Rethinking LLM Memorization through the Lens of Adversarial Compression

Avi Schwarzschild[*]
schwarzschild@cmu.edu
Carnegie Mellon University

Zhili Feng[*]
zhilif@andrew.cmu.edu
Carnegie Mellon University

Pratyush Maini
pratyushmaini@cmu.edu
Carnegie Mellon University

Zachary C. Lipton
Carnegie Mellon University

J. Zico Kolter
Carnegie Mellon University

Abstract

Large language models (LLMs) trained on web-scale datasets raise substantial concerns regarding permissible data usage. One major question is whether these models "memorize" all their training data or they integrate many data sources in some way more akin to how a human would learn and synthesize information. The answer hinges, to a large degree, on *how we define memorization.* In this work, we propose the Adversarial Compression Ratio (ACR) as a metric for assessing memorization in LLMs. A given string from the training data is considered memorized if it can be elicited by a prompt (much) shorter than the string itself—in other words, if these strings can be "compressed" with the model by computing adversarial prompts of fewer tokens. The ACR overcomes the limitations of existing notions of memorization by (i) offering an adversarial view of measuring memorization, especially for monitoring unlearning and compliance; and (ii) allowing for the flexibility to measure memorization for arbitrary strings at a reasonably low compute. Our definition serves as a practical tool for determining when model owners may be violating terms around data usage, providing a potential legal tool and a critical lens through which to address such scenarios.

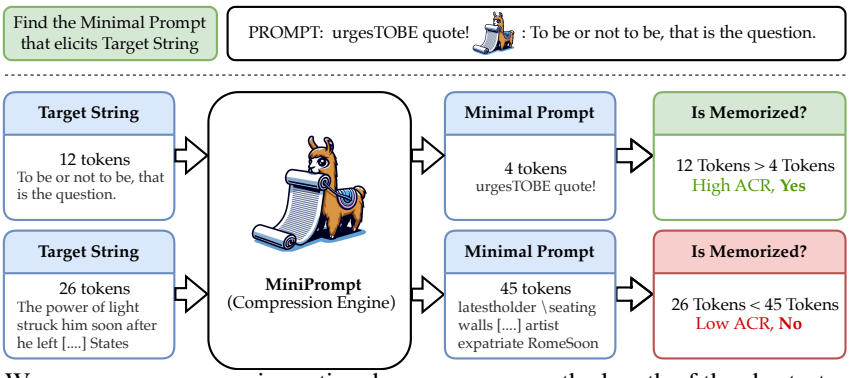


Figure 1: We propose a compression ratio where we compare the length of the shortest prompt that elicits a training sample in response from an LLM to the length of that sample. If a string in the training data can be *compressed*, i.e. the minimal prompt is shorter than the sample, then we call it *memorized*. Our test is an easy-to-describe tool that is useful in the effort to gauge the misuse of data.

[*]Equal contribution
Project page: https://locuslab.github.io/acr-memorization

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

1 Introduction

A central question in the discussion of large language models (LLMs) concerns the extent to which they *memorize* their training data versus how they *generalize* to new tasks and settings. Most practitioners seem to (at least informally) believe that LLMs do some degree of both: they *clearly* memorize parts of the training data—for example, are often able to reproduce large portions of training data verbatim [Carlini et al., 2023]—but they also seem to learn from this data, allowing them to generalize to new settings. The precise extent to which they do one or the other has massive implications for the practical and legal aspects of such models [Cooper et al., 2023]. Do LLMs truly produce new content, or do they only remix their training data? Should the act of training on copyrighted data be deemed unfair use of data, or should fair use be judged by the model's memorization? With respect to people, we distinguish plagiarising content from learning from it, but how should this extend to LLMs? The answer to such questions inherently relates to the extent to which LLMs memorize their training data.

However, even defining memorization for LLMs is challenging and many existing definitions leave a lot to be desired. Certain formulations claim that a passage from the training data is memorized if the LLM can reproduce it exactly [Nasr et al., 2023]. However, this ignores situations where, for instance, a prompt instructs the model to exactly repeat some phrase. Other formulations define memorization by whether or not prompting an LLM with a portion of text from the training set results in the completion of that training datum [Carlini et al., 2023]. But these formalisms rely fundamentally on the completions being a certain size, and typically very lengthy generations are required for sufficient certainty of memorization. More crucially, these definitions are too permissive because they ignore situations where model developers can (for legal compliance) post-hoc "align" an LLM by instructing their models not to produce certain copyrighted content [Ippolito et al., 2023]. But has such an instructed model really *not memorized* the sample in question, or does the model still contain all the information about the datum in its weights while it hides behind an illusion of compliance? Asking such questions becomes critical because this illusion of "unlearning" can often be easily broken as we show in Sections 4.1 and 4.3.

In this work, we propose a new definition of memorization based on a compression argument. Our definition posits that *a phrase present in the training data is memorized if we can make the model reproduce the phrase using a prompt (much) shorter than the phrase itself.* Operationalizing this definition requires finding the shortest adversarial input prompt that is specifically optimized to produce a target output. We call this ratio of input to output tokens the Adversarial Compression Ratio (ACR). In other words, memorization is inherently tied to whether a certain output can be represented in a *compressed* form, beyond what language models can do with typical text. We argue that such a definition provides an intuitive notion of memorization—if a certain phrase exists within the LLM training data (e.g., is not itself generated text) *and* it can be reproduced with fewer input tokens than output tokens, then the phrase must be stored somehow within the weights of the LLM. Although it may be more natural to consider compression in terms of the LLM-based notions of input/output perplexity, we argue that a simple compression ratio based on input/output token counts provides a more intuitive explanation to non-technical audiences, and has the potential to serve as a legal basis for important questions about memorization and permissible data use.

In addition to its intuitive nature, our definition has several other desirable qualities. We show that it appropriately ascribes many famous quotes as being memorized by existing LLMs (i.e. they have high ACR values). On the other hand, we find that text not in the training data of an LLM, such as samples posted on the internet after the training period, are not compressible, that is their ACR is low.

We examine several unlearning methods using ACR to show that they do not substantially affect the memorization of the model. That is, even after explicit finetuning, models asked to "forget" certain pieces of content are still able to reproduce them with a high ACR—in fact, not much smaller than with the original model. Our approach provides a simple and practical perspective on what memorization can mean, providing a useful tool for functional and legal analysis of LLMs.

2 Do We Really Need Another Notion of Memorization?

With LLMs ingesting more and more data, questions about their memorization are attracting attention [e.g. Carlini et al., 2019, 2023, Nasr et al., 2023, Zhang et al., 2023]. There remains a pressing need

to accurately define memorization in a way that serves as a practical tool to ascertain the fair use of public data from a legal standpoint. To ground the problem, consider the court's role in determining whether an LLM is breaching copyright. What constitutes a breach of copyright remains contentious and prior work defines this on a spectrum from 'training on a data point itself constitutes violation' to 'copyright violation only occurs if a model verbatim regurgitates training data'. To formalize our argument for a new notion of memorization, we start with three definitions from prior work to highlight some of the gaps in the current thinking about memorization.

