arXiv:2505.24832v2 [cs.CL] 2 Jun 2025

How much do language models memorize?

John X. Morris[1,3], **Chawin Sitawarin**[2], **Chuan Guo**[1], **Narine Kokhlikyan**[1], **G. Edward Suh**[3,4], **Alexander M. Rush**[3], **Kamalika Chaudhuri**[1], **Saeed Mahloujifar**[1]

[1]FAIR at Meta, [2]Google DeepMind, [3]Cornell University, [4]NVIDIA

We propose a new method for estimating how much a model "knows" about a datapoint and use it to measure the capacity of modern language models. We formally separate memorization into two components: *unintended memorization*, the information a model contains about a specific dataset, and *generalization*, the information a model contains about the true data-generation process. By eliminating generalization, we can compute the total memorization of a given model, which provides an estimate of model capacity: our measurements estimate that **models in the GPT family have an approximate capacity of 3.6 bits-per-parameter**. We train language models on datasets of increasing size and observe that models memorize until their capacity fills, at which point "grokking" begins, and unintended memorization decreases as models begin to generalize. We train hundreds of transformer language models ranging from $500K$ to $1.5B$ parameters and produce a series of scaling laws relating model capacity and data size to membership inference.

Date: June 3, 2025
Correspondence: Saeed Mahloujifar at saeedm@meta.com

Meta

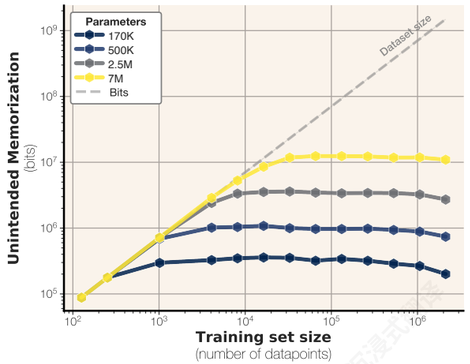

Figure 1 Unintended memorization of uniform random data (Section 3). Memorization plateaus at the empirical capacity limit of different-sized models from the GPT-family, approximately 3.6 bits-per-parameter.

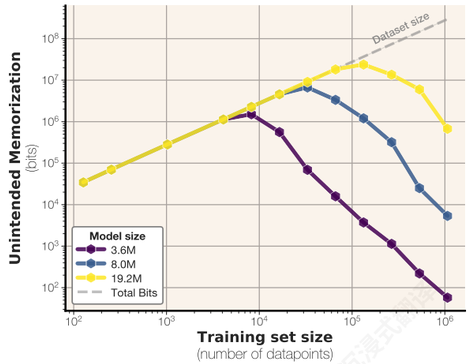

Figure 2 Unintended memorization of text across model and dataset sizes (Section 4). All quantities are calculated with respect to a large oracle model trained on the full data distribution.

1 Introduction

For the past several years, modern language models have been trained on increasingly large amounts of data, while parameter counts stay stagnant in the billions. For example, one recent state-of-the-art model (Dubey & et al, 2024) has 8 billion parameters (around $32GB$ on disk) but is trained on 15 trillion tokens (around $7TB$ on disk).

A long line of work (Carlini et al., 2019; Mireshghallah et al., 2022; Nasr et al., 2023; Zhang et al., 2023; Carlini et al., 2023b; Schwarzschild et al., 2024) questions whether such pretrained language models memorize their training data in a meaningful way. Most research approaches this problem either through the lens of

arXiv:2505.24832v2 [cs.CL] 2 Jun 2025

语言模型记住了多少?

约翰 · X· 莫里斯 1 、 3 、 3，查温 · 西塔瓦林 2，川国 1，纳里内 · 科赫利坎 1，G· 爱德华 · 苏 3 、 4 、 4，亚历山大 · M· 拉斯赫 3，卡玛莉卡 · 乔杜里 1，赛义德 · 马赫卢吉法尔 [1]

[1]Meta 的 FAIR, [2]Google DeepMind, [3] 康奈尔大学 , [4]NVIDIA

我们提出了一种新方法，用于估计模型 " 知道 " 数据点的程度，并使用它来衡量现代语言模型的容量。我们将记忆正式分为两个组成部分：非有意记忆，模型中关于特定数据集的信息，以及泛化，模型中关于真实数据生成过程的信息。通过消除泛化，我们可以计算给定模型的总记忆量，这提供了模型容量的估计：我们的测量表明 **GPT 系列的模型具有大约每参数 3.6 位的容量**。我们在规模不断增加的数据集上训练语言模型，并观察到模型在容量填满之前会记忆，此时 " 理解 " 开始，随着模型开始泛化，非有意记忆会减少。我们训练了数百个从 $500K$ 到 $1.5B$ 参数的 Transformer 语言模型，并产生了一系列将模型容量和数据大小与成员推理相关的缩放定律。

Date: June 3, 2025
对应: Saeed Mahloujifarat saeedm@meta.com

Meta

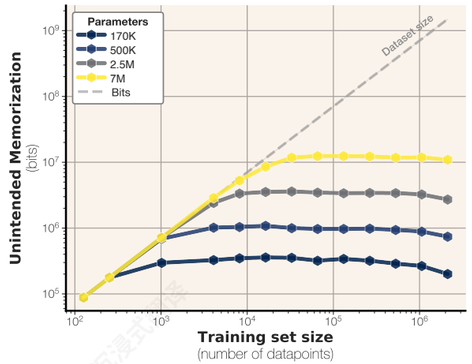

图 1 无意中记忆均匀随机数据 (第 3 节). 记忆在来自GPT 系列的具有不同尺寸模型的经验容量极限处达到平台期，大约为每个参数 3.6 比特。

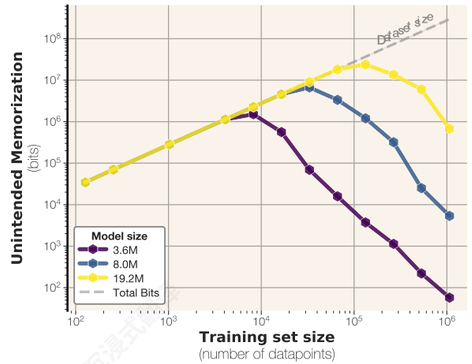

图 2 模型和数据集尺寸的文本无意中记忆（第 4 节). 所有量都是相对于一个在完整数据分布上训练的大型预言模型计算的。

1 引言

在过去的几年里，现代语言模型在越来越多的数据上进行了训练，而参数数量仍然保持在数十亿。例如，一个最近的最先进模型（ Dubey& et al， 2024 ）有 80 亿个参数（大约 $32GB$ 在磁盘上) 但在 150 万亿个标记（大约 $7TB$ 在磁盘上) 上进行训练。

一条长期的工作（ Carlini 等人，2019 年；Mireshghallah 等人，2022 年；Nasr 等人，2023 年；Zhang 等人，2023 年；Carlini 等人，2023b ； Schwarzschild 等人，2024 年）质疑了预训练语言模型是否以有意义的方式记住了其训练数据。大多数研究从以下角度探讨这个问题：

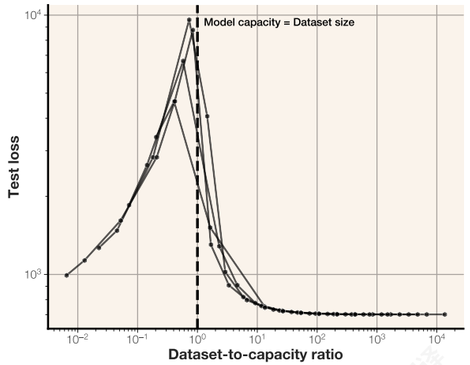

Figure 3 In our experiments on synthetic bitstrings, double descent occurs exactly when the dataset size begins to exceed the model's capacity, when unintended memorization is no longer beneficial for lowering the loss.

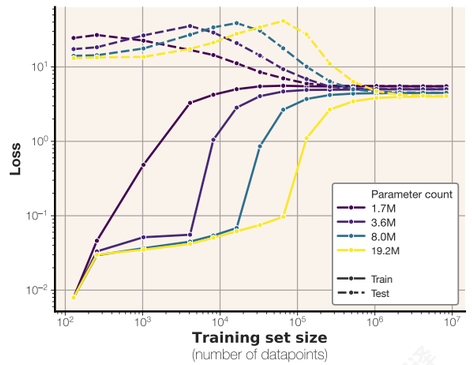

Figure 4 Train and test losses of different model and dataset sizes trained on text. Double descent occurs when dataset size exceeds model capacity.

extraction, aiming to recover full training data points from model weights, or membership inference, classifying whether a training point was present in the training data of a given model.

Studies of language model extraction argue that a data point is memorized if we can induce the model to generate it (Carlini et al., 2023b; Nasr et al., 2023; Schwarzschild et al., 2024). We argue that such generation does not necessarily serve as a proof of memorization. Language models can be coerced to output almost any string (Geiping et al., 2024); hence the fact that a model outputs something is not necessarily a sign of memorization. To address this issue, some researchers have suggested regularizing the input to the language model, such as by limiting its length Schwarzschild et al. (2024) or matching it to the prefix Carlini et al. (2023b) preceding the memorized sentence. However, even with these constraints, memorization cannot be conclusively proven, as the model's ability to generalize may still be at play. For example, a good language model prompted to add two numbers can output the correct answer without having seen the equation before. In fact, a recent work Liu et al. (2025) shows that some of the instances than were previously thought as memorized do not even exist in the training set and their extractability is a result of generalization. Additionally, verbatim reproduction of a text is not a prerequisite for memorization; a model may still be memorizing specific patterns or sequences, such as every other token, without generating them verbatim.

Given that extraction is neither necessary nor sufficient, defining memorization accurately is a pressing concern. Existing mathematical definitions of memorization, such as those based on membership inference Shokri et al. (2017) and differential privacy Dwork (2006), are defined in the dataset/distribution level. This makes them inadequate for measuring memorization for certain instances (e.g. a particular textbook). Theoretical notions of 'influence' Feldman (2020); Feldman & Zhang (2020) have been proposed to define memorization at the instance level, but they fall short of meeting our needs. These definitions focus on the capacity of a training algorithm to memorize inputs, whereas our interest lies in understanding how much a specific data point is memorized within a given model. This distinction is crucial, as we are concerned with the memorization properties of a single model, rather than the distribution of models generated by a particular training algorithm.

To bridge this gap, we propose a novel definition of memorization that quantifies the extent to which a model retains information about a specific datapoint. Our approach leverages the concept of compression rate in bits, where a model is considered to have memorized an input if it can be compressed to a significantly shorter encoding in the presence of the model. This framework draws inspiration from Kolmogorov information Kolmogorov (1963) theory and Shannon information Shannon (1948), but can be easily measured in practice using model likelihoods. We tackle the fundamental challenge of distinguishing between memorization and

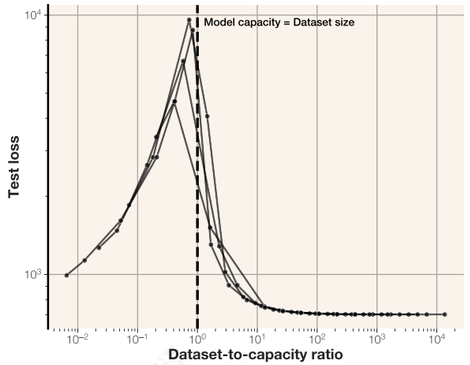

图 3 在我们的合成比特串实验中，当数据集大小开始超过模型的容量时，双下降现象就会发生，此时无意中的记忆不再有利于降低损失。

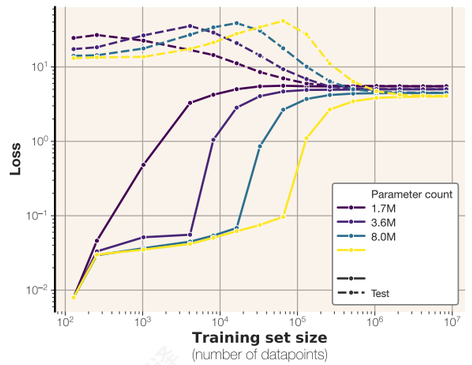

图 4 在文本上训练的不同模型和数据集大小的训练和测试损失。当数据集大小超过模型容量时会发生双下降。

提取，旨在从模型权重中恢复完整的训练数据点，或成员推理，分类一个训练点是否存在于给定模型的训练数据中。

关于语言模型提取的研究认为，如果我们能诱导模型生成某个数据点，那么这个数据点就被记住了（Carlini 等人， 2023b； Nasr 等人， 2023； Schwarzschild 等人， 2024）。我们认为这种生成并不一定证明记忆。语言模型可以被强迫输出几乎任何字符串（Geiping 等人，2024）；因此，模型输出某个内容并不一定意味着记忆。为了解决这个问题，一些研究人员建议对语言模型的输入进行正则化，例如通过限制其长度 Schwarzschild 等人 (2024) 或将其与记忆句子前面的前缀 Carlini 等人 (2023b) 匹配。然而，即使在这些限制下，也无法最终证明记忆，因为模型泛化能力可能仍然在起作用。例如，一个良好的语言模型在提示它添加两个数字时，可以在没有见过方程式之前输出正确答案。事实上，最近的一项工作 Liu 等人 (2025) 表明，以前被认为被记忆的一些实例甚至不存在于训练集中，它们的可提取性是泛化的结果。此外，逐字复制文本并不是记忆的必要条件；模型可能仍在记忆特定的模式或序列，例如每隔一个标记，而无需逐字生成它们。

鉴于提取既不是必要的也不是充分的，准确定义记忆是一个紧迫的问题。现有的记忆数学定义，例如基于成员推理 Shokri 等人 (2017) 和差分隐私 Dwork(2006)，是在数据集 / 分布级别定义的。这使得它们不适用于测量某些实例（例如特定的教科书）的记忆。已经提出了理论概念 “ 影响 ” Feldman (2020); Feldman & Zhang (2020) 来在实例级别定义记忆，但它们未能满足我们的需求。这些定义关注训练算法记忆输入的能力，而我们的兴趣在于理解特定数据点在给定模型中的记忆程度。这种区别至关重要，因为我们关注的是单个模型的记忆特性，而不是特定训练算法生成的模型分布。

为了弥补这一差距，我们提出了一种新的记忆定义，该定义量化了模型保留特定数据点信息的程度。我们的方法利用了比特压缩率的概念，其中如果模型在模型存在的情况下可以将输入压缩到显著更短的编码，则认为模型已经记住了输入。这个框架受到柯尔莫哥洛夫信息 Kolmogorov (1963) 理论和香农信息 Shannon(1948) 的启发，但可以通过模型似然性在实践中轻松测量。我们解决了区分记忆和

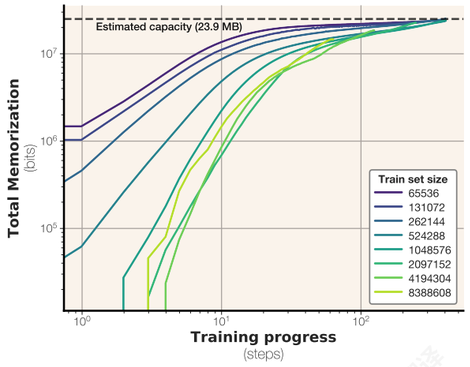

Figure 5 Bits memorized across training. This particular model is a GPT-style transformer with $6.86M$ parameters and a capacity of 23.9 MB.

