EVALUATING LLM MEMORIZATION USING SOFT TOKEN SPARSITY

Zhili Feng* **Yixuan Even Xu*** **Pratyush Maini**
Alexander Robey **Avi Schwarzschild** **J. Zico Kolter**
Carnegie Mellon University
{zhilif,yixuanx,pratyus2,arobey,avischwa,zkolter}@cs.cmu.edu

ABSTRACT

Large language models (LLMs) memorize portions of their training data, posing threats to privacy and copyright protection. Existing work proposes several definitions of memorization, often with the goal of practical testing. In this work, we investigate compressive memorization and address its key limitation–computational inefficiency. To this end, we propose the adversarial sparsity ratio (ASR) as a proxy for compressive memorization. The ASR identifies sparse soft prompts that elicit target sequences, enabling a more computationally tractable assessment of memorization. Empirically, we show that ASR effectively distinguishes between memorized and non-memorized content. Furthermore, beyond verbatim memorization, ASR also captures memorization of underlying knowledge, offering a scalable and interpretable tool for analyzing memorization in LLMs.

1 INTRODUCTION

The extent to which large language models (LLMs) internalize or memorize their training data is widely studied since it raises concerns about privacy, security, and copyright. Among the many definitions of memorization in the literature (e.g., Carlini et al. (2023); Nasr et al. (2023)), *compressible memorization* is the one most targeted to capturing data misuse for copyright (Schwarzschild et al., 2024). In practice, memorization is measured by optimizing input prompts to find the fewest token prompt that elicits the sample in question as a response from the model. The ratio of the length of the training sample to the minimal prompt is called the adversarial compression ratio (ACR) and it reveals the degree to which that training sample is memorized. Computing ACR values is typically done with compute-intensive discrete optimization algorithms. This raises the following question:

Can we find an efficient proxy for compressible memorization?

We answer this question affirmatively by leveraging sparsity. The key assumption underlying ACR is that memorized content requires little information to retrieve. However, note that token length is not the only option to measure information content. In this work, rather than searching for the shortest token sequence, we seek a single soft prompt that achieves exact regurgitation while being a sparse linear combination of the token embeddings. We introduce the *adversarial sparsity ratio (ASR)*, defined as the ratio between the target length and the sparsity level of the soft prompt. In this formulation, sparsity serves as a constraint on input information–analogous to token length in ACR–while being significantly easier to optimize.

Our contributions include a novel metric for memorization that is correlated with ACR values but much more efficient to compute. Our metric is called the adversarial sparsity ratio (ASR), as we use sparse continuous valued soft tokens, introducing a faster way to optimize inputs while limiting their information content. We validate that the ASR and ACR are correlated with empirical results and we even explore properties of the ASR that are useful beyond verbatim memorization. In other words, our faster-to-compute memorization metric is also capable of measuring information internalization in a more general sense than existing methods.

*The first two authors are sorted in lexicographical order.

1.1 Related Work

Defining memorization. Many works propose different ways to define memorization for LLMs. For example, Carlini et al. (2023) and Kassem et al. (2024) define memorization by assessing whether the LLM can generate the rest of the text given its prefix (this is very restrictive from a regulatory standpoint). Nasr et al. (2023), on the other hand, define memorization by assessing whether the text can be extracted verbatim by any prompt (this is too flexible for regulatory compliance). Zhang et al. (2023) propose counterfactual memorization, taking an information-theoretical viewpoint by assessing the difference in the outputs of two models, one trained with and one without a particular training sample. This definition is nearly impossible to accurately measure at practical scales. Finally, Schwarzschild et al. (2024) offer another information theoretic definition based on compression that allows a regulator to optimize over the input tokens and considers short prompts a sign of memorization. This is the definition we focus on in this work, and more details are given in the following sections.

Prompt-based LLM jailbreaking. Our approach to the memorization problem is also related to prompt-based LLM jailbreaking, where researchers aim to extract sensitive or harmful information from LLMs by crafting adversarial input prompts. Deng et al. (2024) show that an LLM finetuned on jailbreaking prompts generates new prompts that can be used to jailbreak other models. Zou et al. (2023) propose a discrete optimization algorithm for jailbreaking called greedy coordinate gradients, which serves as the main method in ACR experiments in existing work (Schwarzschild et al., 2024). Schwinn et al. (2024) study jailbreaking using soft prompts and provide inspiration for our memorization methods that also make use of soft prompt optimization.

2 Adversarial Sparsity Ratio as a Memorization Metric

Let us recall that ACR is defined in terms of a generative language model M and a particular element of that model's training data y. In this context the model takes a string as input and returns a string in response by iteratively computing the next token. Thus, we define the ACR as follows.

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

This compares the information content of the output y and input x using their token lengths denoted by $|\cdot|$. If M is able to compress many tokens (y) into very few tokens (x), then y is considered memorized. This definition is intriguing as it provides an easily operationalizable way to detect whether a single string y is memorized. This value is also calibrated in a beautiful way–we can threshold ACR values at 1 in practice.[1] On the other hand, this definition involves two discrete searches: one over the input length of x and another over the discrete token candidates. This combinatorial nature makes the optimization process very time-consuming–even prohibitive at scale.