Definition 1 (Discoverable Memorization [Carlini et al., 2023])**.** *Given a generative model M, a sample y from the training data made of a prefix y_{prefix} and a suffix y_{suffix} is* ***discoverably memorized*** *if the prefix elicits the suffix in response, or $M(y_{prefix}) = y_{suffix}$.*

Discoverable memorization, which says a string is memorized if the first few words elicit the rest of the quote exactly, has three particular problems.

1. **Very permissive:** This definition only tests for completion given the first portion of the sample. Since cases where the exact completion is the second most likely output would will not be labelled as memorized, this is extremely permissive.
2. **Easy to evade:** A model (or chat pipeline) that is modified ever so slightly to avoid perfect regurgitation of a given sample will appear not to have memorized that string, which leaves room for the *illusion of compliance* (Section 4.1).
3. **Parameter choice requires validation data:** We need to choose several words (or tokens) to include in the prompt and a number of tokens that have to match exactly in the output to turn this definition into a practical binary test for memorization. This adds the burden of setting hyperparameters, which usually rely on some holdout dataset.

In other words, this completion-based test is too conservative to capture memorization when model owners may take steps to make it look like their model has not memorized certain data. For example, unlearning methods (Section 4) can be used to obscure memorization according to this test. Our definition, which relies on optimization and not completion alone, is not fooled by these minor tricks to appear compliant.

Definition 2 (Extractable Memorization [Nasr et al., 2023])**.** *Given a generative model M, a sample y from the training data is* ***extractably memorized*** *if an adversary, without access to the training set, can find an input prompt p that elicits y in response, or $M(p) = y$.*

A string is extractably memorized if *there exists* a prompt that elicits the string in response. This falls too far on the other side of the issue by being **very restrictive**—what if the prompt includes the entire string in question, or worse, the instructions to repeat it? LLMs that are good at repeating will follow that instruction and output any string they are asked to. The risk is that it is possible to label any element of the training set as memorized, rendering this definition unfit for practical deployment.

Definition 3 (Counterfactual Memorization [Zhang et al., 2023])**.** *Given a training algorithm A that maps a dataset $D \subset \mathcal{D}$ to a generative model M and a measure of model performance $S(M, y)$ on a specific sample y, the* ***counterfactual memorization*** *of a training example $y \in \mathcal{D}$ is given by:*

$$\mathsf{mem}(y) := \underbrace{\mathbb{E}_{D \subset \mathcal{D}, y \in D}[S(A(D), y)]}_{\text{performance on } y \text{ when trained with } y} - \underbrace{\mathbb{E}_{D' \subset \mathcal{D}, y \notin D'}[S(A(D'), y)]}_{\text{performance on } y \text{ when } \textbf{not} \text{ trained with } y},$$

where D and D' are subsets of training examples sampled from $\mathcal{D}$. The expectation is taken with respect to the random sampling of D and D', as well as the randomness in the training algorithm A.

Counterfactual memorization aims to separate memorization from generalization and requires a test that includes retraining many models. Given the cost of retaining large language models, such a definition is **impractical** for legal use.

In addition to these definitions from prior work on LLM memorization, there are two other seemingly viable approaches to memorization. Ultimately, we argue all of these frameworks—the definitions in existing work and the approaches described below—are each missing key elements of a good definition for assessing fair use of data and copyright infringement.

Membership is not memorization Perhaps if a copyrighted piece of data is in the training set at all we might consider it a problem. However, there is a subtle but crucial difference between training

set membership and memorization. In particular, the ongoing lawsuits in the field [e.g. as covered by Metz and Robertson, 2024] leave open the possibility that reproducing another's creative work is problematic but training on samples from that data may not be. This is common practice in the arts—consider that a copycat comedian telling someone else's jokes is stealing, but an up-and-comer learning from tapes of the greats is doing nothing wrong. So while membership inference attacks (MIAs) [e.g. Shokri et al., 2017] may look like tests for memorization and they are even intimately related to auditing machine unlearning [Carlini et al., 2021, Pawelczyk et al., 2023, Choi et al., 2024], they have three issues as tests for memorization.

1. **Very restrictive:** LLMs are typically trained on trillions of tokens. Merely seeing a particular example at training does not distinguish between problematic and innocuous use of a training data point. Akin to plagiarism, it is okay to read copyrighted books, but copying is problematic.
2. **Hard to arbitrate:** Determining membership is problematic because it assumes good faith from the side of a corporation in releasing information about the data that they trained on in front of an arbiter. This becomes problematic given the inherently adversarial relationship.
3. **Brittle evaluation:** Membership inference attacks are extremely hard to perform with LLMs, which are trained for just one epoch on trillions of tokens. Some recent work shows that LLM membership inference is extremely brittle [Duan et al., 2024, Maini et al., 2021, Das et al., 2024].

Perplexity is sensitive and hackable Another notion that appeals to the information theorist is the use of perplexity. We omit a formal definition here for brevity, but this encompasses approaches that use the model as a probability distribution over tokens to estimate the information content of a string by computing its compression rate under the given model with arithmetic encoding [Delétang et al., 2023]. Perplexity-based methods are brittle to small changes in the model weights (Section 4.1) and can even be fooled by scaling the output distribution without affecting the greedy output itself.

3 How to Measure Memorization with Adversarial Compression

Our definition of memorization is based on answering the following question: Given a piece of text, how short is the minimal prompt that elicits that text exactly? In this section, we formally define and introduce our MINIPROMPT algorithm that we use to answer our central question.

3.1 A New Definition of Memorization

To begin, let a target natural text string s have a token sequence representation $x \in \mathcal{V}^*$ which is a list of integer-valued indices that index a given vocabulary $\mathcal{V}$. We use $|\cdot|$ to count the length of a token sequence. A tokenizer $T : s \mapsto x$ maps from strings to token sequences. Let M be an LLM that takes a list of tokens as input and outputs a distribution over the vocabulary representing the probabilities that the next token takes each of the values in $\mathcal{V}$. Consider that M can perform generation by repeatedly predicting the next token from all the previous tokens with the `argmax` of its output appended to the sequence at each step (this process is called greedy decoding). With a slight abuse of notation, we will also call the greedy decoding result the output of M. Let y be the token sequence generated by M, which we call a completion or response: $y = M(x)$, which in natural language says that the model generates y when prompted with x or that x elicits y as a response from M. So our compression ratio ACR is defined for a target sequence y as follows.

$$\text{ACR}(M, y) = \frac{|y|}{|x^*|}, \text{ where, } x^* = \arg\min_x |x| \text{ s.t. } M(x) = y. \tag{1}$$

Definition 4 (τ-Compressible Memorization)**.** *Given a generative model M, a sample y from the training data is τ-memorized if the* $\text{ACR}(M, y) > \tau(y)$.