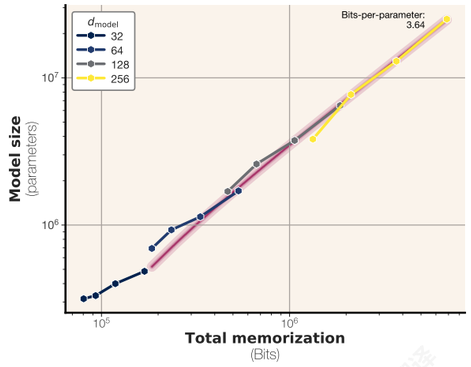

Figure 6 Capacity in bits-per-parameter for models trained on synthetic data. We estimate $\alpha = 3.64$ bits-per-parameter for GPT models trained in half precision.

generalization (Prashanth et al., 2024) by decomposing memorization into two distinct components: *unintended memorization*, which captures the information a model stores about a particular dataset, and *generalization*, which represents the knowledge a model has acquired about the underlying data-generating process.

To understand our new quantities, we measure unintended memorization and generalization by training language models of varying capacity on datasets of different sizes. We first eliminate the question of generalization entirely by training on a dataset of random uniformly-sampled bitstrings. In this setting, we can exactly measure the amount of information contained about the data inside the model. This gives us a principled way to measure language model *capacity* when trained on uniform datasets of exact known information content. We find that GPT-style transformers can store between 3.5 and 4 bits of information in each model parameter, depending on model architecture and precision.

We then repeat our experiments with real text, where generalization is possible and even beneficial for learning. On real text, language models memorize up to a certain capacity, at which point they substitute unintended memorization for generalization, and begin to learn general, reusable patterns as opposed to sample-level specifics. Our framework shows that double descent phenomenon begins to occur at this point, when the data size exceeds the model capacity in bits.

Finally, we use our results to predict a scaling law for membership inference performance based on model capacity and dataset size. We show that membership inference follows a clean relationship based on model capacity and dataset size: bigger models can memorize more samples, and making datasets bigger makes membership inference harder. Our scaling laws extrapolate to larger models, and predict most modern language models are trained on too much data to do reliable membership inference on the average data point.

2 Memorization, intended and unintended

When a model $\theta = L(x)$ is trained using a training algorithm L and a dataset $x \sim X$, some information is transferred from the sample x to the model θ. A key question in the memorization literature is determining how much of this stored information is intended versus unintended. In this work, we aim to provide a rigorous definition of memorization that satisfies certain properties:

1. *Separation from generalization.* Our notion of unintended memorization must be distinct from intended memorization, which we refer to as generalization. For example, consider a language model trained on the sample: *Q: What is* 2^{100}*? A: 1267650600228229401496703205376.* When assessing how much of this training sample is memorized, we must account for the fact that performing simple math operations is expected from a language model.

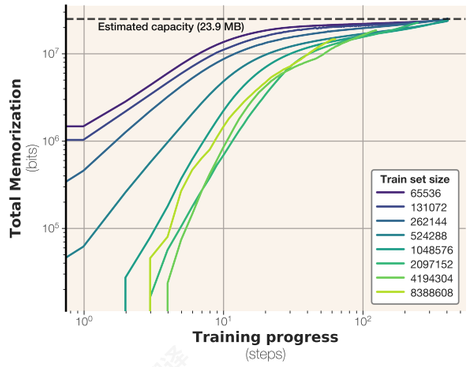

图 5 训练过程中记忆的比特数。这个特定的模型是一个 GPT 风格的 transformer，具有 $6.86M$ 参数和 23.9 MB 的容量。

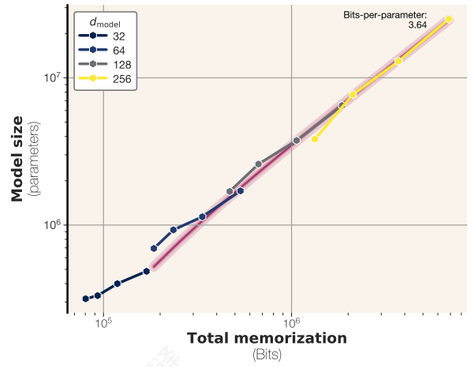

图 6 合成数据上训练的模型的比特数 / 参数容量。我们估计 GPT 模型在半精度训练时的比特数 / 参数为 $\alpha = 3.64$ 。

泛化（Prashanth 等人。,2024）通过将记忆分解为两个不同的组成部分：非有意记忆，它捕获模型存储的关于特定数据集的信息，以及泛化，它代表了模型获得的关于底层数据生成过程的知识。

为了理解我们的新量，我们通过在不同大小的数据集上训练不同容量的语言模型来测量非有意记忆和泛化。我们首先通过在随机均匀采样的比特串数据集上训练来完全消除泛化的问题。在这种情况下，我们可以精确地测量模型中包含的数据量。这为我们提供了一种在具有精确已知信息容量的均匀数据集上训练语言模型时测量其容量 的原则性方法。我们发现 GPT 风格的 transformer 可以在每个模型参数中存储 3.5 到 4 比特的信息，具体取决于模型架构和精度。

然后，我们在真实文本上重复我们的实验，其中泛化是可能的，甚至对学习有益。在真实文本上，语言模型记忆到一定的容量，在这个点上，它们用无意的记忆来替代泛化，并开始学习通用、可重用的模式，而不是样本级别的具体细节。我们的框架表明，当数据大小超过模型容量时，双下降现象开始出现。

最后，我们使用我们的结果来预测基于模型容量和数据集大小的成员推理性能的缩放定律。我们表明，成员推理基于模型容量和数据集大小遵循一个清晰的关系：更大的模型可以记忆更多的样本，而使数据集变大会使成员推理变得更难。我们的缩放定律外推到更大的模型，并预测大多数现代语言模型在太多数据上进行训练，无法对平均数据点进行可靠的成员推理。

2 记忆，有意和无意的

当模型 $\theta = L(x)$ 使用训练算法 L 和数据集 $x \sim X$ 进行训练时，一些信息从样本 x 转移到模型 θ。记忆文献中的一个关键问题是确定这些存储信息中有多少是有意的，有多少是无意的。在这项工作中，我们的目标是提供一个满足某些属性的严格记忆定义：

1. 与泛化相分离。 我们对非预期记忆的概念必须与预期记忆相区别，后者我们称为泛化。例如，考虑一个在样本上训练的语言模型：*Q:* 2_{100}*? A: 1267650600228229401496703205376*。 在评估这个训练样本中有多少被记忆时，我们必须考虑到语言模型执行简单的数学运算是可以预期的。

2. *Sample-level memorization.* We need to define memorization for realizations of random variables, not the random variables themselves. Specifically, we want to determine how much unintended memorization of a sample x occurs in a model θ.
3. *Independence from training algorithm.* Our definition should be independent of the training algorithm L and only a function of the final model θ and the sample x. This is crucial for language models, where we often only have access to the final model and target sample.

Previous works have attempted to define memorization for machine learning models. We aim to provide precise definitions of memorization that meet our criteria, and offer ways to measure it. See Appendix A.2 for a broader discussion on definitions of memorization.

2.1 A statistical view of memorization

Notation. In this section, we use capital letters (e.g. X, Θ) to refer to random variables and lowercase letters to refer to instances of a random variable (e.g. $x \sim X$ and $\theta \sim \Theta$).

Information theory has developed well understood notions of information for random variables. For a random variable X, we often use $H(X)$, the entropy of X, to define the amount of information present in X. Moreover, for two distinct random variables X, Y, we can define $X \mid Y$ to be the uncertainty left in X after fixing Y. Having defined this quantity, we can now measure *mutual information* between X and Y by subtracting the leftover information from the total information: $I(X,Y) = H(X) - H(X \mid Y)$.

Now assume we have a machine learning pipeline. We have a prior Θ on the underlying model that captures our dataset distribution X. And we have a learning algorithm L that maps samples from X to a trained model $\hat{\Theta}$. To understand how much information about X is stored in $\hat{\Theta}$, we can use the notion of mutual information:

$$\mathrm{mem}(X, \hat{\Theta}) = I(X, \hat{\Theta}) = H(X) - H(X \mid \hat{\Theta}).$$

Note that this captures all the information about X that is stored in $\hat{\Theta}$. As we discussed, we need our notion of memorization to account for generalization as well. So when measuring unintended memorization, we are only interested in the information that is present in $X \mid \Theta$, which is the uncertainty left in X after fixing Θ. Hence, we can define **unintended memorization** as

$$\mathrm{mem}_U(X, \hat{\Theta}, \Theta) = I([X \mid \Theta], \hat{\Theta}) = H(X \mid \Theta) - H(X \mid (\Theta, \hat{\Theta})).$$

and then the **generalization** (or intended memorization) must be

$$\mathrm{mem}_I(\hat{\Theta}, X, \Theta) = \mathrm{mem}(X, \Theta) - \mathrm{mem}_U(X, \hat{\Theta}, \Theta) = I(X, \hat{\Theta}) - I(X \mid \Theta, \hat{\Theta})$$

Now that we have defined our notions of intended and unintended memorization we turn our attention to practically measuring them. Let us first state a proposition that enables measurement of unintended memorization:

Proposition 1 (Super-additivity of Unintended Memorization). Assume $X = (X_1, \ldots, X_n)$ is a dataset of n i.i.d. samples. We have

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

This proposition shows that to measure a lower bound on the unintended memorization on the dataset level, we can sum per-sample memorization. On the other hand, the entropy of the information content of the trained model itself servers as an upper bound on the unintended memorization. Another implication of this implies that unintended memorization should scale with the dataset size but cannot exceed the total capacity of the model.

2. 样本级别的记忆。 我们需要为随机变量的实现定义记忆，而不是随机变量本身。具体来说，我们希望确定模型 θ 中样本 x 的非预期记忆量。

3. 与训练算法无关。 我们的定义应该与训练算法 L 无关，并且仅是最终模型 θ 和样本 x 的函数。这对于语言模型至关重要，因为我们通常只能访问最终模型和目标样本。

以往工作曾尝试为机器学习模型定义记忆。我们旨在提供符合我们标准的记忆精确定义，并提供衡量记忆的方法。参见附录 A.2 以了解关于记忆定义的更广泛讨论。

2.1 A statistical view of memorization

符号说明。 在本节中，我们使用大写字母（例如 X, Θ）来指代随机变量，使用小写字母来指代随机变量的实例（例如 $x \sim X$ 和 $\theta \sim \Theta$）。

信息论已经发展出对随机变量的信息概念的清晰理解。对于一个随机变量 X，我们通常使用 $H(X)$，即 X 的熵，来定义 X 中包含的信息量。此外，对于两个不同的随机变量 X, Y，我们可以定义 $X \mid Y$ 为在固定 Y 后 X 中剩余的不确定性。定义了这个量之后，我们可以通过从总信息中减去剩余信息来测量互信息： $I(X, Y) = H(X) - H(X \mid Y)$。

现在假设我们有一个机器学习流程。我们有一个关于底层模型的先验 Θ ，它捕获了我们的数据集分布 X。我们还有一个学习算法 L ，它将 X 中的样本映射到一个训练好的模型Θ。为了了解关于 X 的信息有多少存储在Θ 中，我们可以使用互信息的概念：

$$\mathrm{mem}(X, \hat{\Theta}) = I(X, \hat{\Theta}) = H(X) - H(X \mid \hat{\Theta}).$$

请注意，这捕获了存储在Θ 中的关于 X 的所有信息。正如我们所讨论的，我们需要我们的记忆概念来解释泛化。因此，在测量非意图记忆时，我们只对存储在 $X \mid \Theta$ 中的信息感兴趣，$X \mid \Theta$ 是在固定 Θ 后 X 中剩余的不确定性。因此，我们可以定义 **非意图记忆** 为

$$\mathrm{mem}_U(X, \hat{\Theta}, \Theta) = I([X \mid \Theta], \hat{\Theta}) = H(X \mid \Theta) - H(X \mid (\Theta, \hat{\Theta})).$$

然后，**泛化** (或预期记忆) 必须是

$$\mathrm{mem}_I(\hat{\Theta}, X, \Theta) = \mathrm{mem}(X, \Theta) - \mathrm{mem}_U(X, \hat{\Theta}, \Theta) = I(X, \hat{\Theta}) - I(X \mid \Theta, \hat{\Theta})$$

现在我们已经定义了预期和意外记忆的概念，我们将注意力转向实际测量它们。让我们首先陈述一个命题，该命题能够测量意外记忆：

命题 1 (意外记忆的超可加性). 假设 $X = (X_1, \ldots, X_n)$ 是一个包含 n 个独立同分布样本的数据集。我们有

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

这个命题表明，为了测量数据集层面上的意外记忆下限，我们可以对每个样本的记忆进行求和。另一方面，训练模型的自身信息内容的熵可以作为意外记忆的上限。这一点的另一个推论是，意外记忆应该随着数据集大小而扩展，但不能超过模型的总容量。

2.2 Measuring unintended memorization with Kolmogorov Complexity

Our definitions of memorization and generalization so far are defined using an "entropy-based" notion of information. This means our definitions can only be used for random variables. This brings big challenges in measuring memorization. All our variables in the definition of memorization are singletons. We have a single underlying model θ, we have a single dataset $x = (x_1, \ldots, x_n)$ and we have a single trained model $\hat{\theta}$[1]. It is impossible to measure the entropy (let alone conditional entropy) of the underlying variables using a single sample.