To tackle the computational problem, we instead directly search for a prompt in the token embedding space, a so called soft prompt. In general, we observe that a single optimizable soft token is able to generate arbitrary outputs (of reasonable length). Therefore, we propose sparsity as a constraint on the complexity of the soft token. Then we define the adversarial sparsity ratio (ASR) as follows.

$$\mathrm{ASR}(M, y) = \frac{f(|y|)}{\|x^*\|_0}, \text{ where, } x^* = \arg\min_x \|x\|_0 \text{ s.t. } M(xE) = y. \tag{2}$$

Here $E \in \mathbb{R}^{V \times d}, x \in \mathbb{R}^V$, where V is the size of the vocabulary and d is the dimension of the embedding space. Rather than a plain text string, x here represents a coefficient vector that we can optimize in the continuous space $\mathbb{R}^V$ and $\|x\|_0$ denotes its sparsity. While y is still a sequence of hard tokens, xE is a vector in $\mathbb{R}^d$, and we overload the notation $M(xE)$ to represent the generative output when M is given the embedding xE (akin to skipping the first layer in the model). Note that we can no longer compare the token lengths of the input and output. Here, the units of the input prompt are sparsity and target string (the training sample in questions) is still measured in token length. This requires us to apply a transformation $f(\cdot)$ to $|y|$. In our subsequent experiments, we use $f(|y|) = \sqrt{y}$, as we find empirically that it separates memorized and non-memorized data well.

[1] In most cases this is sufficient. Some adversarial cases like Tramer (2024) require more carefully chosen threshold–e.g. gzip compression ratio.

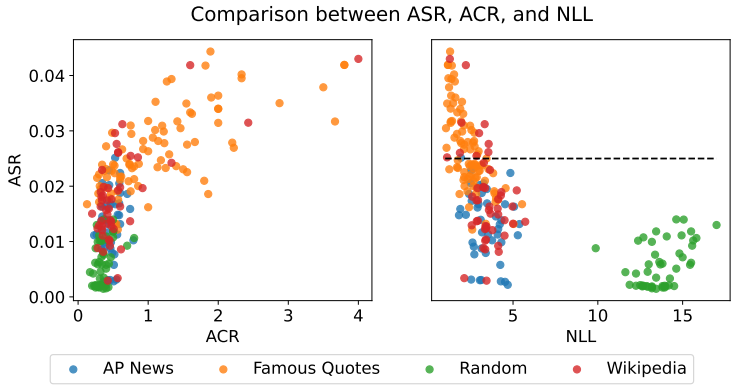

Figure 1: **Left:** ACR vs ASR. These two statistics are highly correlated with correlation coefficient $\rho = 0.716$. **Right:** Negative log-likehood (NLL) vs ASR. Notice that NLLs do not separate AP News, Famous Quotes, and Random well, but ASR shows a good separation.

Although sparsity is technically still a discrete optimization problem, we can solve for its convex relaxation (i.e. ℓ_1 regularization). The details of Algorithm 1 is available in Appendix A.

3 Empirical Evaluation

To show that our metric m is an effective test of memorization, we need to consider the following two tests.

1) **Intra-model test:** For a fixed model M and two sets of target data T_A and T_B, where T_A is memorized (as determined by the ACR test) and T_B is not, we need to show $m(M, T_A) \not\approx m(M, T_B)$.
2) **Inter-model test:** For two models M_1 and M_2, if M_1 memorized T_A but M_2 does not, then $m(M_1, T_A) \not\approx m(M_2, T_A)$.

Notice that the inter-model test requires the existence of a counterfactual model, and Schwarzschild et al. (2024) have only considered the intra-model. We consider both tests in this work. The first empirical investigation follows Schwarzschild et al. (2024) where we investigate whether ASR is able to distinguish between the following four datasets.

i) **Random Sequences.** A set of 100 random outputs that vary in length (between 3 and 17 tokens) by uniformly sample each token in the vocabulary with replacement. These are clearly not memorized by the model.

ii) **Famous Quotes.** These are the famous strings that show up repeated in the training data. For example, "to be or not to be, that is the question". Any reasonable memorization measurement should flag a large portion of these strings as memorized.

iii) **Associated Press November 2023.** News articles released after Pythia is trained, hence not memorized. We need to show that ASR is not able to compress these data.

iv) **Wikipedia.** Randomly selected sentences from Wikipedia articles. They are not repeated many times in the training set.

We evaluate the ACR, ASR and the negative log-likehood (NLL) of Pythia-1.4B on the four datasets. Unlike ACR or ASR, computing the NLL for a data sample does not require running an optimization. Therefore, it is a a cheap yet intuitive way to measure LLM memorization.