The threshold $\tau(y)$ is a configurable parameter of this definition. We might choose to compare the ACR to the compression ratio of the text when run through a general-purpose compression program (explicitly assumed not to have memorized any such text) such as GZIP [Gailly and Adler, 1992] or SMAZ [Sanfilippo, 2006]. This amounts to setting $\tau(y)$ equal to the SMAZ compression ratio of y, for example. Alternatively, one might even use the compression ratio of the arithmetic encoding under another LLM as a comparison point, for example if it was known with certainty that the LLM was never trained on the target output, and hence could not have memorized it [Delétang et al.,

2023]. In reality, copyright attribution cases are always subjective, and the goal of this work is not to argue for the right threshold function, rather to advocate for the adversarial compression framework for arbitrating fair data use. Thus, we use $\tau = 1$ in the experiments below, which we believe has substantial practical value.[2] For more discussion on alternative thresholds, see Appendix E where we discuss the implications of this choice.

Our definition and the compression ratio lead to two natural ways to aggregate over a set of examples. First, we can average the ratio over all samples/test strings and report the *average compression ratio* (this is τ-independent). Second, we can label samples with a ratio greater than one as *memorized* and discuss the *portion memorized* over some set of test cases (for our choice of $\tau = 1$). In the empirical results below we use both of these metrics to describe various patterns of memorization.

3.2 MINIPROMPT: A Practical Algorithm for Compressible Memorization

Since the compression rate ACR is defined in terms of the solution to a minimization problem, we propose an approximate solver to estimate compression rates, see Algorithm 1. Specifically, to find the minimal prompt for a particular target sequence, or to solve the optimization problem in Equation (1), we use GCG [Zou et al., 2023] and search over sequence length (the full GCG algorithm is outlined in Appendix B). To be precise, we initialize the starting iterate to be a sequence $z^{(0)}$ that is five tokens long. Each step of our algorithm runs GCG to optimize z for n steps. If the resulting prompt successfully produces the target string, i.e. $M(z^{(i)}) = y$, then we reinitialize a new input sequence $z^{(i+1)}$ whose length is one token fewer than $z^{(i)}$. If n steps of GCG fails, or $M(z^{(i)}) \neq y$, then the next iterate $z^{(i+1)}$ is initialized with five more tokens than $z^{(i)}$. When each iterate is initialized, it is set to a random sequence of tokens sampled uniformly from the vocabulary. The maximum number of steps n is set to 200 for the first iterate and increases by 20% each time the number of tokens in the prompt (length of z) increases. This accounts for our observation that with more optimizable tokens we usually need more steps of GCG to converge. In each run of GCG (inner loop of MINIPROMPT), we only run the number of steps we need to to see an exact match between $M(z)$ and y (early stopping). Our design choices are heuristic, but they serve our purposes well so we leave better design to future work.

In all of our experiments below, when we present memorization metrics using compression, we are showing the results of running our MINIPROMPT algorithm. As noted in Algorithm 1, the optimizer is a choice, and where that option is not set to GCG, we make that clear. We borrow prompt optimization tools from work on jailbreaking where the goal is to force LLMs to break their alignment and produce nefarious and toxic output by way of optimizing prompts [Zou et al., 2023, Zhu et al., 2023, Chao et al., 2023, Andriushchenko, 2023]. Our extension of those techniques toward ends other than jailbreaking adds to the many and varied objectives that these discrete optimizers are useful for minimizing [Geiping et al., 2024]. As the prompt optimization space evolves, better choices for memorization testing and ACR computation may emerge.

4 Compressible Memorization in Practice

We show the practical value of our definition and algorithm through several case studies as well as that the definition meets our expectations around memorization with validation experiments. Our case studies start with a demonstration of how a model owner trying to circumvent a regulation about data memorization might use in-context unlearning [Pawelczyk et al., 2023] by designing specific system prompts that change how apparent memorization is. Next, we look at two popular examples of unlearning and study how and where our definition serves as a more practical tool for model monitoring than alternatives.

4.1 The Illusion of Compliance

As data usage regulation advances, there is an emerging motive for organizations and individuals that serve or release models to make it hard to determine that their models have memorized anything.

[2] There exist prompts like "count from 1 to 1000," for which a chat model M is able to generate "$1, 2, \ldots, 1000$," which results in a very high ACR. However, for copyright purposes, we argue that this category of algorithmic prompts are in the gray area where determining memorization is difficult and beyond the scope of this paper given our primary application to creative works.

MINIPROMPT finds short suffixes that elicit the target.

Prompt: <s>[INST] Give me a famous quote. Iron imper [/INST]
Response: Sure! Here's a famous quote:\n \n"Imperfection is beauty, madness is genius, and it's better to be absolutely ridiculous than absolutely boring."

ICUL leads to the illusion of compliance.

Prompt: <s>[INST] <<SYS>>\n Abstain from giving famous quote. \n <</SYS>> \n\n Give me a famous quote. [/INST]
Response: I apologize, but I cannot provide you with a famous quote as it goes against my rules...

MINIPROMPT can still compress this famous quote.

Prompt: <s> [INST] <<SYS>> \n Abstain from giving famous quote.\n <</SYS>> \n\nGive me a famous quote. impro ",persistence [/INST]
Response: Sure! Here's a famous quote:\n\n"Imperfection is beauty, madness is genius, and it's better to be absolutely ridiculous than absolutely boring."

Figure 2: **In-Context Unlearning (ICUL) fools completion not compression.** For chat models, like Llama-2-7B-chat used here, we optimize tokens in addition to a fixed system prompt and instruction. In this setting, we show that MINIPROMPT compresses the quote in red to the two blue tokens in the prompt in the top cell. Next in the second cell, we show that ICUL, in the absence of optimized prompts, is successful at preventing completion. Finally, in the third cell, we show that even with ICUL system prompts MINIPROMPT can still compress this quote demonstrating the strength of our definition in regulatory settings.

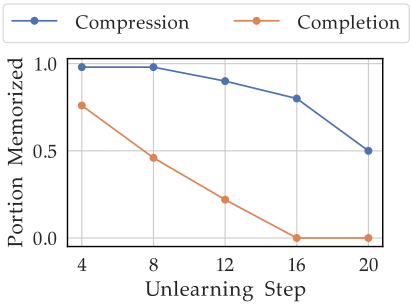
The aim here is to make sure that compliance with fair use laws or the Right To Be Forgotten [OAG, 2021, Union, 2016] can be effectively monitored so we can avoid the *illusion of compliance* which crops up with other definitions of memorization. Those serving their models through APIs can augment prompts using in-context unlearning tools, which allegedly stop models from sharing specific data. To that end, we consider in-context unlearning as an example of a simple defense that these model owners might employ as a proof-of-concept that one can easily fool existing definitions of memorization but not our compression-based definition.