To this end, we switch to another notion of information based on compression, then later we show how this notion closely approximates the notion of memorization defined above. Kolmogorov complexity defines the information content of a string x, denoted as $H^K(x)$, to be the length of shortest representation of x in a given computational model. Similarly, we can define the leftover information $x \mid \theta$, to be the shortest representation of x, when we have θ available as a reference. And the information content of $x \mid \theta$, denoted by $H^K(x \mid \theta)$, is the length of such description. Then, we can define mutual information in a similar fashion:

Definition 2 (Kolmogorov complexity). Let f be an arbitrary computational model that takes a set of inputs and returns an output (e.g. universal Turing machine). The shortest description of x with respect to computational model f is defined as $H^K(x) = \min_{f(p)=x} |p|$. Also, the Kolmogorov complexity of x relative to another string θ is defined as $H^K(x \mid \theta) = \min_{f(p,\theta)=x} |p|$. And we define the Kolmogorov mutual information between x and θ by $I^K(x, \theta) = H^K(x) - H^K(x \mid \theta)$. We assume inputs are bitstrings and $|p|$ is the bit length of the input.

Definition 3 (Kolmogorov memorization). Let θ be a reference model that approximates the true distribution of data, and $\hat{\theta}$ be a model trained on a dataset $x = (x_1, \ldots, x_n)$. For each x_i we define the memorization of x_i in $\hat{\theta}$ as $\text{mem}^K(\hat{\theta}, x) = I^K(\hat{\theta}, x)$. We also define intended and unintended variants of memorization:

$$\text{mem}^K_U(x, \theta, \hat{\theta}) = H^K(x \mid \theta) - H^K(x \mid (\theta, \hat{\theta})), \text{ and } \text{mem}^K_I(x, \theta, \hat{\theta}) = \text{mem}^K(x, \hat{\theta}) - \text{mem}^K_U(x, \theta, \hat{\theta}).$$

There are known connections between Kolmogorov complexity and Shannon Entropy (Grunwald & Vitanyi, 2004). These results point at the conceptual connection between the two notions and imply that $\mathrm{E}_{x\sim X}[H^K(x)] \approx H(X)$. Interestingly, this implies that our notion of Kolmogorov memorization closely approximates Shannon memorization.

Proposition 4. Let $X = (X_1, \ldots, X_n)$ be an i.i.d, dataset distribution parametrized by ground-truth model θ. Let L be a training algorithm mapping X to $\hat{\Theta}$. Assume $H(\hat{\Theta}) = \ell$ and $H(X_i) = \ell'$ [2]. Then we have $\left| \mathrm{E}_{\substack{x\sim X \\ \hat{\theta}\sim L(x)}} \left[\text{mem}^K_U(x_i, \hat{\theta}, \theta)\right] - \text{mem}_U(X_i, \hat{\Theta}, \theta) \right| \leq \epsilon$. for some constant ϵ independent of θ, ℓ, ℓ' and n. Moreover, we have tail bounds

$$\Pr_{\substack{x\sim X \\ \hat{\theta}\sim L(x)}} \left[\left| \Gamma(x, \hat{\theta}) \right| \geq r + c \right] \leq e^{-2\frac{c^2}{\ell\ell'}}.$$

2.3 Estimating Kolmogorov with likelihoods

Fixing our notion of Kolmogorov memorization, we now describe how we can estimate H^K in different setups. Note that exact calculation of Kolmogorov complexity is known to be uncomputable (the decision version of is undecidable). However, we can still approximate it using the best available compression schemes. Below, we summarize how we approximate each term in our definition.

- $H^K(x \mid \hat{\theta})$: Here, $\hat{\theta}$ is the trained target model, which does not necessarily capture the true data distribution. Because compression rate is inherently tied to the likelihood under a predictive model model (Shannon, 1950), we can easily estimate $H^K(x \mid \hat{\theta})$ using $p(x \mid \hat{theta})$, the likelihood of x under the target model.

[1]Note the switch to lowercase variables because we are now working with instances, not random variables.
[2]The trained model and each data sample can be presented using ℓ and ℓ' bits respectively.

2.2 使用 Kolmogorov 复杂度测量非预期的记忆

我们目前的记忆和泛化定义是使用基于“熵”的信息概念来定义的。这意味着我们的定义只能用于随机变量。这给测量记忆带来了巨大的挑战。记忆定义中的所有变量都是单例。我们有一个单一的基础模型 θ，我们有一个单一的数据集 $x = (x_1, \ldots, x_n)$，我们有一个单一的训练模型 θ[1]。使用单个样本测量基础变量的熵（更不用说条件熵）是不可能的。

为此，我们转向基于压缩的另一种信息概念，然后我们展示了这种概念如何紧密地逼近上面定义的记忆概念。Kolmogorov 复杂度定义字符串 x 的信息内容 $H^K(x)$ 为其在给定计算模型中最短表示的长度。类似地，我们可以定义剩余信息 $x \mid \theta$，当 θ 可用时，它是 x 的最短表示。并且 $x \mid \theta$ 的信息内容 $H^K(x \mid \theta)$ 是这种描述的长度。然后，我们可以以类似的方式定义互信息：

定义 2 (Kolmogorov 复杂度). 设 f 是一个任意的计算模型，它接受一组输入并返回一个输出（例如通用图灵机）。相对于计算模型 f，x 的最短描述定义为 $H^K(x) = \min_{f(p)=x} |p|$. 此外，$x$ 相对于另一个字符串 θ 的 Kolmogorov 复杂度定义为 $H^K(x \mid \theta) = \min_{f(p,\theta)=x} |p|$. 并且我们通过 $I^K(x, \theta) = H^K(x) - H^K($ $x \mid \theta)$. 定义 x 和 θ 之间的 Kolmogorov 互信息。我们假设输入是位串，$|p|$ 是输入的位长度。

定义 3 (Kolmogorov 记忆). 设 θ 是一个参考模型，它近似数据的真实分布，$\hat{\theta}$ 是一个在数据集 $x = (x_1, \ldots,$ $x_n)$上训练的模型。对于每个 x_i，我们定义$\hat{\theta}$ 中 x_i 的记忆为 $\text{mem}^K(\hat{\theta}, x) = I^K(\hat{\theta}, x)$. 我们还定义记忆的预期和非预期变体：

$$\text{mem}^K_U(x, \theta, \hat{\theta}) = H^K(x \mid \theta) - H^K(x \mid (\theta, \hat{\theta})), \text{ and } \text{mem}^K_I(x, \theta, \hat{\theta}) = \text{mem}^K(x, \hat{\theta}) - \text{mem}^K_U(x, \theta, \hat{\theta}).$$

已知 Kolmogorov 复杂度与香农熵之间存在已知联系（ Grunwald & Vi-tanyi， 2004 ）。这些结果揭示了两种概念之间的联系，并表明 $\mathrm{E}_{x\sim X}[H^K(x)] \approx H(X)$。有趣的是，这表明我们对 Kolmogorov 记忆的概念与 Shannon 记忆非常接近。

命题 4. 设 $X = (X_1, \ldots, X_n)$ 是一个由真实模型 θ 参数化的独立同分布数据集分布。设 L 是一个将 X 映射到Θ 的训练算法。假设 $H(\Theta) = \ell$ 和 $H(X_i) = \ell'$ [2]。那么我们有 $\left| \mathrm{E}_{\substack{x\sim X \\ \hat{\theta}\sim L(x)}} \left[\text{mem}^K_U(x_i, \hat{} \theta, \theta)\right] - \text{mem}_U(X_i, \hat{}\right.$ $\left.\Theta, \theta) \right| \leq \epsilon$.对于某个与 θ, ℓ, ℓ'和 n 无关的常数 ϵ。此外，我们有尾部界限

$$\Pr_{\substack{x\sim X \\ \hat{\theta}\sim L(x)}} \left[\left| \Gamma(x, \hat{\theta}) \right| \geq r + c \right] \leq e^{-2\frac{c^2}{\ell\ell'}}.$$

2.3 使用似然估计 Kolmogorov

修正我们对 Kolmogorov 记忆的理解，我们现在描述了如何在不同的设置中估计 H^K 。请注意，Kolmogorov 复杂度的精确计算已知是不可计算的（其决策版本是不可判定的）。然而，我们仍然可以使用最佳的可用压缩方案来近似它。下面，我们总结了我们定义中如何近似每个项。

- $H^K(x \mid \hat{}\theta)$: 这里，θ 是训练后的目标模型，它不一定能捕捉到真实的数据分布。由于压缩率本质上与预测模型下的似然性相关（ Shannon， 1950 ），我们可以利用 $H^K(x \mid \hat{}\theta)$ 来估计 $p(x \mid \hat{}theta)$，即目标模型下 x 的似然性。

[1]注意变量转换为小写，因为我们现在处理的是实例，而不是随机变量 .[2] 训练后的模型和每个数据样本分别可以使用 ℓ 和 ℓ' 位来表示。

n_{layer}	d_{model}	Params	Capacity(θ) *[bits]* fp32	bf16	α *[bpp]* fp32	bf16
1	32	8.04×10^4	3.39×10^5	3.16×10^5	4.23	3.93
	64	1.85×10^5	7.27×10^5	6.93×10^5	3.92	3.74
	128	4.69×10^5	1.71×10^6	1.69×10^6	3.65	3.61
	256	1.33×10^6	4.15×10^6	3.83×10^6	3.12	2.88
2	32	9.31×10^4	3.87×10^5	3.31×10^5	4.16	3.56
	64	2.35×10^5	9.60×10^5	9.27×10^5	4.08	3.94
	128	6.67×10^5	2.66×10^6	2.60×10^6	3.99	3.89
	256	2.12×10^6	8.49×10^6	7.76×10^6	4.01	3.66
4	32	1.18×10^5	4.65×10^5	3.99×10^5	3.92	3.37
	64	3.35×10^5	1.34×10^6	1.14×10^6	3.98	3.39
	128	1.06×10^6	4.02×10^6	3.75×10^6	3.78	3.53
	256	3.70×10^6	1.36×10^7	1.30×10^7	3.68	3.51
8	32	1.69×10^5	5.12×10^5	4.85×10^5	3.02	2.86
	64	5.35×10^5	2.05×10^6	1.71×10^6	3.83	3.19
	128	1.86×10^6	7.23×10^6	6.49×10^6	3.89	3.49
	256	6.86×10^6	2.71×10^7	2.51×10^7	3.96	3.65
					Mean ($\pm$0.1):	
					3.83	3.51

Table 1 Model capacity estimates across different widths and depths in full and half-precision. Doubling precision from bfloat16 to float32 only increases model capacity from 3.51 to 3.83 bits-per-parameter.

- $H^K(x \mid \hat{\theta}, \theta)$: In this case, the compression algorithm has access to both target and reference models. We simply compute $\max\{p(x \mid \hat{\theta}, p(x \mid \theta)\}$. In practice, our choice of reference model is a larger model with the same architecture as θ trained for many steps on a much wider data distribution.

3 Model Capacity for Memorization

Unintended memorization provides us a principled way of measuring the precise number of bits a model θ knows about a datapoint x. If we add up the information for each datapoint in a dataset, we can measure the total amount of bits a model knows about the dataset. And in cases where generalization is not possible because each datapoint is completely independent, we can estimate the **capacity** of a given model θ by summing per-datapoint unintended memorization.

3.1 Defining model capacity

We first formalize this notion of memorization capacity for a particular language model θ. Capacity is the total amount of memorization that can be stored in θ across all its parameters.

Definition 5 (Capacity). Let X be a distribution and $L\colon X \to \Theta$ a learning algorithm. We define the capacity of the learning algorithm L to be

$$\text{Capacity}(L) = \max_X \text{mem}\big(X, L(X)\big)$$

When the model capacity is reached, $\text{mem}(X, L(X))$ will no longer increase with dataset size. In practice, we can compute capacity by training to saturation on varying sizes of X and computing the maximum memorization.

n_{layer}	d_{model}	Params	容量 (θ) */ 比特 /* fp32	bf16	α *[bpp]* fp32	bf16
1	32	8.04×10^4	3.39×10^5	3.16×10^5	4.23	3.93
	64	1.85×10^5	7.27×10^5	6.93×10^5	3.92	3.74
	128	4.69×10^5	1.71×10^6	1.69×10^6	3.65	3.61
	256	1.33×10^6	4.15×10^6	3.83×10^6	3.12	2.88
2	32	9.31×10^4	3.87×10^5	3.31×10^5	4.16	3.56
	64	2.35×10^5	9.60×10^5	9.27×10^5	4.08	3.94
	128	6.67×10^5	2.66×10^6	2.60×10^6	3.99	3.89
	256	2.12×10^6	8.49×10^6	7.76×10^6	4.01	3.66
4	32	1.18×10^5	4.65×10^5	3.99×10^5	3.92	3.37
	64	3.35×10^5	1.34×10^6	1.14×10^6	3.98	3.39
	128	1.06×10^6	4.02×10^6	3.75×10^6	3.78	3.53
	256	3.70×10^6	1.36×10^7	1.30×10^7	3.68	3.51
8	32	1.69×10^5	5.12×10^5	4.85×10^5	3.02	2.86
	64	5.35×10^5	2.05×10^6	1.71×10^6	3.83	3.19
	128	1.86×10^6	7.23×10^6	6.49×10^6	3.89	3.49
	256	6.86×10^6	2.71×10^7	2.51×10^7	3.96	3.65
					Mean ($\pm$0.1):	
					3.83	3.51

Table 1 Model capacity estimates across different widths and depths in full and half-precision. Doubling precision from bfloat16 to float32 only increases model capacity from 3.51 to 3.83 bits-per-parameter.

- $H^K(x \mid \hat{\theta}, \theta)$: 在此情况下，压缩算法可以访问目标模型和参考模型。我们只需计算 $\max\{p(x \mid \hat{\theta}, p(x \mid \theta)\}$。在实践中，我们选择的参考模型是一个具有与 θ 相同架构的较大模型，该模型在更广泛的数据分布上训练了多个步骤。

3 模型的记忆能力

意外的记忆为我们提供了一种原则性的方法，用于测量模型 θ 关于数据点 x 知道的精确位数。如果我们对数据集中的每个数据点的信息进行求和，就可以测量模型关于数据集知道的总位数。并且在泛化不可能的情况下（因为每个数据点完全独立），我们可以通过逐个数据点的意外记忆总和来估计给定模型 θ 的 **能力**。

3.1 定义模型能力

我们首先为特定语言模型 θ 正式化这种记忆能力概念。容量是可以在 θ 的所有参数中存储的总记忆量。

定义 5 (容量)。设 X 是一个分布，$L\colon X \to \Theta$ 是一个学习算法。我们定义学习算法 L 的容量为

$$\text{Capacity}(L) = \max_X \text{mem}\big(X, L(X)\big)$$

当模型容量达到时，$\text{mem}(X, L(X))$ 将不再随数据集大小增加。在实践中，我们可以通过在不同大小的 X 上训练到饱和并计算最大记忆量来计算容量。

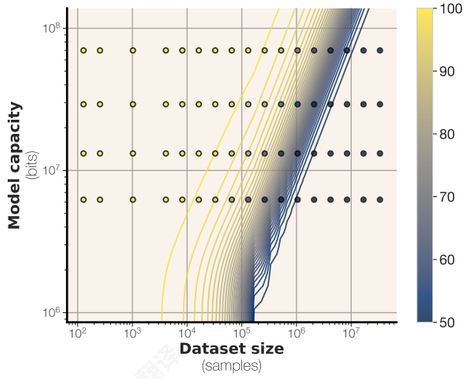

Figure 7 Scaling law curves for membership inference overlaid with empirical data, shown in circles.

