgitate the entire sequence given just a single token (囚). Out of the top twenty memorized sequences, all but three contain sequences of tokens from other languages (Japanese, Chinese, and Hebrew).

Manual analysis (Table 5) indicates that the most memorized datapoints have extremely rare tokens, typically ones not found in English.

A.8 Scaling law fit

Here we demonstrate the fit of our sigmoidal scaling law to experimental data. We show points in tokens-per-parameter vs. fit in Figure 16. Although the sigmoidal function is slightly simplistic (the points do not perfectly fit the curve) our fit produces estimates within $1 - 2\%$ of observations.

A.9 Proofs

In the section we provide the proofs missing from the main body.

	Text	TFIDF	Memorization	Language
0	人気エリアであるフォンニャに位置するRock & Roll Hostelは、ビジネス出張と観光のどちらにも最適なロケーションです。◆	78553.72	2.98	Japanese
1	このトピックには0件の返信が含まれ、1人の参加者がいます。1 年、 6 ヶ月前に Dave Gant さんが最後の更新	71279.19	1.09	Japanese
2	Label: Living Records\nDestroy All MonstersメンバーBen Millerによる自主レーベルからのソロCD-R。こちらは付属の抽象画をサウンド化したと	68064.46	2.73	Japanese
3	《左傳》記「崔氏側莊公于北郭。丁亥，葬諸士孫之里，四翼，不◆	60820.46	2.89	Chinese
4	歡迎客人自備紋身圖案或要求本紋身店代客起圖， 設計起圖須	60018.53	2.16	Chinese
5	By 小森 栄治,向山 洋一\nRead Online or Download 中学の理科「総まとめ」を7日間で攻略する本 「◆	59625.40	2.27	Japanese
6	統合分析是將一些議題相關但彼此獨立的臨床實驗之研究結果(大◆	59624.37	1.73	Chinese
7	在SIA-Smaart Pro的Real-Time Module实时模块上，将功能扩展，实时显示相位和Fixed Point Per Octave （	59128.54	1.95	Chinese
8	Progress in Intelligent Transportation Systems and IoT/M2M Communications: Markets, Standardization, Technologies\n|出版日||ページ情報||英文 173 Pages\n|インテリジェント交通システムお	58953.67	0.50	Japanese
9	English Title: Kingdom Hearts: Chain of Memories\nJapanese Title: キングダム ハーツ チェイン オブ メモリーズ — "Kingdom Hearts: Chain of Memories"\nAuthor: Tomoco Kanemaki\nIllustrator: Shiro	58605.92	0.99	Japanese
10	אשכול זה הועבר לארכיון. נא לשאול שאלה חדשה אם יש לך צורך	58420.30	1.37	Hebrew
11	在《易經》里单数为阳, 双数为阴. 我曾怀疑马来西亚政府也会	58382.40	1.98	Chinese
12	「XXI c.—21世紀人」第3回企画展 三宅一生ディレクション\n21_21 DESIGN SIGHT 第 3 回企画展の	57797.99	2.55	Japanese
13	无敌神马在线观看 重装机甲 睿峰影院 影院 LA幸福剧本\n时间：2020-12	57399.24	2.67	Chinese
14	季末小邪 回复 dgutkai: 楼主 您好 可以把项目源码发我吗？ 可以付◆	56539.93	2.46	Chinese
15	בכל סדנא אפשר לזהות את הילדים שהוריהם מאפשרים חופש יצי◆	56478.18	1.41	Hebrew
16	Ακαδημαϊκές Δημοσιεύσεις Μελών ΔΕΠ σε άλλα Ιδρύματα >\n◆	56376.74	0.75	Greek
17	Larry想和李华，还有她那些中国朋友多在一起玩儿，了解更多的中国文	56152.72	1.16	Chinese
18	Mark 5:18 wrote:καὶ ἐμβαίνοντος αὐτοῦ εἰς τὸ πλοῖον παρεκάλει αὐ	55391.28	0.19	Greek
19	.בתחילה הייתי סקפטית לגבי השקעת כסף בשיווק אינטרנטי	55014.00	1.41	Hebrew

Table 5: Highest TF-IDF training examples from a $20M$ param model trained past its capacity on 2^{16} sequences of English text. All of the highest TF-IDF examples are considered memorized, and contain text from non-English languages (Japanese, Chinese, Hebrew, and Greek).