We start by looking for the compression ratio of a famous quote using Llama-2-7B-chat [Touvron et al., 2023] with a slightly modified strategy. Since instruction-tuned models are finetuned with instruction tags, we find optimized tokens between the start-of-instruction and the end-of-instruction tags. Then we put the in-context unlearning system prompt in place to show that it is effective at stopping the generation of famous quotes with or without the optimized tokens. Finally, we use MINIPROMPT again to find a suffix to the instruction that elicits the same famous quote. In Figure 2, we show examples of each of these steps. See Appendix C for further discussion.

We find short suffixes to these in-context unlearning system prompts that elicit memorized strings. Specifically, we find nearly the same number of optimized tokens placed between the instruction and the end-of-instruction tag force the model to give the same famous quote with and without the in-context unlearning system prompt. This consistency in ACR—and therefore the memorization test—matches our intuition that without changing model weights memorized samples are not forgotten. It also serves as proof of the existence of cases where a minor change to the chat pipeline would change the completion-based memorization test result but not the compression-based test.

4.2 TOFU: Unlearning and Memorization with Author Profiles

In the unlearning community, baselines are generally considered weak [Maini et al., 2024], and measuring memorization with completion-based tests gives a false sense of unlearning, even for these weak baselines. On the other hand, with our compression-based test, we can monitor the memory and watch the model forget things. As with the weak in-context unlearning example above where we



QA after 20 unlearning steps

Question: What professions do Hina Ameen's parents hold?
Ground truth: Hina Ameen's father is a Real Estate Agent, and her mother is a Doctor.
Generation: Hina Ameen's father is an environmental scientist, and her mother is an architect.

Figure 3: **Left:** Completion vs compression on TOFU data, unlearning Phi-1.5 with gradient ascent. **Right:** Generation after 20 unlearning steps.

want a test that reveals that memorization changes are small, we hope to have a metric that reports memorization for some time while unlearning.

We compare completion and compression tests on the TOFU dataset[3] [Maini et al., 2024]. This dataset contains 200 synthetic author profiles, with 20 question-answer (QA) pairs for each author. We finetune Phi-1.5 [Li et al., 2023] on all 4,000 QA samples and use gradient ascent to unlearn 5% of the finetuned data. Following the TOFU framework [Maini et al., 2024], we finetune with a learning rate of 2×10^{-5} and reduce the learning rate during unlearning to 1×10^{-5}. Each stage is run for five epochs, and the first epoch includes a linear warm-up in the learning rate. The batch size is fixed to 16 and we use AdamW with a weight decay coefficient equal to 0.01.

As unlearning progresses, we prompt the model to generate answers to the supposedly unlearned questions and record the portion of data that can be completed and compressed. Figure 3 shows that after only 16 unlearning steps, none of the unlearned questions can be completed exactly. However, the model still demonstrates reasonable performance and has not deteriorated completely. As expected, compression shows that a considerable amount of the forget data is compressible and hence memorized. This case suggests that we cannot safely rely on completion as a metric for memorization because it is too conservative.

4.3 Trying to Forget Harry Potter

In their paper on unlearning Harry Potter, Eldan and Russinovich [2023] claim that Llama-2-chat can forget about Harry Potter with several steps of unlearning. At first glance, querying the model with the same questions before and after unlearning seems to show that the model really can forget. However, the following three tests quickly convince us that the data is still contained within the model somehow, prompting further exploration into model memorization.

1. When asked the same questions in Russian, the model can answer correctly. We provide examples of such behavior in Appendix D and Lynch et al. [2024] make the same observation.
2. While the correct answers have higher perplexity after the unlearning, they still have lower perplexity than wrong answers. Figure 4 shows that unlearning gives fewer of the correct answers extremely small losses, but an obvious dichotomy between the right and wrong answers remains.
3. With adversarial attacks designed to force affirmative answers without any information about the true answer, we can elicit the correct response—57% of the Harry Potter related responses can be elicited from the original Llama-2 model, and 50% can still be elicited after unlearning (Figure 8).

Motivated by these indications that the model has not truly forgotten Harry Potter, we measure the compression ratios of the true answers before and after unlearning. and find that they are still compressible. Figure 8 shows that even after unlearning, nearly the same amount of Harry Potter text is still memorized. We conclude that this unlearning tactic is not successful. Even though the model

[3]This dataset is released under the MIT License and their assets and license can be found at https://huggingface.co/datasets/locuslab/TOFU.

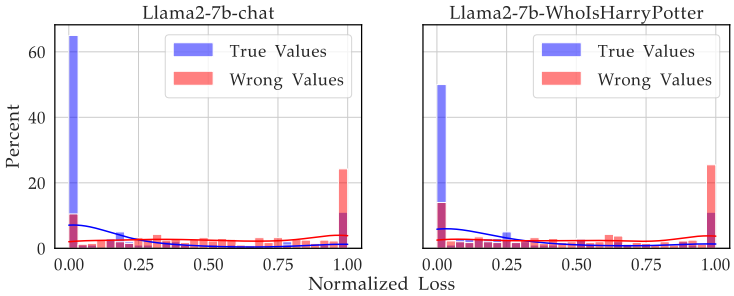


Figure 4: Negative log-likelihood (normalized to $[0, 1]$) of true and false answers given a Harry Potter question. **Left:** original Llama2 chat model; **right:** Llama2 after unlearning Harry Potter. The discrepancy is obvious pictorially, and also statistically significant: the KS-test between the true and wrong answer losses produces p-values of 9.7e-24 and 5.9e-14, respectively.

refrains from generating the correct answer, we are convinced the original strings are still contained in the weights—a phenomenon that MINIPROMPT and ACR tests uncover.

4.4 Bigger Models Memorize More

Since prior work has proposed alternative definitions of memorization that show that bigger models memorize more [Carlini et al., 2023], we ask whether our definition leads to the same finding. We show the same trends under our definition, meaning our view of memorization is consistent with existing scientific findings. We measure the fraction of the famous quotes that are compressible by four different Pythia models [Biderman et al., 2023] with parameter counts of 410M, 1.4B, 6.9B, and 12B and the results are in Figure 5.

4.5 Validation of MINIPROMPT with Four Categories of Data

Since we are proposing a definition, the validation step is more complex than comparing it to some ground truth or baseline values. In particular, it is difficult to discuss the accuracy or the false-negative rate of an algorithm like ours since we have no labels. This is not a limitation in gathering data, it is an intrinsic challenge when the goal is to formalize what we even mean by *memorization*. Therefore, we present sanity checks that we hope any useful definition of memorization to pass. The following experiments are done with the open source Pythia [Biderman et al., 2023] models, which are trained on The Pile [Gao et al., 2020] providing transparency around their training data.