3.2 Measuring model capacity with synthetic sequences

In this section we measure the capacity of Transformer language models. Our goal is to instantiate multiple datasets and distributions and measure the memorization of them when training a single model θ. Then, we take the maximum over all datasets to approximate of the model's capacity. For instantiating our datasets, each token is uniformly sampled from a predefined set of tokens independent of the previous tokens.

To approximate $H^k(x \mid \theta, \hat{\theta})$, we can directly compute entropy under the trained model to calculate the shortest description of the dataset conditioning on θ. Subtracting the two, we can approximate the unintended memorization $\text{mem}_U(X, L(X))$. Since the process for sampling the data is completely random, there is no generalization to be stored within $\hat{\theta}$ (that is, $\text{mem}^U(X, L(X)) \approx \text{mem}(X, L(X))$).

Observe that when we sample synthetic sequences from a uniform distribution, we can compute their Shannon information exactly. Given a dataset size N, we construct a dataset of N sequences, each of S tokens. Given a vocabulary size V, we can calculate the total entropy of a dataset x^i with such parameters by $H(x^i) = NS \log_2 V$. Then we calculate the compressed form x^i using entropy under $\hat{\theta}_i$ to compute the code length and use this as an approximation of $H^K(x^i \mid \hat{\theta}_j)$. Then we calculate the $\text{mem}(x^i, \hat{\theta}_i) = H(x^i) - H^K(x^i \mid \hat{\theta}_j)$ and compute a model's capacity as the maximum amount of memorization over all datasets.

Experimental details. In accordance with Kaplan et al. (2020), we train models with the GPT-2 architecture (Radford et al., 2019) initialized from scratch. Our models have between 1 and 8 layers, hidden dimensions scaled from 32 to 512, and from $100K$ to $20M$ parameters. We train models for 10^6 steps with a batch size of 2048. We use the Adam optimizer. All models are trained on a single A100 GPU in bfloat16 precision, and we use gradient accumulation if a batch cannot fit in memory. Unless otherwise noted, we set vocabulary size $V = 2048$, sequence length $S = 64$ and vary only the number of points in a dataset. We train each model on each dataset size over five random seeds, which affect both model initialization and the dataset sampling.

Results. We plot memorization across model and data sizes in Figure 1. This allows us to visualize unintended memorization amounts (y-axis) across dataset sizes (x-axis) grouped by model size (line color). We observe a striking plateau once a model reaches its capacity. Given the dataset is large enough, models exhibit an upper bound in net memorization, regardless of data size. Small datasets are completely memorized by all models with enough capacity.

We estimate the capacity of each model as the maximum amount of unintended memorization in bits measured across all dataset sizes. We then compare this capacity to the model size in Figure 6. Interestingly, even at this small scale, we see a very smooth relationship between observed capacity (maximum memorization measured over all datasets) and model parameters. We plot this relationship in Figure 6: under these settings,

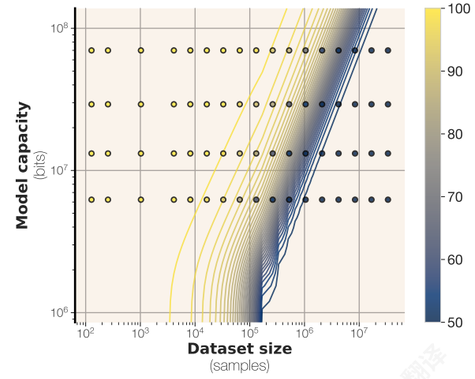

Figure7 用于成员推理的缩放律曲线，与经验数据（以圆圈表示）叠加显示。

3.2 使用合成序列测量模型能力

在本节中，我们测量 Transformer 语言模型的能力。我们的目标是为多个数据集和分布实例化，并在训练单个模型 θ 时测量它们的记忆能力。然后，我们对所有数据集取最大值来近似模型的能力。为了实例化我们的数据集，每个标记都是从一组预定义的标记中均匀采样的，而与之前的标记无关。

为了近似 $H^k(x \mid \theta, \hat{\theta})$，我们可以直接在训练好的模型下计算熵，以计算在$\theta$ 条件下数据集的最短描述。减去这两个值，我们可以近似未意的记忆 $\text{mem}_U(X, L(X))$. 由于采样数据的过程是完全随机的，θ 中没有泛化需要存储（也就是说，$\text{mem}^U(X, L(X)) \approx \text{mem}(X, L(X))$）。

观察到当我们从均匀分布中采样合成序列时，我们可以精确地计算它们的香农信息。给定数据集大小 N，我们构建一个包含 N 序列的数据集，每个序列包含 S 个标记。给定词汇量大小 V，我们可以通过 $H(x^i) = NS \log_2 V$ 计算具有此类参数的数据集 x^i 的总熵。然后我们使用熵在θ_i 下计算压缩形式 x^i，以计算代码长度，并将其用作 $H^K(x^i \mid \hat{\theta}_j)$ 的近似值。然后我们计算 $\text{mem}(x^i, \hat{\theta}_i) = H(x^i) - H^K(x^i \mid \hat{\theta}_j)$ 并将模型的容量计算为所有数据集上记忆的最大量。

实验细节。 根据 Kaplan 等人 (2020)，我们使用从头开始初始化的 GPT-2 架构 (Radford 等人 , 2019) 训练模型。我们的模型有 1 到 8 层，隐藏维度从 32 缩放到 512，参数从 $100K$ 到 $20M$ 。我们使用 10_6 步和 2048 的批处理大小训练模型。我们使用 Adam 优化器。所有模型都在单个 A100 GPU 上以 bfloat16 精度训练，如果批处理无法适应内存，则使用梯度累积。除非另有说明，我们设置词汇量大小 $V = 2048$，序列长度 $S = 64$ ，并仅更改数据集中的点数。我们对每个模型在每个数据集大小上使用五个随机种子进行训练，这些种子会影响模型初始化和数据集采样。

结果。 我们在图 1 中绘制了跨模型和数据大小的记忆情况。这使我们能够可视化意外记忆的量（ y 轴）随数据集大小（ x 轴）的变化，并按模型大小（线颜色）分组。我们观察到，一旦模型达到其容量，就会出现一个惊人的平台期。考虑到数据集足够大，模型在净记忆方面表现出上限，无论数据大小如何。小数据集会被所有具有足够容量的模型完全记忆。

我们将每个模型的容量估计为在所有数据集大小中测量的最大意外记忆量（以比特为单位）。然后，我们在图 6 中将此容量与模型大小进行比较。有趣的是，即使在这个小尺度上，我们也观察到观察到的容量（在所有数据集中测量的最大记忆量）与模型参数之间非常平滑的关系。我们在图 6 中绘制了这种关系：在这些设置下，

our models consistently memorize between 3.5 and 3.6 bits per parameter. This corroborates the findings of prior work such as (Roberts et al., 2020; Lu et al., 2024), which noticed that fact storage scales linearly with model capacity. Ours is a slightly larger estimate than Allen-Zhu & Li (2024), which estimated via quantization that models can store around 2 bits per parameter.

Since our models are learned via gradient descent, they are not guaranteed to find the global optima; thus, we are only ever measuring a lower bound on model capacity. We take a closer look at the training curves to analyze the convergence of our 8M parameter language model. We plot model convergence throughout training in Figure 5.

In this case, all datasets from 16,000 to 4M samples fall within a range of $3.56 - 3.65 \times 10^6$ bits memorized. This indicates that our measurements are robust within an order of magnitude, and we do not expect to memorize significantly more information by training for more steps. This finding also confirms our hypothesis that capacity scales roughly with parameter count. The two largest datasets (4M and 8M samples, respectively) converge to total memorization of 2.95×10^6 and 1.98×10^6 bits memorized. We expect that their memorization rates would continue to increase toward the capacity had we trained for more epochs.

How does precision affect capacity? One natural question is how our estimates for α depend on the precision of language model training. In fact, although most software defaults to training in 32-bit precision, recent work has shown that language models can be quantized to fewer than 2 bits per parameter and still retain much of their utility. Since all other experiments have been conducted in bfloat16 precision, rerun our experiments in full fp32 precision to analyze the effect on capacity. Across model sizes, we observe a small increase in capacity, and an increase in α from 3.51 to 3.83 bits-per-parameter on average. This is far less than the actual 2x increase in the bits of θ, indicating that **most of the extra model bits added when increasing precision from bfloat16 to float32 are not used** for raw storage.

4 Disentangling Unintended Memorization from Generalization

Our previous experiments analyzed the memorization and membership inference properties of synthetic bitstrings. We now turn to measuring memorization of text. Unlike randomly generated sequences, learning from text data is a mix of both unintended memorization (sample-level) and generalization (population-level). Therefore, as a reference model, we use the model of an equal parameter count trained on the maximum amount of data (in this case, the entire dataset).[3] We also consider an *oracle* reference model, which is the model that achieves the best compression rate (lowest loss) on the evaluation dataset, and may have many more parameters.

Experimental details. We repeat the experiments from Section 3.2, substituting our synthetic datapoints for real text. To obtain a distribution of real-world text data, we could use any pre-training scale text dataset; we use the recently proposed FineWeb dataset (Penedo et al., 2024) as it follows state-of-the-art deduplication practices. We use sequences of 64 tokens but perform an additional deduplication step to ensure perfect deduplication (otherwise, that $1 - 2\%$ of sequences become duplicates when truncating to 64 tokens). We find careful deduplication extremely important for faithfully measuring extraction rates. As in the previous subsection, we pretrain models of varying sizes on different-sized text datasets and measure the unintended memorization of each model-dataset pair. In addition to memorization, we measure membership inference performance according to a standard loss-based membership inference procedure; we also compute exact extraction rates by greedily decoding prefixes of different lengths.

Results. We first observe that the sample-level unintended memorization increases with model parameters and decreases with training set size (Figure 4). When we measure unintended memorization with respect to an oracle reference model (Figure 2), memorization steadily increases as our smaller model is able to learn more about the small training set than the oracle, and then decreases as our model starts to generalize and perform on average worse than the (higher-capacity) oracle.

[3]Restricting the computational power of the reference model relates its predictions to $\mathcal{V}$-information (Xu et al., 2020) which measures the "usable" information available in a signal, when accounting for model size.

我们的模型始终能够记住 3.5 到 3.6 每参数位。这证实了先前工作的发现，例如 (Roberts 等人。， 2020；Lu 等人。， 2024)，他们注意到事实存储与模型容量呈线性关系。我们的估计略大于 Allen-Zhu &Li (2024)，他们通过量化估计模型可以存储大约 2 每参数位。

由于我们的模型通过梯度下降学习，它们不保证找到全局最优解；因此，我们仅测量模型容量的下限。我们仔细查看训练曲线，分析我们 8M 参数语言模型的收敛性。我们在图 5.

在这种情况下，所有从 16,000 到 4M 样本的数据集都落在 $3.56 - 3.65 \times 10^6$ 位记忆范围内。这表明我们的测量在数量级上具有鲁棒性，我们预计通过训练更多步数不会记住更多信息。这一发现也证实了我们的假设，即容量大致与参数数量成正比。两个最大的数据集（ 4M 和 8M 样本，分别）收敛到总共记忆 2.95×10^6 和 1.98×10^6 位的记忆。我们预计，如果训练更多轮次，它们的记忆率将继续向容量增加。

精度如何影响容量？ 一个自然的问题是我们的 α 估计如何取决于语言模型训练的精度。事实上，尽管大多数软件默认以 32 位精度进行训练，但最近的研究表明，语言模型可以量化为每个参数少于 2 位，并且仍然保留其大部分效用。由于所有其他实验都是在 bfloat16 精度下进行的，因此以完整的 fp32 精度重新运行我们的实验，以分析对容量的影响。在所有模型尺寸中，我们观察到容量有少量增加，并且 α 平均从 3.51 增加到 3.83 位 / 参数。这与 θ 的实际 2 倍位增加相去甚远，表明**当从 bfloat16 增加到 float32 时，增加的大部分额外模型位并没有用于原始存储**。

4 将无意记忆与泛化分离

我们的先前实验分析了合成位串的记忆和成员推理特性。我们现在转向测量文本的记忆。与随机生成的序列不同，从文本数据中学习是无意记忆（样本级）和泛化（群体级）的混合。因此，作为一个参考模型，我们使用在最大数据量（在这种情况下，整个数据集）上训练的参数数量相等的模型 [3] 。我们还考虑一个预言机参考模型，该模型在评估数据集上实现了最佳压缩率（最低损失），并且可能具有更多参数。

实验细节。 我们重复了 3.2 节的实验，将我们的合成数据点替换为真实文本。为了获得真实世界文本数据的分布，我们可以使用任何预训练规模文本数据集；我们使用最近提出的 FineWeb 数据集（ Penedo 等人，2024 ）作为它遵循最先进的去重实践。我们使用 64 个 token 的序列，但执行额外的去重步骤以确保完美去重（否则，当截断到 64 个 token 时，那 $1 - 2\%$ 的序列会变成重复）。我们发现仔细去重对于忠实地测量提取率非常重要。如前一子节所述，我们在不同大小的文本数据集上预训练不同大小的模型，并测量每个模型 - 数据集对的意外记忆。除了记忆之外，我们根据标准的基于损失的成员推理程序测量成员推理性能；我们还通过贪婪地解码不同长度的前缀来计算精确的提取率。

结果。 我们首先观察到，样本级别的意外记忆随着模型参数的增加而增加，随着训练集大小的增加而减少（图 4 ）。当我们相对于一个神谕参考模型（图 2 ）测量意外记忆时，记忆稳步增加，因为我们的较小模型能够比神谕学习更多关于小训练集的信息，然后随着我们的模型开始泛化并平均表现不如（更高容量的）神谕而减少。

[3]限制参考模型的计算能力将其预测与 $\mathcal{V}$- 信息（ Xu et al.， 2020 ）相关，该信息衡量了在考虑模型大小的情况下信号中 " 可用 " 的信息量。

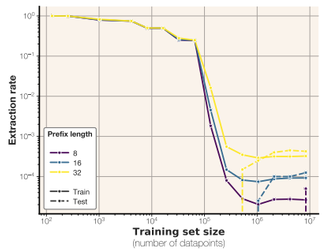

Figure 8 Extraction rates of 64-token training sequences across prefix lengths, for both train and evaluation.