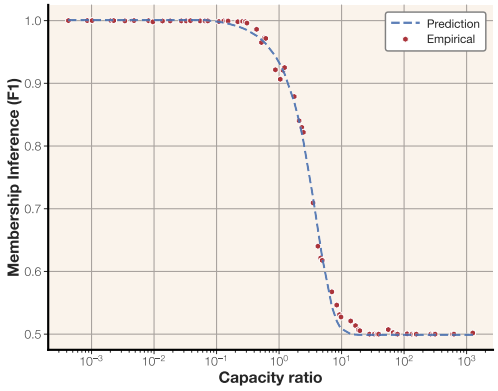

Figure 16: Our sigmoidal scaling law for membership inference fit to experimental data.

A.10 Proof of Proposition 1

Here we prove Proposition 1

Proof. we have

$$\begin{aligned}\mathrm{mem}_U(X, \hat{\Theta}, \Theta) &= I(X \mid \Theta, \hat{\Theta}) \\ &= I((X_1 \mid \Theta, \ldots, X_n \mid \Theta), \hat{\Theta}).\end{aligned}$$

And since the data is sampled i.i.d., all random variables in $\{R_i = [X_i \mid \Theta]\}_{i\in[n]}$ are independent. [3] So we have,

$$I((X_1 \mid \Theta, \ldots, X_n \mid \Theta), \hat{\Theta}) \geq \sum_{i\in[n]} I(X_i \mid \Theta, \hat{\Theta})$$

which implies

$$\text{mem}_U(X, \hat{\Theta}, \Theta) \geq \sum_{i\in[n]} \text{mem}_U(X_i, \hat{\Theta}, \Theta).$$

On the other hand, we have

$$\begin{aligned}\text{mem}_U(X, \hat{\Theta}, \Theta) &= I(X \mid \Theta, \hat{\Theta}) \\ &= H(\hat{\Theta} - H(\hat{\Theta} \mid (X \mid \Theta)) \\ &\leq H(\hat{\Theta})\end{aligned}$$

□

A.11 Proof of Proposition 4

Proof. We first state a Lemma about connection between algorithmic (kolmogorov) mutual information and mutual information.

Lemma 6. [Theorem 3.6 in Grunwald & Vitányi (2004)] Assume (X, Y) be a pair of joint random variables. Let f be the density function, $f(x, y) = \Pr[(X, Y) = (x, y)]$. Then we have

$$\begin{aligned}I(X, Y) - H_K(f) &\leq \mathop{\text{E}}_{(x,y)\sim(X,Y)}[I_K(x, y)] \\ &\leq I(X, Y) + 2H_K(f).\end{aligned}$$

Now we use this lemma to prove the statement of the Proposition. Let f be a the density function for the joint distribution $(X_i \mid \theta, \hat{\Theta})$. That is $f_i(x_i, \hat{\theta}) = \Pr[X_i = x_i \mid \theta \text{ and } \hat{\Theta} = \hat{\theta}]$. Note that this function is independent of n and θ. By definition we have

$$\text{mem}_U(X_i, \hat{\Theta}, \theta) = I(X_i \mid \theta, \hat{\Theta}).$$

Now using Lemma 6 we have

$$\begin{aligned}I(X_i \mid \theta, \hat{\Theta}) - H_K(f) &\leq \mathop{\text{E}}_{x_i\sim X_i|\theta}[I_K(x_i, \hat{\theta})] \\ &\leq I(X_i \mid \theta, \hat{\Theta}) + 2H_K(f).\end{aligned}$$

and this concludes the statement of Proposition by setting $\epsilon = 2H_K(f)$ □

A.12 Limitations

Our efforts to measure language model memorization come from a line of recent research to discover whether models have analyzed certain texts, and if so, how much. However, our main experimental contributions relate to the practice of training and evaluating language models, including a new perspective on the phenomenon of grokking (Nakkiran et al., 2019) and a new measurement of capacity. Our results are specific to the environment proposed and do not necessarily generalize to other datasets, architectures, or training setups.

[3] Note that X_i themselves are not independent because they are sampled by first sampling an underlying model Θ. However, they are conditionally independent once the underlying model Θ is given.