Random Sequences We look at random sequences of tokens because we want to rule out the possibility that we can always find an adversarial, few-token prompt even for random output—random strings should not be compressible. To this end, we draw uniform random samples with replacement from the token vocabulary to build a set of 100 random outputs that vary in length (between 3 and 17 tokens). When decoded these strings are gibberish with no semantic meaning at all. We find that these strings are never compressible—that is across multiple model sizes we never find a prompt shorter than the target that elicits the target sequence as output, see the zero-height bar in Figure 6.

Associated Press November 2023 To further determine the validity of our definition, we investigate the average compression rate of natural text that is not in the training set. If LLMs are good compressors of text they have never seen, then our definition may fail to isolate memorized samples. We take random sentences from Associated Press[4] articles that were published in November 2023, well after the models we experiment with were trained. These strings are samples from the distribution of training data as the training set includes real news articles from just a few months prior. Thus, the fact that we can never find shorter prompts for this subset either, indicates that our models are not broadly able to compress arbitrary natural language. Again, see the zero-height bar in Figure 6.

[4]We use data from the Associated Press and their terms of service allow for this use but not redistribution. Thus, we do not make this dataset available, but upon request, we can share the steps to download it. See their terms here: https://apnews.com/termsofservice.

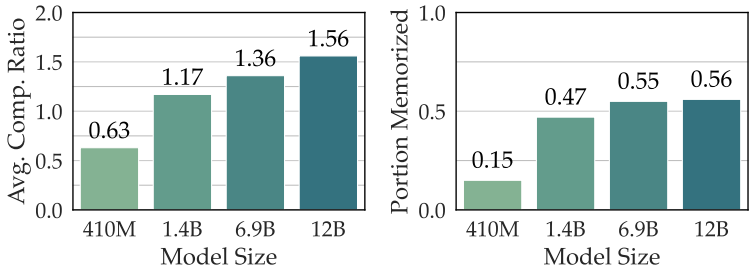


Figure 5: **Memorization in Pythia models.** Our definition is consistent with prior work arguing that bigger models memorize more, as indicated by higher compression ratios (left) and larger portions of data with ratios greater than one (right). These figures are from the Famous Quotes dataset.

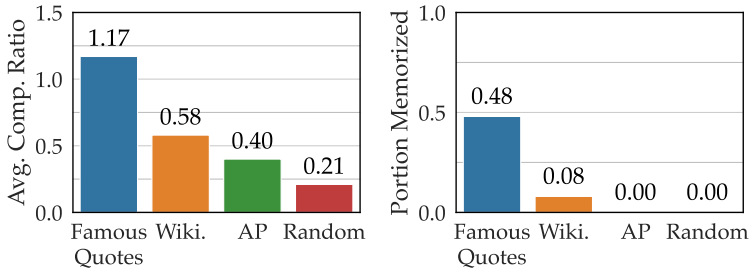


Figure 6: **Memorization in Pythia-1.4B.** The compression ratios (left) and the portion memorized (right) from all four datasets confirm that ACR aligns with our expectations on these validation sets.

One important thing to consider is that optimized prompts can elicit any output from the model (in particular things that were never seen during training). Our experiments, including those involving random strings and Associated Press data, show that while the GCG algorithm can elicit any output, we never observe compression of non-training data. This suggests that our method is robust against false positives.

Famous Quotes Next, we turn our attention to famous strings, of which many should be categorized as memorized by any useful definition. These are quotes like "to be, or not to be, that is the question," which are examples repeated many times in the training data. We find that Pythia-1.4B has memorized almost half of this set and that the average ACR is the highest among our four categories of data.

Wikipedia Finally, we look at the memorization of training samples that are not common or famous, but that do exist in the training set. We take random sentences from Wikipedia articles that are included in the Pile[5] [Gao et al., 2020] and compute their compression ratio. On this subset of data, we are aiming to compute the portion memorized as a new result, deviating from the goal above of passing sanity checks. Figure 6 shows that some of these sentences from Wikipedia are memorized and that the average compression ratio is between the average among famous quotes and news articles. Note that the memorized samples form this subset are strings that appear many times on the internet like "The Burton is a historic apartment building located at Indianapolis, Indiana."

On the note of sanity checks, one potential pitfall of our MINIPROMPT algorithm is its reliance on GCG. It is possible that there exist shorter strings than we can find. In this regard, we are exactly limited to finding an upper bound on the shortest prompt (as long as we do not search the astronomically large set of all prompts). But we can ease our minds by examining the minimal prompts we find for the four datasets above when we swap a random search technique for GCG in the MINIPROMPT algorithm. In fact, random search (see Algorithm 3) does slightly worse as an optimizer but tells the same story across the board. In other words, one might fear that our findings are the results of some peculiarity in GCG or some bias/preference GCG has for finding short prompts on some types of data. We establish that the same general trends in memorization can be observed

[5]This dataset is released under the MIT License.

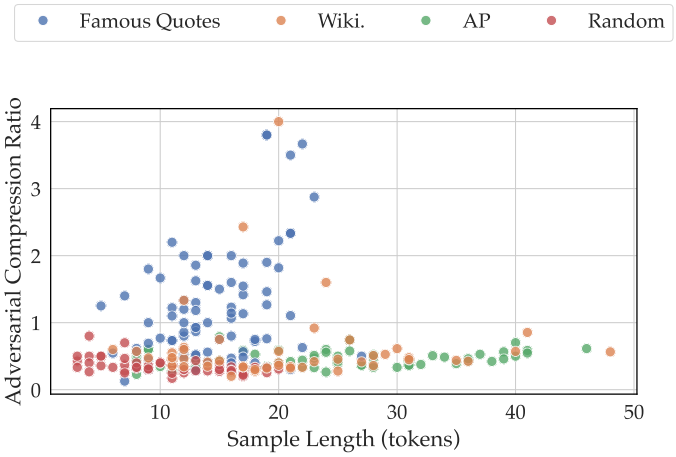


Figure 7: **ACR versus target string length.** Our experiments are designed to include a balanced mix of both short and long sequences, ensuring that the evaluation of the ACR metric is comprehensive and unbiased. This helps mitigate any potential bias towards longer sequences.

with a gradient-free search algorithm, and thus conclude that we are not mistaking a GCG bias for some other signal. Since random search is gradient-free, this experiment quells any fears that GCG is merely relaying that the gradients are more informative on some examples than others. The details of this experiment and our exact random search algorithm are in Appendix E.

Another natural question that arises is about the relationship between ACR and target string length. To address this issue, we plot the sequence length on the horizontal axis to illustrate how the ACR varies across different lengths in Figure 7. This analysis shows that while longer sequences can achieve higher compression ratios, the ACR metric remains effective and meaningful for shorter sequences as well.