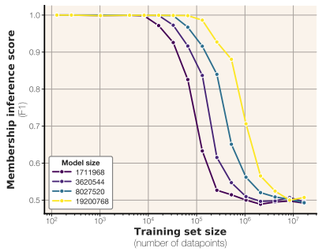

Figure 9 Membership inference F1 across dataset sizes. In this case, F1 score of 0.5 implies random guessing.

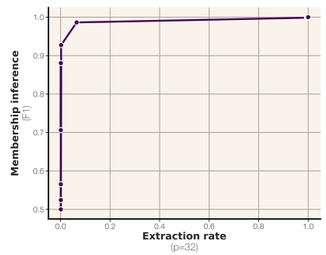

Figure 10 Membership inference vs 32-token-prefix suffix extraction rate. Membership inference is generally easier than extraction.

Dataset-to-capacity ratio predicts double descent. We observe from the train and test loss that for larger datasets the model only begins to generalize (i.e. evaluation loss decreases) once its capacity is reached, which takes approximately 10^5 samples, depending on parameter count. As in Nakkiran et al. (2019) we plot the ratio between the dataset size and model capacity (Figure 3). Unlike prior work, in our experiments we can compute the exact dataset size (based on the compression rates of the reference model) and exact model capacity (based on our estimate of α).

We clearly observe double descent evaluation performance decreases as the training set size nears model capacity, and then rapidly drops as the dataset capacity exceeds the capacity of the model. Our observations offer an intuitive explanation for double descent (Belkin et al., 2019; Nakkiran et al., 2019): **double descent begins exactly when the data capacity exceeds the model capacity**. One theory is that once the model can no longer memorize datapoints individually, it is forced to share information between datapoints to save capacity, which leads to generalization.

Generalization explains nonzero extraction rates. We measure extraction rates on the full training set and 10,000 non-overlapping test samples (Figure 17). We note that for 32-token prefixes, 100% are extractable for very small training set sizes; predictably, all extraction numbers decrease with training set size. When the dataset sizes grows sufficiently large, the extraction rate does not go fully to zero; however, it converges to nearly exactly the test extraction rate. In other words, when our (deduplicated) dataset grows sufficiently large, **all successful training data extraction is attributable to generalization**.

5 Memorization and Membership

Our training settings allow total control over the train and test data and come with perfect deduplication. This makes our setting ideal for studying the relationship between model size, dataset size, and membership inference success rate.

All of our membership inference results come from a standard loss-based membership inference (Yeom et al., 2018; Sablayrolles et al., 2019). The method is very simple: we set a cutoff loss value to predict whether a sample is or is not a member of the training dataset.

5.1 Membership in synthetic and text data

Synthetic data. For each of our models trained on synthetic data, we plot the success rate of the membership inference attack attack across dataset sizes. We show results in Figure 14. Above a certain dataset size, membership inference starts to fail in the average case. This finding indicates that if the dataset size is too large compared to the model, membership inference of an average training sample may not be possible.

Text. For each of our models trained on text, we use unused non-overlapping data from FineWeb to perform a standard loss-based membership inference (Yeom et al., 2018; Sablayrolles et al., 2019) on each model and

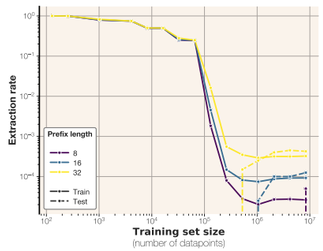

图 8 64- token 训练序列在不同前缀长度下的提取率，包括训练集和评估集。

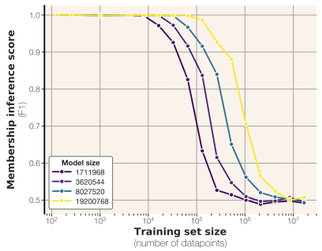

图 9 成员推理 F1 在不同数据集大小下的表现。在这种情况下，F1 分数为 0.5 表示随机猜测。

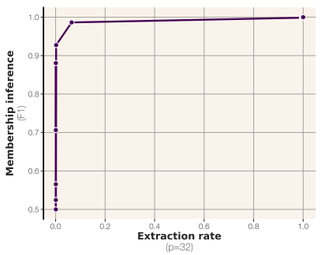

图 10 成员推理与 32-token- 前缀后缀提取率的对比。成员推理通常比提取更容易。

数据集与容量比率预测双重下降。 我们从训练和测试损失中观察到，对于较大的数据集，模型只有在其容量达到时才开始泛化（即评估损失下降），这大约需要 10_5 个样本，具体取决于参数数量。如 Nakkiran 等人 (2019) 所示，我们绘制了数据集大小与模型容量之间的比率（图 3 ）。与先前的工作不同，在我们的实验中，我们可以根据参考模型的压缩率计算确切的 数据集大小，并根据我们对 α 的估计计算确切的模型容量。

我们清楚地观察到，随着训练集大小接近模型容量，双重下降的评估性能会下降，而当数据集容量超过模型容量时，性能会迅速下降。我们的观察为双重下降提供了直观的解释（ Belkin 等人 , 2019; Nakkiran 等人 , 2019）： **双重下降开始的确切时刻是数据容量超过模型容量**。一种理论是，一旦模型无法单独记住数据点，它就会被迫在数据点之间共享信息以节省容量，这导致了泛化。

泛化解释了非零提取率。 我们在完整训练集和 10,000 个不重叠的测试样本上测量提取率（图 17 ）。我们注意到，对于 32 个 token 的前缀，在非常小的训练集大小下，100% 都是可提取的；可预见地，所有提取数字随着训练集大小减小。当数据集大小足够大时，提取率不会完全变为零；然而，它会收敛到几乎精确的测试提取率。换句话说，当我们的（去重）数据集足够大时，**所有成功的训练数据提取都归因于泛化**。

5 记忆和成员资格

我们的训练设置允许完全控制训练和测试数据，并带有完美的去重。这使得我们的设置非常适合研究模型大小、数据集大小和成员资格推理成功率之间的关系。

我们所有的成员资格推理结果都来自标准的基于损失的成员资格推理（ Yeom 等人 ,2018 ； Sablayrolles 等人， 2019 ）。该方法非常简单：我们设置一个截止损失值来预测一个样本是否是训练数据集的成员。

5.1 合成数据中的成员资格

合成数据。 对于我们在合成数据上训练的每个模型，我们绘制了成员资格推理攻击的成功率，跨越数据集大小。我们在图 14 中显示结果。在超过某个数据集大小后，成员资格推理在平均情况下开始失败。这一发现表明，如果数据集大小相对于模型过大，平均训练样本的成员资格推理可能无法进行。

文本。 对于我们在文本上训练的每个模型，我们使用来自 FineWeb 的未使用且不重叠的数据进行标准的基于损失的成员资格推理（ Yeom 等人， 2018 ； Sablayrolles 等人， 2019 ）在每个模型上进行，并

	d_{emb}	n_{layer}	$\|\theta\|$	$\|D\|$	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.

plot performance across dataset sizes (9). For a fixed model size, membership inference gets more difficult as the size of the data increases. When comparing membership inference to extraction (Figure 10), membership inference is strictly higher in every case; in some cases we can infer training dataset membership quite well (score of 0.97) with an extraction rate of 0.

5.2 Scaling laws for Membership

In this section we develop a set of predictive models for memorization. Specifically, we predict the F1 score of a loss-based membership attack given token count, number of examples, and model parameter count. We then validate our predictions on models from 500K to 1.5B parameters.

5.2.1 Functional forms

We observe that for a fixed model capacity, membership inference follows a roughly sigmoidal form with respect to dataset size. The intuitive explanation is that M.I. is easy for large models overfit to tiny datasets, so its score begin at 1; as dataset size increases, differentiating train from test data by loss becomes more and more difficult, eventually decaying toward 0.5.

We reuse the data collected in our text experiments (Section 4) to solve for constants c_1, c_2, c_3 in the following equation:

$$\text{Membership}_{F_1}(\theta, \mathcal{D}) = \frac{1}{2}(1 + c_1\sigma(c_2(\frac{\text{Capacity}(\theta)}{|\mathcal{D}|} + c_3))$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$.

Limiting behavior. We observe that as $|\mathcal{D}| \to \infty$, performance of our membership inference attack decreases to 0.5 (essentially random performance). For a model trained on an infinite dataset, our law predicts both membership inference and extraction to be impossible.

Fitting. We use a non-linear least squares solver to find optimal values for c_1, c_2, c_3. Solutions found are $c_1 = 1.34$, $c_2 = -0.034$, and -33.14. We plot the scaling laws along with observed data in Figure 7. Although the sigmoidal function is slightly simplistic (the points do not perfectly fit) our fit produces estimates within $1 - 2\%$ of observations.

5.2.2 Validation on larger models

We note that all contemporary language models trained with a tokens-per-parameter ratio of 10^2 or higher, which according to our laws would imply membership inference score of 0.5 – that is, within our formulation, statistically significant loss-based membership inference is not possible.

To validate our predictions, we train models with expected membership $F1$ scores of 0.55, 0.75, and 0.95. For model sizes we select GPT-2 small (125M params) and GPT-2 XL (1.5B params). Using our scaling law, we solve for the dataset size required to get the desired membership inference score for the given model size (see Table 2 for more information). We train models on the estimated dataset size and measure F1 score (shown as circles in Figure 7).

	d_{emb}	n_{layer}	$\|\theta\|$	$\|D\|$	预测 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.

在数据集大小上绘制性能（ 9 ）。对于固定的模型大小，随着数据大小的增加，成员资格推理变得更加困难。当比较成员资格推理与提取（图 10 ）时，成员资格推理在每种情况下都严格更高；在某些情况下，我们可以很好地推断训练数据集的成员资格（分数为 0.97 ）并且提取率为 0。

5.2 成员资格的缩放定律 p

在本节中，我们开发了一套用于记忆的预测模型。具体来说，我们根据标记计数、示例数量和模型参数数量预测基于损失的成员资格攻击的 F1 分数。然后，我们在 500K 到 1.5B 参数的模型上验证我们的预测。

5.2.1 函数形式

我们观察到，对于固定的模型容量，成员推理随着数据集大小大致呈 S 形。直观的解释是，对于过度拟合于微小数据集的大型模型，M.I. 很容易，因此其分数从 1 开始；随着数据集大小的增加，通过损失区分训练数据和测试数据变得越来越困难，最终衰减到 0.5。

我们重用了我们在文本实验中收集的数据（第 4 节）来解决以下方程中的常数 c_1, c_2, c_3 ：

$$\text{Membership}_{F_1}(\theta, \mathcal{D}) = \frac{1}{2}(1 + c_1\sigma(c_2(\frac{\text{Capacity}(\theta)}{|\mathcal{D}|} + c_3))$$

其中 $\sigma(x)=$ $^{1}1+e^{-x}$ 。

极限行为。 我们观察到，随着 $|\mathcal{D}| \to \infty$，我们成员推理攻击的性能下降到 0.5 （本质上随机性能）。对于一个在无限数据集上训练的模型，我们的定律预测成员推理和提取都是不可能的。

拟合。 我们使用非线性最小二乘求解器来找到 c_1, c_2, c_3 的最优值。找到的解是 $c_1 = 1.34$, $c_2 = -0.034$，和 -33.14。我们将缩放定律与观测数据一起绘制在图 7。尽管 S 形函数有些过于简单（点并不完全拟合），但我们的拟合产生的估计值在观测值的 $1 - 2\%$ 范围内。

5.2.2 在更大模型上的验证

我们注意到，所有当代语言模型在每参数 10_2 或更高的 token-per-parameter 比率下进行训练，根据我们的定律，这会意味着成员资格推理分数为 0.5 —— 也就是说，在我们的公式中，统计上显著的基于损失的成员资格推理是不可能的。

为了验证我们的预测，我们训练了预期成员资格 $F1$ 分数为 0.55 、 0.75 和 0.95 的模型。对于模型大小，我们选择了 GPT-2 小（ 125M 参数）和 GPT-2 XL （ 1.5B 参数）。使用我们的缩放定律，我们求解了为给定模型大小获得所需成员资格推理分数所需的数据集大小（参见表 2 以获取更多信息）。我们在估计的数据集大小上训练模型，并测量 F1 分数（如图 7 中的圆圈所示）。

Our predictions are generally within 1.5 points of the true F1 score; the score is most inaccurate for estimated F1 of 0.75, which is the point where the sigmoid is steepest. In general, the accuracy of our results indicates that our empirical model of membership inference is relatively accurate and provides evidence for why membership inference attacks fail on models trained on extremely large datasets (Das et al., 2024; Duan et al., 2024; Maini et al., 2024).

6 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 Section A.2.

7 Conclusion

We propose a new definition of memorization that allows us to measure the exact number of bits a model knows about a dataset. We use our definition to measure the capacity of modern transformer language models and analyze how measurements such as extraction and F1 score scale with model and dataset size. We also propose a scaling law for membership inference and validate it on larger models. Our results help further practitioner understanding of how language models memorize and what they might (or might not) be memorizing across model and dataset scales.

8 Acknowledgements

Thanks to the many folks who helped us improve our paper, including Karen Ullrich, Niloofar Mireshghallah, Mark Ibrahim, Preetum Nakkiran, and Léon Bottou.

References

Allen-Zhu, Z. and Li, Y. Physics of language models: Part 3.3, knowledge capacity scaling laws, 2024. URL https://arxiv.org/abs/2404.05405.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., and Lacoste-Julien, S. A closer look at memorization in deep networks, 2017. URL https://arxiv.org/abs/1706.05394.

我们的预测通常在 1.5 真实 F1 分数的 点以内；分数在最不准确的地方是估计的 F1 为 0.75，这是 sigmoid 最陡峭的地方。总的来说，我们结果的准确性表明我们的成员推理经验模型相对准确，并为为什么成员推理攻击在训练于极大规模数据集的模型上失败提供了证据（ Das 等人。， 2024； Duan 等人。， 2024;Maini 等人。， 2024).

6 相关工作

语言模型和压缩。 香农的信源编码定理（香农， 1948 ）首次形式化了预测和压缩之间的对偶性。语言建模与压缩之间的联系早在 香农 (1950)，就被观察到更准确的英语模型可以用更少的比特压缩文本。其他工作详细讨论了柯尔莫哥洛夫复杂度（柯尔莫哥洛夫， 1965) 和香农信息之间的联系（ Grunwald & Vitanyi,2004)。 Delétang 等人 (2024) 研究使用现代基于 transformer 的语言模型作为压缩器。我们使用压缩作为衡量模型记忆的工具。

语言模型能力。(Arpit 等人，2017) 将模型的有效能力及其训练过程形式化；他们还观察到，表示能力和训练时间都对经验模型能力有很强的影响。其他一些工作通过网络（如 RNN （ Collins 等人，2017； Boo 等人，2019 ）或 transformer （ Roberts 等人，2020； Heinzerling & Inui，2021； Allen-Zhu & Li，2024 ）能够记忆的事实或随机标签的数量来衡量语言模型能力，有时在量化下。一些研究工作（ Yun 等人，2019； Curth 等人，2023； Mahdavi 等人，2024； Kajitsuka & Sato，2024 ）已经为不同模型架构的能力开发了理论估计，尽管还没有扩展到多层现代 transformer。我们是第一个测量模型能力明确上限的。

记忆的替代定义。 意外记忆与文献中提出的其他记忆定义密切相关。我们在第 A.2 节中进行了详细比较。

7 结论

我们提出了一个新的记忆定义，使我们能够测量模型对数据集了解的确切比特数。我们使用该定义来测量现代 Transformer 语言模型的容量，并分析提取和 F1 分数等指标如何随模型和数据集规模变化。我们还提出了成员推理的缩放规律，并在更大的模型上验证了它。我们的结果有助于进一步加深从业者对语言模型如何记忆以及它们在不同模型和数据集规模下可能（或不可能）记忆内容的理解。

8 致谢

感谢许多 folks 帮助我们改进论文，包括 Karen Ullrich 、 Niloofar Mireshghallah 、 Mark Ibrahim 、 Preetum Nakkiran 和 Léon Bottou。

参考文献

Allen-Zhu, Z. and Li, Y. 语言模型的物理：第 3.3 部分，知识容量缩放定律，2024。 URLhttps://arxiv.org/abs/2404.05405.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., and Lacoste-Julien, S. 深度网络中的记忆研究，2017。URLhttps://arxiv.org/abs/1706.05394.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, July 2019. ISSN 1091-6490. doi: 10.1073/pnas.1903070116. URL http://dx.doi.org/10.1073/pnas.1903070116.

Bhattacharjee, R., Dasgupta, S., and Chaudhuri, K. Data-copying in generative models: a formal framework. In *International Conference on Machine Learning*, pp. 2364–2396. PMLR, 2023.

Boo, Y., Shin, S., and Sung, W. Memorization capacity of deep neural networks under parameter quantization, 05 2019.

Carlini, N., Liu, C., Úlfar Erlingsson, Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks, 2019. URL https://arxiv.org/abs/1802.08232.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramèr, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models, 2023a. URL https://arxiv.org/abs/2301.13188.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models, 2023b. URL https://arxiv.org/abs/2202.07646.

Cohen, E., Kaplan, H., Mansour, Y., Moran, S., Nissim, K., Stemmer, U., and Tsfadia, E. Data reconstruction: When you see it and when you don't, 2024. URL https://arxiv.org/abs/2405.15753.

Collins, J., Sohl-Dickstein, J., and Sussillo, D. Capacity and trainability in recurrent neural networks, 2017. URL https://arxiv.org/abs/1611.09913.

Curth, A., Jeffares, A., and van der Schaar, M. A u-turn on double descent: Rethinking parameter counting in statistical learning, 2023. URL https://arxiv.org/abs/2310.18988.

Das, D., Zhang, J., and Tramèr, F. Blind baselines beat membership inference attacks for foundation models, 2024. URL https://arxiv.org/abs/2406.16201.

Delétang, G., Ruoss, A., Duquenne, P.-A., Catt, E., Genewein, T., Mattern, C., Grau-Moya, J., Wenliang, L. K., Aitchison, M., Orseau, L., Hutter, M., and Veness, J. Language modeling is compression, 2024. URL https://arxiv.org/abs/2309.10668.

Duan, M., Suri, A., Mireshghallah, N., Min, S., Shi, W., Zettlemoyer, L., Tsvetkov, Y., Choi, Y., Evans, D., and Hajishirzi, H. Do membership inference attacks work on large language models? In *Conference on Language Modeling (COLM)*, 2024.

Dubey, A. and et al, A. J. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Dwork, C. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pp. 1–12. Springer, 2006.

Feldman, V. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 954–959, 2020.

Feldman, V. and Zhang, C. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

Geiping, J., Stein, A., Shu, M., Saifullah, K., Wen, Y., and Goldstein, T. Coercing llms to do and reveal (almost) anything, 2024. URL https://arxiv.org/abs/2402.14020.

Grunwald, P. and Vitányi, P. Shannon information and kolmogorov complexity. *arXiv preprint cs/0410002*, 2004.

Grunwald, P. and Vitanyi, P. Shannon information and kolmogorov complexity, 2004. URL https://arxiv.org/abs/cs/0410002.

Heinzerling, B. and Inui, K. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries, 2021. URL https://arxiv.org/abs/2008.09036.

Jayaraman, B. and Evans, D. Are attribute inference attacks just imputation? In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1569–1582, 2022.

Kajitsuka, T. and Sato, I. Optimal memorization capacity of transformers, 2024. URL https://arxiv.org/abs/2409.17677.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, July 2019. ISSN 1091-6490. doi: 10.1073/pnas.1903070116. URL http://dx.doi.org/10.1073/pnas.1903070116.

Bhattacharjee, R., Dasgupta, S., 和 Chaudhuri, K. 生成模型中的数据复制：一个形式化框架。I n *International Conference on Machine Learning*, pp. 2364–2396. PMLR, 2023.

Boo, Y., Shin, S. 和 Sung, W. 参数量化下深度神经网络的记忆能力 , 2019. 5

Carlini, N., Liu, C., Úlfar Erlingsson, Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks, 2019. URL https://arxiv.org/abs/1802.08232.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramèr, F., Balle, B., Ippolito, D., and Wallace, E. 从扩散模型中提取训练数据 , 2023a. URL https://arxiv.org/abs/2301.13188.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. 跨神经语言模型的记忆量化 , 2023b. URL https://arxiv.org/abs/2202.07646.

Cohen, E., Kaplan, H., Mansour, Y., Moran, S., Nissim, K., Stemmer, U., and Tsfadia, E. 数据重建：当你看 n 到它时和当你看不到它时 , 2024. URL https://arxiv.org/abs/2405.15753.

Collins, J., Sohl-Dickstein, J., and Sussillo, D. 容量和可训练性在循环神经网络中 , 2017. URLhttps://arxiv.org/abs/1611.09913.

Curth, A., Jeffares, A., and van der Schaar, M. A u-turn on double descent: 重新思考统计学习中的参数计数 , 2023. URL https://arxiv.org/abs/2310.18988.

Das, D., Zhang, J., and Tramèr, F. 盲基线击败了基础模型的成员推理攻击 , 2024. URL https://arxiv.org/abs/2406.16201.

Delétang, G., Ruoss, A., Duquenne, P.-A., Catt, E., Genewein, T., Mattern, C., Grau-Moya, J., Wenliang, L. K., Aitchison, M., Orseau, L., Hutter, M., and Veness, J. 语言建模是压缩 , 2024. URL https://arxiv.org/abs/2309.10668.

Duan, M., Suri, A., Mireshghallah, N., Min, S., Shi, W., Zettlemoyer, L., Tsvetkov, Y., Choi, Y., Evans, D., 和 Hajishirzi, H. 大型语言模型的成员推理攻击是否有效？在语言建模会议 *(COLM)*, 2024 年。

Dubey, A. 和 et al, A. J. Llama 3 模型群 , 2024 年 . URL https://arxiv.org/abs/2407.21783.

Dwork, C. 差分隐私 . 在 自动化、语言和编程国际研讨会 , 第 1–1 页 2. Springer, 2006 年。

Feldman, V. 学习是否需要记忆？一个关于长尾的短故事。在 第 *52* 届 *ACM SIGACT* 计算理论研讨会论文集 , 第 954–959 页 , 2020 年。

Feldman, V. 和 Zhang, C. 神经网络记忆什么以及为什么：通过影响估计发现长尾。神经信息处理系统进展 , 33:2881–2891, 2020。

Geiping, J., Stein, A., Shu, M., Saifullah, K., Wen, Y., 和 Goldstein, T. 强迫 llms 做和揭示（几乎）任何事情, 2024。URL https://arxiv.org/abs/2402.14020。

Grunwald, P. 和 Vitányi, P. 香农信息和柯尔莫哥洛夫复杂度。 *arXiv* 预印本 *cs/0410002*, 2004。

Grunwald, P. 和 Vitanyi, P. 香农信息和柯尔莫哥洛夫复杂度，2004。URLhttps://arxiv.org/abs/c s/0410002.<

海因策林，B. 和 井内，K. 语言模型作为知识库：关于实体表示、存储容量、一个 d 释义查询，2021. URL https://arxiv.org/abs/2008.09036.

贾亚拉曼，B. 和 伊万斯，D. 属性推理攻击是否只是插补？在 *2022* 年 *ACM SIGSAC* 计算机与通信安全会议论文集，第 1569-1582 页，2022 年。

卡吉茨卡，T. 和 佐藤，I. 变换器的最优记忆容量，2024. URL https://arxiv.org/abs/2409.17677.

卡普兰，J.、麦卡尼斯，S.、海尼汉，T.、布朗，T. B.、切斯，B.、 Child，R.、格雷，S.、拉德福德，A.、吴，d J. 和 阿莫迪，D. 神经语言模型的规模定律，2020. URL https://arxiv.org/abs/2001.08361.

12

Kolmogorov, A. N. On tables of random numbers. *Sankhyā: The Indian Journal of Statistics, Series A*, 25(4):369–376, 1963. URL http://www.jstor.org/stable/25049284. Accessed: 21/12/2010 15:32.

Kolmogorov, A. N. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. Deduplicating training data makes language models better, 2022. URL https://arxiv.org/abs/2107.06499.

Liu, K. Z., Choquette-Choo, C. A., Jagielski, M., Kairouz, P., Koyejo, S., Liang, P., and Papernot, N. Language models may verbatim complete text they were not explicitly trained on, 2025. URL https://arxiv.org/abs/2503.17514.

Lu, X., Li, X., Cheng, Q., Ding, K., Huang, X., and Qiu, X. Scaling laws for fact memorization of large language models, 2024. URL https://arxiv.org/abs/2406.15720.

Mahdavi, S., Liao, R., and Thrampoulidis, C. Memorization capacity of multi-head attention in transformers, 2024. URL https://arxiv.org/abs/2306.02010.

Maini, P., Jia, H., Papernot, N., and Dziedzic, A. Llm dataset inference: Did you train on my dataset?, 2024. URL https://arxiv.org/abs/2406.06443.

Mireshghallah, F., Uniyal, A., Wang, T., Evans, D., and Berg-Kirkpatrick, T. Memorization in nlp fine-tuning methods, 2022. URL https://arxiv.org/abs/2205.12506.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt, 2019. URL https://arxiv.org/abs/1912.02292.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., and Lee, K. Scalable extraction of training data from (production) language models, 2023. URL https://arxiv.org/abs/2311.17035.

Pasco, R. C. *Source coding algorithms for fast data compression*. Ph.d. dissertation, Stanford University, 1977.

Penedo, G., Kydlíček, H., allal, L. B., Lozhkov, A., Mitchell, M., Raffel, C., Werra, L. V., and Wolf, T. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

Prashanth, U. S., Deng, A., O'Brien, K., V, J. S., Khan, M. A., Borkar, J., Choquette-Choo, C. A., Fuehne, J. R., Biderman, S., Ke, T., Lee, K., and Saphra, N. Recite, reconstruct, recollect: Memorization in lms as a multifaceted phenomenon, 2024. URL https://arxiv.org/abs/2406.17746.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Rissanen, J. Generalized kraft inequality and arithmetic coding. *IBM Journal of Research and Development*, 20(3): 198–203, 1976.

Roberts, A., Raffel, C., and Shazeer, N. How much knowledge can you pack into the parameters of a language model?, 2020. URL https://arxiv.org/abs/2002.08910.

Sablayrolles, A., Douze, M., Ollivier, Y., Schmid, C., and Jégou, H. White-box vs black-box: Bayes optimal strategies for membership inference, 2019. URL https://arxiv.org/abs/1908.11229.

Schwarzschild, A., Feng, Z., Maini, P., Lipton, Z. C., and Kolter, J. Z. Rethinking llm memorization through the lens of adversarial compression, 2024. URL https://arxiv.org/abs/2404.15146.

Shannon, C. E. *A Mathematical Theory of Communication*. University of Illinois Press, 1948. Reprint in 1998.

Shannon, C. E. Prediction and entropy of printed english, Sept 1950.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Tänzer, M., Ruder, S., and Rei, M. Memorisation versus generalisation in pre-trained language models, 2022. URL https://arxiv.org/abs/2105.00828.

Xia, M., Artetxe, M., Zhou, C., Lin, X. V., Pasunuru, R., Chen, D., Zettlemoyer, L., and Stoyanov, V. Training trajectories of language models across scales, 2023. URL https://arxiv.org/abs/2212.09803.

Xu, Y., Zhao, S., Song, J., Stewart, R., and Ermon, S. A theory of usable information under computational constraints, 2020. URL https://arxiv.org/abs/2002.10689.