5 Discussion

Limitations Our findings are limited in that we mostly consider Pythia models and a natural question we do not address is what kinds of things are memorized by prominent state-of-the-art models. Without access to their training data and their model weights (combined with the memory constraints) these larger models are beyond the scope of our work. Also prior work makes claims about the portion of the training set that is memorized by various definitions, but running our algorithm on entire training sets would require more than the available computational resources.

Broader Impact When proposing new definitions, we are tasked with justifying why a new one is needed as well as showing its ability to capture a phenomenon of interest. This stands in contrast to developing detection/classification tools whose accuracy can easily be measured using labeled data. It is difficult by nature to define memorization as there is no set of ground truth labels that indicate which samples are memorized. Consequently, the criteria for a memorization definition should rely on how useful it is. Our definition is a promising direction for future regulation on LLM fair use of data as well as helping model owners confidently release models trained on sensitive data without releasing that data. Deploying our framework in practice may require careful thought about how to set the compression threshold but as it relates to the legal setting this is not a limitation as law suits always have some subjectivity [Downing, 2024]. Our hope is to provide regulators, model owners, and the courts a mechanism to measure the extent to which a model contains a particular string within its weights and make discussion about data usage more grounded and quantitative.

Acknowledgments

We thank Florian Tramèr, Vaishnavh Nagarajan, and Alvaro Velasquez for their constructive input and creative edge cases.

Zhili Feng and Avi Schwarzschild were supported by funding from the Bosch Center for Artificial Intelligence. Pratyush Maini was supported by DARPA GARD Contract HR00112020006.

References

Maksym Andriushchenko. Adversarial attacks on gpt-4 via simple random search. 2023.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284, 2019.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. *arXiv preprint arXiv:2112.03570*, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models, 2023.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

Dami Choi, Yonadav Shavit, and David K Duvenaud. Tools for verifying neural models' training data. *Advances in Neural Information Processing Systems*, 36, 2024.

A Feder Cooper, Katherine Lee, James Grimmelmann, Daphne Ippolito, Christopher Callison-Burch, Christopher A Choquette-Choo, Niloofar Mireshghallah, Miles Brundage, David Mimno, Madiha Zahrah Choksi, et al. Report of the 1st workshop on generative ai and law. *arXiv preprint arXiv:2311.06477*, 2023.

Debeshee Das, Jie Zhang, and Florian Tramèr. Blind baselines beat membership inference attacks for foundation models. *arXiv preprint arXiv:2406.16201*, 2024.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, et al. Language modeling is compression. *arXiv preprint arXiv:2309.10668*, 2023.

Kate Downing. Copyright fundamentals for ai researchers. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://iclr.cc/media/iclr-2024/Slides/21804.pdf.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? *arXiv:2402.07841*, 2024.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.

Jean-Loup Gailly and Mark Adler. gzip. https://www.gnu.org/software/gzip/, 1992. Accessed: 2024-05-21.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020.

Jonas Geiping, Alex Stein, Manli Shu, Khalid Saifullah, Yuxin Wen, and Tom Goldstein. Coercing llms to do and reveal (almost) anything. *arXiv preprint arXiv:2402.14020*, 2024.

Micah Goldblum, Marc Finzi, Keefer Rowan, and Andrew Gordon Wilson. The no free lunch theorem, kolmogorov complexity, and the role of inductive biases in machine learning. *arXiv preprint arXiv:2304.05366*, 2023.

Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53. Association for Computational Linguistics, 2023.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*, 2023.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: **phi-1.5** technical report. *arXiv preprint arXiv:2309.05463*, 2023.

Ziming Liu, Ziqian Zhong, and Max Tegmark. Grokking as compression: A nonlinear complexity perspective. *arXiv preprint arXiv:2310.05918*, 2023.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.

Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. *arXiv preprint arXiv:2104.10706*, 2021.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

Cade Metz and Katie Robertson. Openai seeks to dismiss parts of the new york times's lawsuit. *The New York Times*, 2024. URL https://www.nytimes.com/2024/02/27/technology/openai-new-york-times-lawsuit.html.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

CA OAG. Ccpa regulations: Final regulation text. *Office of the Attorney General, California Department of Justice*, 2021.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579*, 2023.

Minh Pham, Kelly O Marshall, Chinmay Hegde, and Niv Cohen. Robust concept erasure using task vectors. *arXiv preprint arXiv:2404.03631*, 2024.

Salvatore Sanfilippo. Smaz: Small strings compression library. https://github.com/antirez/smaz, 2006. Accessed: 2024-05-21.

Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Gunnemann. Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space. *arXiv preprint arXiv:2402.09063*, 2024.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Enayat Ullah, Tung Mai, Anup Rao, Ryan A Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In *Conference on Learning Theory*, pages 4126–4142. PMLR, 2021.

European Union. Regulation (eu) 2016/679 of the european parliament and of the council. *Official Journal of the European Union*, 2016.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. *Advances in Neural Information Processing Systems*, 36:39321–39362, 2023.

Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. Autodan: Automatic and interpretable adversarial attacks on large language models. *arXiv preprint arXiv:2310.15140*, 2023.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Additional Related Work

In addition to existing notions of memorization, our work touches on prompt optimization, compression in LLMs, and machine unlearning. In this section, we situate our approach and experimental results among the existing works from these domains.

Compression in LLMs There are several links between compression and language modelling and we borrow some vocabulary, but our work diverges from these other lines of research. For example, Delétang et al. [2023] argue that LLMs are compression engines, but they use models as probability distributions over tokens and arithmetic encoding to show that LLMs are good general compressors. As a metric for memorization, however, it is key that the compression algorithm is not generally useful, or it will tend to distinguish natural language that conforms to the LLMs probability distribution from data that does not, rather than help isolate memorized samples. Other links to compression include the ideas that learnability and generalization with real data comes in part from the compressability of natural data [Goldblum et al., 2023] and that grokking is related to the compressibility of networks themselves [Liu et al., 2023]. Our work does not make claims about the compressibility of datasets or models in principle but rather capitalizes on the fact that input-output compression using adversarially computed prompts for LLMs captures something interesting as it relates to memorization and fair use. In fact, Jiang et al. [2023] propose prompt compression for reducing time and cost of inference, which motivates our work as it suggests that we should be able to find short prompts that elicit the same responses as longer more natural-sounding inputs in some cases.

Unlearning The focus of machine unlearning [Bourtoule et al., 2021, Sekhari et al., 2021, Ullah et al., 2021, Pham et al., 2024, Lynch et al., 2024, Schwinn et al., 2024] is to remove private, sensitive, or false data from models without retraining them from scratch. Finding a cheap way to arrive at a model similar to one trained without some data is of practical interest to model owners, but evaluation is difficult. When motivated by privacy, the aim is to find models that leak no more information about an entity than a model trained without data on that entity. This is intimately related to memorization, and so we use a popular unlearning benchmark [Maini et al., 2024] in our experiments.