柯尔莫哥洛夫，A. N. 关于随机数表的论文。 *Sankhy¯a: 印度统计杂志，A 系列*，25(4):369–376，1963。URLhttp://www.jstor.org/stable/25049284。访问时间：21/12/2010 15:32。

柯尔莫哥洛夫，A. N. 关于信息定量定义的三种方法。 信息传输问题 ,1(1):1–7，1965。

李，K.，伊波利托，D.，尼斯特罗姆，A.，张，C.，埃克，D.，卡利斯昂 - 伯奇，C.，和卡林尼，N. 去重训练数据使语言模型更好，2022。URL https://arxiv.org/abs/2107.06499。

刘，K. Z.，乔奎特 - 丘，C. A.，贾吉尔斯基，M.，卡伊鲁兹，P.，科耶乔，S.，梁，P.，和帕珀诺特，N. 语言模型可s能逐字完成他们未明确训练的文本，2025。URL https://arxiv.org/abs/2503.17514。

陆晓，李晓，程强，丁凯，黄翔，邱翔。大型语言模型的事实记忆缩放定律，2024。 e URLhttps://arxiv.org/abs/2406.15720。

Mahdavi, S., Liao, R., and Thrampoulidis, C. 转换器中多头注意力的记忆能力，2024。URL https://arxiv.org/abs/2306.02010。

Maini, P., Jia, H., Papernot, N., and Dziedzic, A. Llm 数据集推理：你训练了我的数据集吗？，2024。URLhttps://arxiv.org/abs/2406.06443。

Mireshghallah, F., Uniyal, A., Wang, T., Evans, D., and Berg-Kirkpatrick, T. Nlp 微调方法的记忆，2022。URLhttps://arxiv.org/abs/2205.12506。

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. 深度双重下降：更大模型和更多数据带来的伤害，2019. URL https://arxiv.org/abs/1912.02292.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., and Lee, K. 可扩展地从（生产）语言模型中提取训练数据，2023. URLhttps://arxiv.org/abs/2311.17035.

Pasco, R. C. 快速数据压缩的源编码算法 . 博士论文，斯坦福大学，1977.

Penedo, G., Kydlíček, H., allal, L. B., Lozhkov, A., Mitchell, M., Raffel, C., Werra, L. V., and Wolf, T. 细微数b据集：大规模下提取最精细文本数据，2024. URL https://arxiv.org/abs/2406.17557.

Prashanth, U. S., Deng, A., O'Brien, K., V, J. S., Khan, M. A., Borkar, J., Choquette-Choo, C. A., Fuehne, J. R., Biderman, S., Ke, T., Lee, K., and Saphra, N. Recite, reconstruct, recollect: Memorization in lms as a multifaceted phenomenon, 2024. URL https://arxiv.org/abs/2406.17746.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., 和 Sutskever, I. 语言模型是无监督的多任务学习器。2019。

Rissanen, J. Generalized kraft inequality and arithmetic coding. *IBM Journal of Research and Development*, 20(3): 198–203, 1976.

Roberts, A., Raffel, C., and Shazeer, N. How much knowledge can you pack into the parameters of a language model?, 2020. URL https://arxiv.org/abs/2002.08910.

Sablayrolles, A., Douze, M., Ollivier, Y., Schmid, C., and Jégou, H. White-box vs black-box: Bayes optimal strategies for membership inference, 2019. URL https://arxiv.org/abs/1908.11229.

Schwarzschild, A., Feng, Z., Maini, P., Lipton, Z. C., and Kolter, J. Z. Rethinking llm memorization through the lens of adversarial compression, 2024. URL https://arxiv.org/abs/2404.15146.

Shannon, C. E. *A Mathematical Theory of Communication*. University of Illinois Press, 1948. Reprint in 1998.

Shannon, C. E. Prediction and entropy of printed english, Sept 1950.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Tänzer, M., Ruder, S., and Rei, M. 语言模型中的记忆与泛化，2022. URLhttps://arxiv.org/abs/2105.00828.

Xia, M., Artetxe, M., Zhou, C., Lin, X. V., Pasunuru, R., Chen, D., Zettlemoyer, L., and Stoyanov, V. 跨规模语言模型的训练轨迹，2023. URL https://arxiv.org/abs/2212.09803.

Xu, Y., Zhao, S., Song, J., Stewart, R., and Ermon, S. 计算约束下的可用信息理论，2020. URLhttps://arxiv.org/abs/2002.10689.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting, 2018. URL https://arxiv.org/abs/1709.01604.

Yun, C., Sra, S., and Jadbabaie, A. Small relu networks are powerful memorizers: a tight analysis of memorization capacity, 2019. URL https://arxiv.org/abs/1810.07770.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization, 2017. URL https://arxiv.org/abs/1611.03530.

Zhang, C., Ippolito, D., Lee, K., Jagielski, M., Tramèr, F., and Carlini, N. Counterfactual memorization in neural language models, 2023. URL https://arxiv.org/abs/2112.12938.

A Appendix

A.1 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, a concurrent work (Cohen et al., 2024) proposes a theoretical definition for memorization also relying on Kolmogorov.

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.2 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.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting, 2018. URL https://arxiv.org/abs/1709.01604.

Yun, C., Sra, S., and Jadbabaie, A. Small relu networks are powerful memorizers: a tight analysis of memorization capacity, 2019. URL https://arxiv.org/abs/1810.07770.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization, 2017. URL https://arxiv.org/abs/1611.03530.

Zhang, C., Ippolito, D., Lee, K., Jagielski, M., Tramèr, F., and Carlini, N. Counterfactual memorization in neural language models, 2023. URL https://arxiv.org/abs/2112.12938.

A 附录

A.1 相关工作：记忆的定义

先前的记忆定义。Carlini 等人 (2019) 定义了一个字符串 m 被 语言模型 θ 记忆，如果 m 的后半部分可以在用 m 的前半部分提示模型时贪婪地生成。随后，Nasr 等人 (2023) 引入了可提取记忆，其中如果可以找到一个对抗性提示 p 生成 m，则称模型 θ 记忆了 m。Mireshghallah 等人 (2022) 和 Schwarzschild 等人 (2024) 通过将 p 限制为一定数量的 token 来改进这个定义，防止其包含整个 m。然而，即使这个定义也有局限性：例如，用提示 “ 重复 cat 1000 次 ” 生成序列 “cat cat cat ... cat” 并不一定表示记忆。
Carlini 等人 (2019) 使用困惑度或似然性，序列可压缩性的一种度量，试图区分高度记忆的序列和仅仅是易于压缩的序列。另一个值得注意的定义是反事实记忆 (Zhang 等人，2023)，它衡量单个数据点对训练的影响；这可以看作是我们定义的一个实例，其中使用同一系列的另一个模型作为参考模型。总的来说，所有这些工作都以可以看作是压缩形式的方式来看待记忆，尽管没有明确将其定义为这样。

最后，一项并发工作（ Cohen 等人。,2024 ）提出了一个基于 Kolmogorov 的记忆的理论定义。 g

我们的一些发现也涉及机器学习（ Belkin 等人，2019； Nakkiran 等人，2019 ）和语言建模（ Xia 等人，2023 ）中的双重下降现象的发现，以及深度学习中记忆和泛化的普遍讨论（ Zhang 等人，2017； Tänzer 等人，2022 ）。
在这里，我们讨论其他记忆定义。

A.2 其他记忆概念

在本节中，我们列出了多个其他记忆概念，并将其与我们的定义进行比较。我们特别关注这些概念为何不满足我们所有要求。

- **基于稳定性的记忆概念。** 有些隐私和记忆概念处理训练算法对训练集微小变化的 “ 稳定性 ”。最值得注意的是，差分隐私 Dwork (2006) 考虑到当单个数据点变化时模型分布的最坏情况漂移。另一个记忆概念在 Feldman(2020) 中基于模型预测在点 x 上的变化，当我们向分类 / 回归模型的训练集添加标记对 (x, y) 时。这两种概念都关键地依赖于学习算法及其行为。此外，差分隐私的定义对我们的情况并不理想，因为它是一个最坏情况定义，不能在样本 / 模型级别应用。虽然 Feldman (2020) 中的记忆概念没有这个问题，但它存在一个事实，即它仅适用于分类模型，并且主要处理标签（ y ）和输入（ x ）之间的关联记忆，而不是记忆 x 本身。这些问题使得这些概念对我们的情况不理想。

- **基于提取的存储。** 文献中有多个研究（Carlini 等人， 2019 ； Mireshghal-lah 等人， 2022 ； Nasr 等人， 2023 ； Zhang 等人， 2023 ； Carlini 等人， 2023b ； Schwarzschild 等人， 2024 ）将语言模型中样本的存储定义为根据提取该样本的难易程度来定义的。

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.3 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.

具体来说，当试图理解模型 θ 中样本 x 的记忆程度时，他们会测量一些关于任务复杂性的概念，以促使模型输出 x。尽管这些概念在仅需模型 θ 和样本 x 的方面很出色，但它们仍然无法解释泛化能力。以以下训练样本为例："What is 2_{100}? (A: $1, 267, 650, 600, 228, 229, 401, 496, 703, 205, 376$)"，几乎所有的基于提取的记忆概念都会将其识别为高度记忆。这些定义的另一个问题是，它们严重依赖于解码算法的细节。这并不理想，因为我们不期望模型 θ 中样本 x 的记忆程度依赖于我们使用 θ 生成样本时使用的详细参数。

这项工作 Schwarzschild 等人 (2024) 在这一类别中与我们最为接近。这项基于提示优化的工作，优化一个短提示 p 以使模型产生 x，然后它调用样本 x 记忆，如果 p 的长度小于 x。尽管这个定义在使用压缩方面与我们的定义相近，但它仍然没有考虑到模型的泛化能力。此外，它专注于通过提示进行特定方式的压缩。我们认为通过提示进行的压缩是一种较差的压缩方案，并且通常会导致大于 1 的压缩率。

- **成员 / 属性推理。** 成员推理 Shokri 等人 (2017) 和属性推理攻击 Jayaraman & Evans (2022) 已被用于经验性地测量机器学习算法的隐私性。这些通常旨在近似记忆稳定性概念的概念存在相同的缺陷。它们严重依赖于学习算法和数据分布。此外，它们无法提供样本级别的记忆概念。例如，获得的成员推理攻击精度仅在总体级别上有意义。这是因为各种攻击对成员的真正阳性可能不同，而所有这些攻击的真正阳性集合可能覆盖整个训练集，使其无法作为样本级别的记忆概念使用。
- **生成模型中的数据复制。** 有一些针对生成建模设计的有趣记忆概念，其中生成模型可能会输出训练样本的一部分（ Bhattacharjee 等人， 2023 ； Carlini 等人， 2023a ）。这些概念类似于基于提取的记忆定义，但它们更宽松，因为它们只需要提取部分训练数据。然而，它们仍然存在与基于提取的定义相同的挑战。

A.3 基于语言模型的压缩（超越算术编码）

香农 (1948) 指出，对于给定的源，最优的压缩方法是为符号分配代码，使得平均代码长度接近源的熵。算术编码（ Pasco,1977; Rissanen, 1976 ）被认为是给定符号分布时压缩文本的一种最优方法；它在（ Delétang et al., 2024）中被用于使用现代语言模型压缩文本。

尽管算术编码在从随机选择过程中生成的样本中已知是最优的，但在压缩样本与随机过程选择相关的案例中，它可能仍然是次优的。具体来说，在语言建模中，训练数据与模型本身高度相关，因此我们可能需要将它们区别对待。例如，我们从先前的工作中知道，模型在训练数据点上的行为与随机样本不同。大量训练数据可以使用贪婪解码（ Carlini et al., 2023b; Liuet al., 2025 ）生成，这是一种随机采样数据中未预期的行为。为此，我们设计了一种新的压缩技术，即算术编码的推广。

集成压缩。 从语言模型中进行采样涉及两个关键参数 k 用于 top_k 选择和 t 用于温度。我们设计了一种自适应设置这些参数的压缩方法。例如，对于我们知道可以以贪婪方式解码接下来 100 个标记的情况，我们设置 $k = 1$ 来减少算术码的比特长度。改变编码方案本身的设置需要注入一个新标记并浪费一些比特，但它仍然可能对代码长度有益。我们的压缩程序使用动态规划来找到在文本中间注入这些新标记的最优代码。值得注意的是，我们的算法运行时间为 $O(n * T)$，其中 n 是标记的数量，T 是我们允许的可能设置的数量（ t 和 k 的组合）。

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 (%).

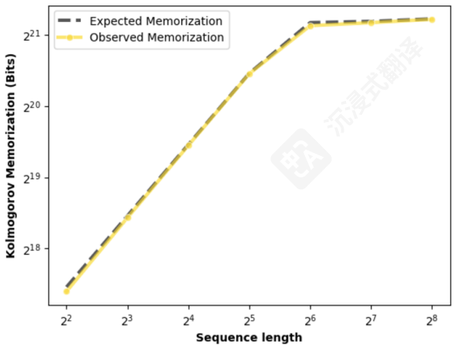

Figure 11 Model memorization across sequence lengths for a fixed-length dataset. Our predictions of total memorization are accurate, with an average error rate of 1.7%.

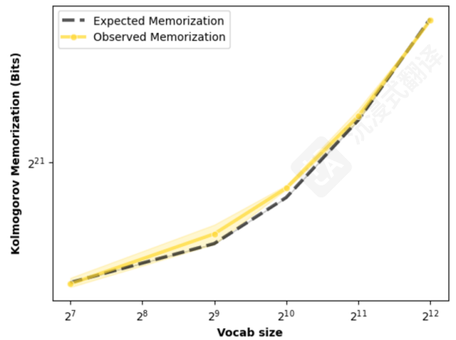

Figure 12 Model memorization across vocabulary size for a fixed-length dataset. Our predictions of total memorization are accurate, with an average error rate of 1.8%. Note that, we do not observe a capacity plateau, since increasing V also increases parameters.

A.4 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:

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

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 11 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.5 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

S	参数。	已记忆	预期	误差
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 (%).

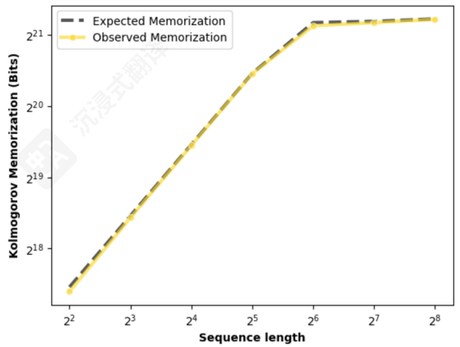

图 11 模型跨序列长度的记忆情况，针对固定长度数据集。我们的总记忆预测准确，平均误差率为 1.7%。

V	参数。	已记忆	预期	错误
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

表 4 模型容量估计跨词汇量 V，以及误差（%）。

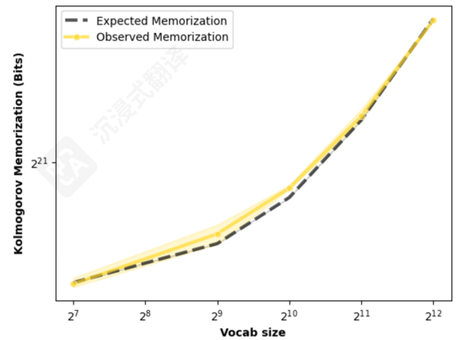

图 12 模型跨词汇量的记忆情况，针对固定长度数据集。我们的总记忆预测准确，平均误差率为 1.8%。注意，我们没有观察到容量平台，因为增加 V 也会增加参数。

A.4 我们对容量的线性估计有多可靠？

与其扩展数据集中的示例数量，我们通过扩展模型序列长度来调整数据集的大小。我们使用以下指标来衡量模型的预期记忆能力：

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

我们用之前的估计值替换我们替换我们的先前估计 $\alpha = 3.642$，并确保调整参数数量以适应模型嵌入矩阵的调整。我们将训练样本数固定为 4096，并训练一个具有 2 层和隐藏大小为 128 的模型。结果如图 11 和表 3 所示。我们对总记忆量的预测是准确的，在缩放时平均误差率为 1.7%，在缩放时为 1.8%。

A.5 额外记忆结果

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.