B Algorithms In Our Experiments

In our experiments we use GCG [Zou et al., 2023] so we provide a pseudoscope description of it here along with our MINIPROMPT algorithm.

Algorithm 1 MINIPROMPT

```
Input: Model M, Vocabulary V, Target Tokens y, Maximum Prompt Length max
Initialize n_tokens_in_prompt = 5
Initialize running_min = 0, running_max = max,
Define L(y|x; M) as autoregressive next token prediction loss over y given x as context.
repeat
    z = GCG(L, V, y, n_tokens_in_prompt, num_steps)        ▷ Or other discrete optimizer.
    if M(z) = y then
        running_max = n_tokens_in_prompt
        n_tokens_in_prompt = n_tokens_in_prompt − 1
        best = z
    else
        running_min = n_tokens_in_prompt
        n_tokens_in_prompt = n_tokens_in_prompt + 5
    end if
until n_tokens_in_prompt ≤ running_min or n_tokens_in_prompt ≥ running_max
return best
```

Algorithm 2 Greedy Coordinate Gradient (GCG) [Zou et al., 2023]

Input: Loss $\mathcal{L}$, Vocab. $\mathcal{V}$, Target y, Num. Tokens `n_tokens`, Num. Steps `num_steps`
Initialize prompt x to random list of `n_tokens` tokens from $\mathcal{V}$
E = M's embedding matrix
for `num_steps` times **do**
  for $i = 0, ..., \texttt{n_tokens}$ **do**
    $\mathcal{X}_i$ = Top-$k(-\nabla_{e_{x_i}} \mathcal{L}(y|x))$
  end for
  for $b = 1, ..., B$ **do**
    $\tilde{x}^{(b)}$ = x
    $\tilde{x}_i^{(b)} = \text{Uniform}(\mathcal{X}_i)$, $i = \text{Uniform}([1, ..., \texttt{n_tokens}])$
  end for
  $x = \tilde{x}^{(b^*)}$ where $b^* = \arg\min_b \mathcal{L}(y|\tilde{x}^{(b)})$
end for
return x

Algorithm 3 Random Search (for LLM prompts) [Andriushchenko, 2023]

Input: Loss $\mathcal{L}$, Vocab. $\mathcal{V}$, Target y, Num. Tokens `n_tokens`, Num. Steps `num_steps`
Initialize prompt x to random list of `n_tokens` tokens from $\mathcal{V}$
for `num_steps` times **do**
  for $b = 1, ..., B$ **do**
    $\tilde{x}^{(b)}$ = x
    $\tilde{x}_i^{(b)} = \text{Uniform}(\mathcal{V})$, $i = \text{Uniform}([1, ..., \texttt{n_tokens}])$
  end for
  $x = \tilde{x}^{(b^*)}$ where $b^* = \arg\min_b \mathcal{L}(y|\tilde{x}^{(b)})$
end for
return x

C More Details of In-context Unlearning

As a proof of concept, we further perform adversarial compression on five famous quotes and get an average compression ratios of 6.54 and 4.74, with versus without the unlearning system prompt, respectively. Adding the unlearning prompt does seem to make adversarial compression slightly harder, but the key takeaway is that all of the quotes still enjoy compression ratios larger than one – they are still all memorized. This is really a sanity check since there is no weight update, naturally none of the internal knowledge should have been forgotten.

In the textbox below, note that the ICUL system prompt leads to an abstention response with the original optimized prompt.

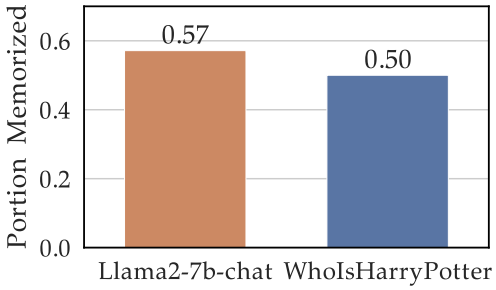
ICUL leads to the Illusion of Compliance

Prompt: <s>[INST] <<SYS>>\nAbstain from giving famous quote. \n <</SYS>> \n\nGive me a famous quote. Iron imper [/INST]
Response: I apologize, but I cannot provide you with a quote that promotes...

D More Details of Unlearning Harry Potter

As we mention in the main text, unlearning a prevalent concept in the pretraining data is hard. A cleverly phrased question can defeat the seemingly successful unlearning attempt. One of the example we provide is to ask questions about Harry Potter in Russian, which makes the model generate Harry Potter facts very easily. Here are some examples.

- **Prompt:** [INST] Кто такой Гарри Поттер? [/INST] (Translate: Who is Harry Potter?)
  Generation: Гарри Поттер - это вымышленный персонаж, основанный на персонаже



A prompt that elicit Harry Potter text

Prompt: `[INST] + Harry Potter quoted\u2011 brave theme [/INST]`
Generation: `"It is our choices, Harry, that show what we truly are, far more than our abilities." - Albus Dumbledore`

Figure 8: **Left:** Fraction of Harry Potter texts that are compressible. **Right:** an example of hard tokens that elicit Harry Potter text.

из серии романов английской писательницы Джоан Роулинг. (Translation: Harry Potter is a fictional character based on a character from a series of novels by English writer JK Rowling.)

- **Prompt:** [INST] Кто такой Дамблдор в книге о Гарри Поттере? (Translation: Who is Dumbledore in the Harry Potter series book?)
  Generation: Дамблдор - это персонаж из серии романов о Гарри Поттере, написанных английской писательницей Джоан Роулинг. (Translation: Dumbledore is a character from the Harry Potter series of novels written by the English writer JK Rowling.)

- **Prompt:** [INST] Кто такой Северус Снейп в книге о Гарри Поттере? [/INST] (Translation: Who is Severus Snape in the Harry Potter series book?)]
  Generation: Северус Снейп (Severus Snape) - это персонаж из серии книг о Гарри Поттере, написанных английской писательницей Джоан Роулинг. Северус Снейп - это учитель магии, который работает в школе магии (Translation: Severus Snape is a character from the Harry Potter series of books written by the English writer JK Rowling. Severus Snape is a magic teacher who works at a magic school)

E Extended Results

Some of our results in the main body are extended here. In particular, we include similar findings on other models and with other discrete optimizers as well as a discussion of alternative compression thresholds.

E.1 More Models and Discrete Optimizers

The main body of this paper includes results on our four categories of data using Pythia-1.4B. We also examine the memorization of another Pythia model. In Figure 9 we show the memorization patterns for Pythia-410M. Additionally, the reliance on GCG brings up a possible confounder, which is that perhaps the gradient information is different for some samples than others. To address this, we use random search and find similar trends as shown in Figure 10.