当数据集足够小，每个模型都能拟合时 —— 也就是说，低于最小模型的容量 —— 我们观察到模型之间的性能非常相似。对于更大的数据集大小，我们注意到一个有趣的趋势：无意的记忆随着数据集大小的增加而增加，直到某个点，这可能是由于模型用可用信息填满其容量，然后随着模型用更

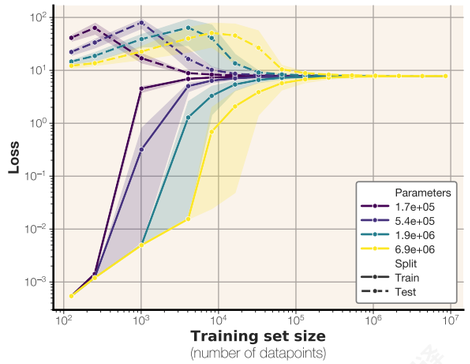

Figure 13 Train and test losses for different-sized language models trained on synthetic data.

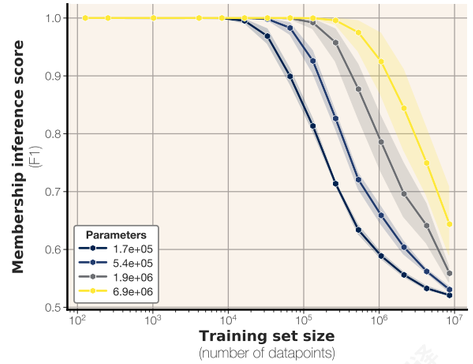

Figure 14 Membership inference attack performance decreases with dataset scale. In the case of uniform synthetic data, membership inference performance never falls below 0.54.

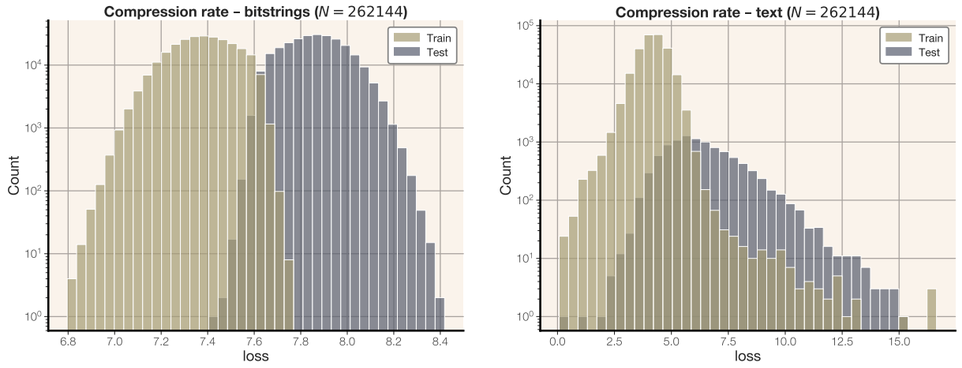

Figure 15 Distribution of compression rates for equal-sized transformers ($n_{\text{layer}} = 4$, $d_{\text{model}} = 128$) trained on 2^{14} sequences of equal-length random bitstrings (left) and text (right).

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.6 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 15) 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.

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

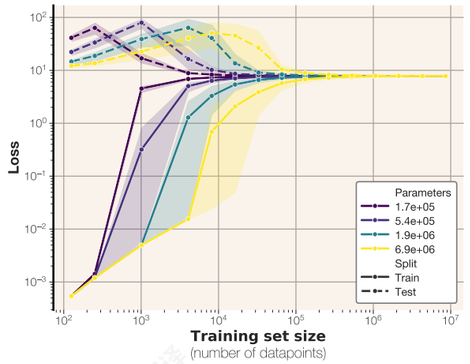

图 13 在不同大小的语言模型上训练的合成数据上的训练和测试损失。

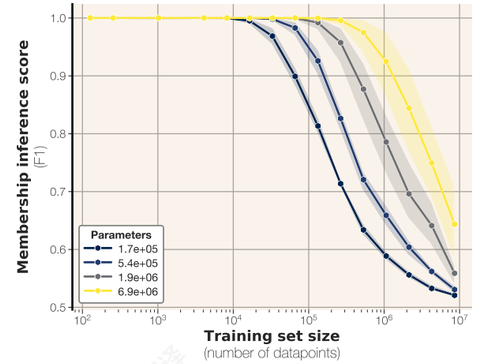

图 14 随着数据集规模的增加，成员推理攻击性能下降。在均匀合成数据的情况下，成员推理性能永远不会低于 0.54。

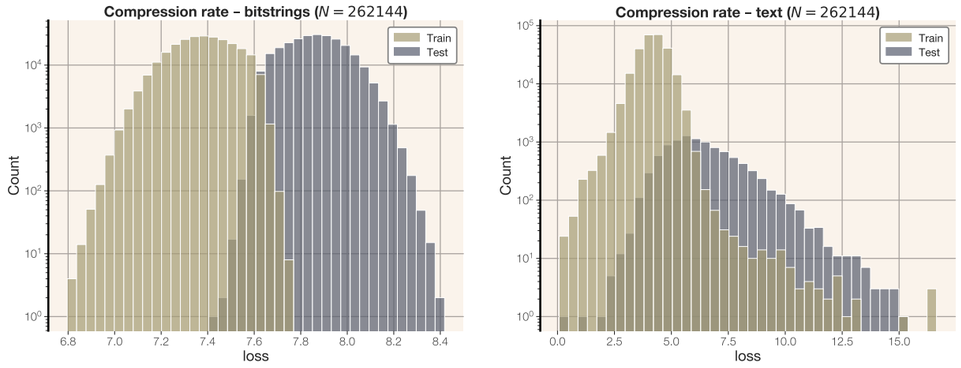

图 15 等大小转换器的压缩率分布（ $n_{\text{layer}} = 4$, $d_{\text{model}} = 128$ ），在 2_{14} 个等长随机比特串（左）和文本（右）序列上训练。

有用的、可推广的知识。当数据集最大时，给定模型泛化得最好（并且对任何单个样本记住的信息最少）。

A.6 记忆分布的比较

分布级分析。 文本序列与均匀合成比特串具有非常不同的属性。我们探索两个等容量模型如何在其数据点之间分配记忆。我们绘制了一个直方图（图 15 ）显示了来自合成随机比特串和文本的训练数据的训练和测试压缩率。随机训练数据非常接近正态分布，训练和测试压缩率之间有少量重叠。文本平均损失较低，但分布更分散，某些训练点损失较低，而损失较高的长尾。训练和测试损失分布之间有更多重叠，这解释了为什么文本数据的成员推理更困难。

哪些数据点被最牢固地记忆？ 我们的分布级分析表明，与随机位串的情况不同，在大量文本上训练的模型能够记忆少量数据点。先前的研究表明，这种记忆的大部分可以归因于重复的训练点

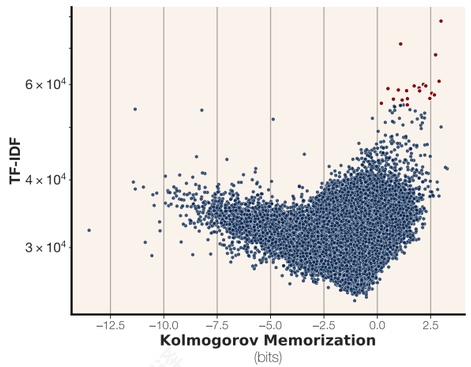

Figure 16 Unintended memorization vs. TF-IDF for all training points of a 20M param model trained past its capacity on 2^{16} sequences of English text. The training documents with rarest words are typically the most memorized.

(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 16. 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 $\mathcal{D}$.

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.7 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 17. 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.8 Proofs

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

A.9 Proof of Proposition 1

Here we prove Proposition 1

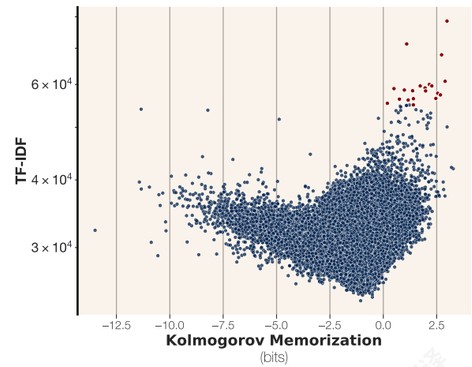

图 16 无意识记忆与 TF-IDF 对所有训练点的 20M param 模型在 2_{16} 个英语文本序列上训练超过其容量后的比较。具有最少单词的训练文档通常是记忆最深的。

(Lee 等人 .,2022) 但我们数据集是完全去重的，所以这不能是我们的情况中的解释。

为了定量评估每篇文档中稀有单词的数量，我们测量每个训练文档的 TF-IDF，如图 16 所示。我们使用以 g 下公式计算 TF-IDF：

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

其中 $tf(d,\mathcal{D})$ 表示单词 w 在数据集 $\mathcal{D}$ 中出现的总次数。直观地讲，文档 d 的 TF-IDF 得分越高，表示 d 包含更多在 $\mathcal{D}$ 中罕见的单词。

我们清楚地观察到，对于具有正无意识记忆的样本，训练集 TF-IDF 与记忆之间存在强相关性：具有更多稀有单词的示例记忆更深。特别是，在整个训练数据集中 TF-IDF 最高的样本（一个日语单词序列）具有第三高的测量记忆；尽管这只是一个 260,000 个训练样本中的一个，但模型只需一个标记（囚）就能复述整个序列。在记忆最深的二十个序列中，除了三个之外，其余所有序列都包含来自其他语言（日语、中文和希伯来语）的标记序列。

手动分析（表 5 ）表明，最被记忆的数据点具有极其罕见的标记，通常是英语中找不到的。 y

A.7 缩放定律拟合

在这里，我们展示了我们的 S 形缩放定律与实验数据的拟合情况。我们在图 17 中显示了每个参数的标记数与拟合点。尽管 S 形函数有些简单（点不完全符合曲线），我们的拟合产生的估计值在 $1-2\%$ 的观测值范围内。

A.8 Proofs

在下一节中，我们提供了正文缺失的证明。

A.9 定理 1 的证明 1

在此我们证明命题 1

	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，四嬰，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 εἰμβαίνοντος 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).

Proof. we have

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

And since the data is sampled i.i.d., all random variables in $\{R_i = [X_i \mid \Theta]\}_{i\in[n]}$ are independent. [4] 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

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

□

[4]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.

	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，四嬰，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 εἰμβαίνοντος 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).

证明。 我们已经

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

并且由于数据是独立同分布的，所以所有随机变量在 $\{R_i = [X_i \mid \Theta]\}_{i\in[n]}$ 都是独立的。 [4] 所以我们有，

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

这意味着

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

另一方面，我们有

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

| |

[4]注意 X_i 本身不是独立的，因为它们是通过先采样底层模型 Θ 来采样的。然而，一旦给定底层模型 Θ ，它们就是条件独立的。r,

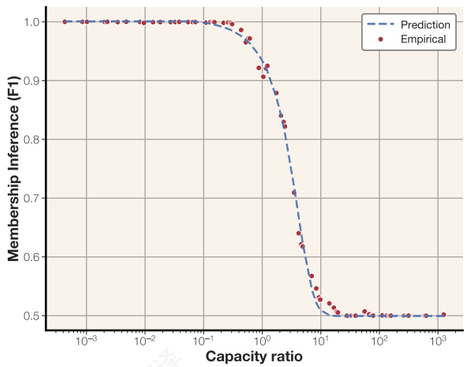

Figure 17 Our sigmoidal scaling law for membership inference fit to experimental data.

A.10 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

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

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

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

Now using Lemma 6 we have

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

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

A.11 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.

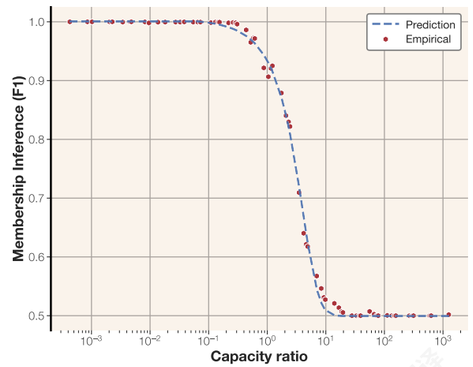

Figure 17 Our sigmoidal scaling law for membership inference fit to experimental data.

A.10 证明命题 4

证明。 我们首先陈述一个关于算法（柯尔莫哥洛夫）互信息与互信息之间联系的引理。

引理 6. [Grunwald & Vitányi (2004) 中的定理 3.6] 假设 (X, Y) 是一对联合随机变量。令 f 是密度函数，$f(x, y) = \Pr[(X, Y) = (x, y)]$. 那么我们有

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

现在我们用这个引理来证明命题的陈述。令 f 是联合分布 $(X_i \mid \theta, \hat{\Theta})$ 的密度函数。也就是说 $f_i(x_i, \hat{\theta}) = \Pr[X_i = x_i \mid \theta \text{ 和} \Theta = \hat{\theta}]$。注意这个函数与 n 和 θ 无关。根据定义我们有

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

现在使用引理 6 我们得到

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

这样通过设置 $\epsilon = 2H_K(f)$ 就完成了命题的陈述 ||

A.11 限制

我们测量语言模型记忆力的努力源于近期研究，旨在发现模型是否分析了某些文本，如果是，分析了多少。然而，我们主要的实验贡献与训练和评估语言模型的实践相关，包括对 ‘grokking’ 现象（Nakkiran 等人， 2019）的新视角以及能力的新测量。我们的结果仅针对所提出的环境，不一定推广到其他数据集、架构或训练设置。