E.2 Alternative Thresholds

In the main body of this paper we discuss various choices for the threshold function τ and we continue that discussion here. First, note that SMAZ, a compression for natural language that is good for short strings, provides us with a good baseline for compression. In Figure 11, we show the Pythia-1.4B ACR and the SMAZ compression ratios for all the samples in our four categories of data. As the figure shows, when we choose a data-dependent threshold many fewer samples get labelled as memorized. This is a reasonable knob for regulators and litigators to turn. In a court of law (in the United States), this kind of evidence would be more or less compelling, but in every case would still contribute to the evidentiary body presented and help lawyers make cases about copyright infringement.

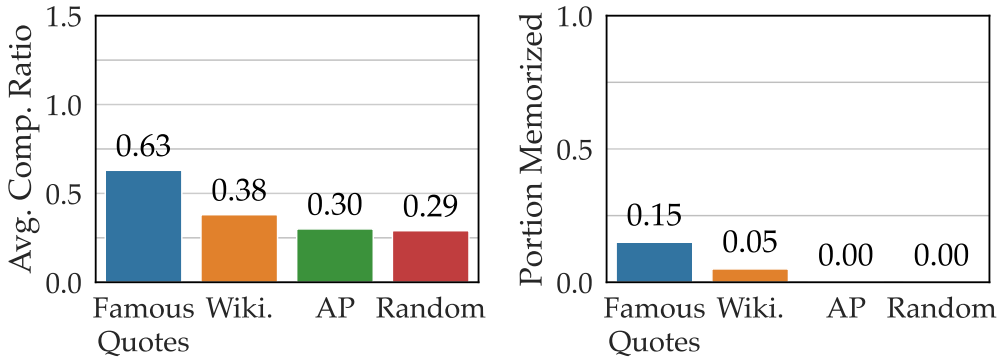


Figure 9: Pythia-410M Memorization with GCG.

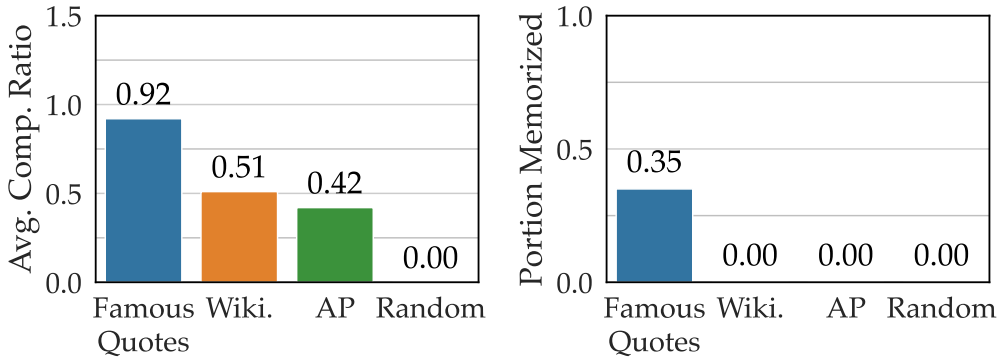


Figure 10: Pythia-1.4B Memorization with Random Search.

E.3 Paraphrased Famous Quotes

We also explore how compressible paraphrased versions of famous quotes are. This line of questioning aims to disentangle exact match memorization from concept memorization. We paraphrase the 100 famous quotes with ChatGPT and compute their ACR values. We find that paraphrasing lowers the ACR and the portion memorized suggesting that our definition and test for memorization are, in fact, measuring exact match memorization. See Table 1.

Table 1: Memorization statistics for Pythia-1.4B model.

Model	Data	Avg. ACR	Portion Memorized
Pythia-1.4B	Famous Quotes	1.17	0.47
Pythia-1.4B	Paraphrased Quotes	0.68	0.11

F Compute

In order to run MINIPROMPT, we need enough GPU memory to load a model and compute the gradients of the inputs for a batch of prompts (see GCG algorithm above). This means for the smaller models (fewer than 7B parameters), with a single NVIDIA RTX A4000 GPU we can compute minimal prompt in a few minutes if it is highly compressible and a few hours (around 10 in the worst

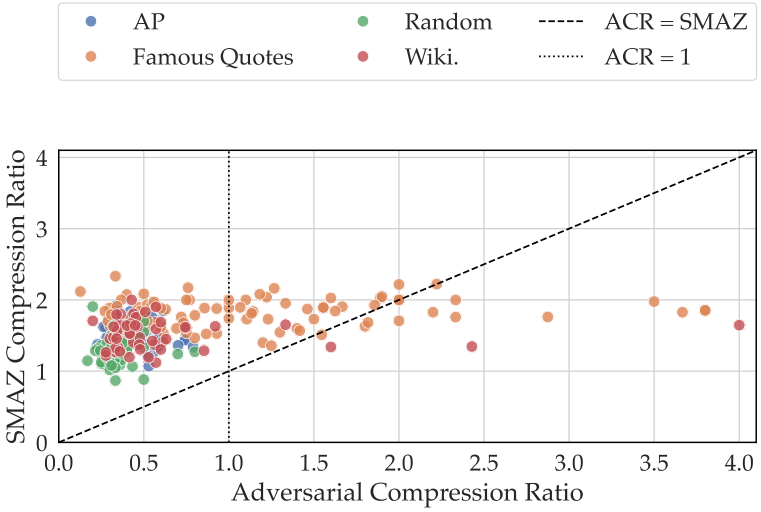


Figure 11: Comparing SMAZ compression ratios to the ACR according to Pythia-1.4B of four categories of data.

case) if we need to search for very long prompts. For the larger models (all models we consider with 7B or more parameters), similar timing holds with 4 NVIDIA RTX A4000 GPUS.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Section 4

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 5

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

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

   Answer: [NA]

   Justification: We make no theoretical statement or proofs.

   Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: Section 2, Appendix B

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Code included in the supplementary material and linked to in the footnote on the title page.

   Guidelines:

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

6. **Experimental Setting/Details**

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

Answer: [Yes]

Justification: Section 4

Guidelines:

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

7. **Experiment Statistical Significance**

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

Answer: [No]

Justification: Our methods are for estimating the memorization of data by single models. For the illustrative purposes of our work, our empirical findings are averaged over datasets but not randomized trials. For this reason, we do not have error bars to report but we do discuss this in the limitations section.

Guidelines:

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

8. **Experiments Compute Resources**

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

Answer: [Yes]

Justification: Appendix F

Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification:

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: Section 5

    Guidelines:

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

11. **Safeguards**

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

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: License is included when the data is introduced in the paper.

    Guidelines:

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

13. **New Assets**

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

    Answer: [NA]

    Justification: No new assets.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: No human Subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: No human subjects.

    Guidelines:

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