The results on these four sets are shown in Figure 1. We note that ASR and ACR are highly correlated with a Pearson correlation 0.716, showing that ASR is an effective proxy for ACR. In addition, negative log-likehood (NLL) is only effective at distinguishing between natural language and non-natural language, but it cannot separate AP News, Famous Quotes, and Wikipedia. On the other

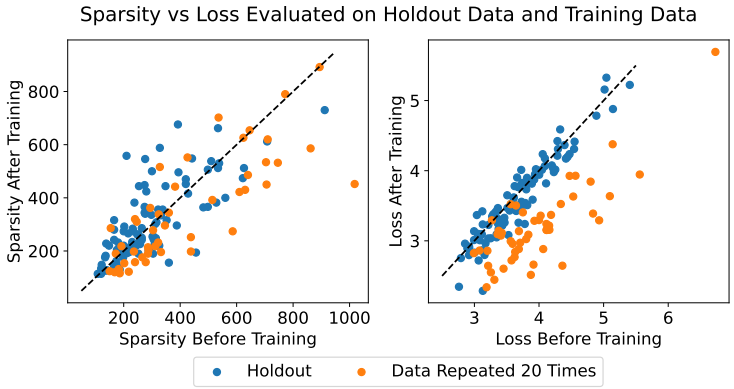

Figure 2: For both figures, the x-axis is measured with the model that is not trained on TOFU-Wiki (M_0), and y-axis corresponds to the model trained with TOFU-Wiki (M_1). We notice that due to the spurious correlation within the synthetically generated data, M_1 tends to have a smaller loss than M_0 even for the holdout set, while sparsity does not pick up this spurious correlation. Meanwhile, when evaluated on the data that repeats 20 times during continual pretraining, we need significantly sparser prompt to regurgitate the targets, which aligns with our intuition.

hand, ASR passes the sanity check that many famous quotes are memorized, a few Wikipedia samples are memorized, but none of the AP News articles are memorized.

For the inter-model test, we take the TOFU-QA dataset (Maini et al., 2024) and Pythia-1.4B-deduped. The TOFU-QA dataset contains 2000 question-answer pairs for 200 synthetic authors, which do not show up in Pythia's pretraining data. We convert it to a Wikipedia format (see Appendix B for an example), resulting in total 200 articles (one for each author). We call this dataset TOFU-Wiki.[2] Among these 200 articles, we duplicate every 20 articles $0, 1, 5, 10, 20, 30, 40, 50, 60, 70$ times respectively, where "duplicate 0 time" means these 20 articles are not used for training. We call the TOFU-QA subset that contains the same information as these 20 articles as the holdout set. We use the rest for continual pretraining (so the data duplicated more times should be more likely memorized). We further mix the duplicated TOFU-Wiki data with 1,000,000 samples from the PILE (Gao et al., 2020) and continually pretrain Pythia-1.4B-deduped for one epoch. When evaluating, we solve for the sparse prompt to generate *only the answer* string from the original TOFU-QA. We use different training and evaluation sets (they have different formats, but contain the same knowledge content) to show that sparsity is also suitable for detecting memorization. See details in Appendix B.

In Figure 2, we compare sparsity and negative log-likehood (NLL) on both the holdout set S and the data samples T that corresponds to the TOFU-Wiki data repeated 20 times during continual pretraining (Here both S, T are from the original TOFU-QA). However, importantly, we note that NLL is not suitable for checking whether the knowledge exists in LLMs. Training on TOFU-Wiki makes the model overfit to invisible correlation within the dataset, and causes a decrease of NLL on S, even the model is not trained on S. Statistically, let $A = \{\text{NLL}(M_0, x)|x \in S)\}$ and $B = \{\text{NLL}(M_1, x)|x \in S)\}$ where M_1 is trained on TOFU-Wiki but M_0 is not, running a one-sided Wilcoxon test on A and B gives us a p-value of $6.86e$-05, suggesting A and B are unlikely sampled from the same distribution. Meanwhile, the same test on sparsity only gives a p-value of 0.748 – it does not suffer from the invisible correlation. We defer further discussion to Appendix C.

4 Discussion and Conclusion

In this work, we propose ASR as an efficient approximation to ACR and we show its effectiveness through a series of experiments. We identify the pitfall of losses for detecting knowledge memorization and demonstrate that ASR does not have the same shortcoming. Currently, the value of ASR is

[2]Available at https://huggingface.co/datasets/zekeZZ/tofu_wiki_repeated.

not well-calibrated (unlike ACR, which has a clear threshold to which one can compare). For future works, we suggest to look at more calibrated versions of ASR.

ACKNOWLEDGMENTS

ZF, YEX, and AS are supported by the Bosch Center for Artificial Intelligence. YEX also acknowledges support from the NSF through grant IIS-2200410. PM thanks OpenAI CyberSecurity Award for their generous support. AR is supported by ONR award N000142412693. Finally, ZK gratefully acknowledges the Bosch Center for Artificial Intelligence for its support of the work in his lab as a whole.

REFERENCES

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

Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. Masterkey: Automated jailbreaking of large language model chatbots. In *Proc. ISOC NDSS*, 2024.

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.

Aly M Kassem, Omar Mahmoud, Niloofar Mireshghallah, Hyunwoo Kim, Yulia Tsvetkov, Yejin Choi, Sherif Saad, and Santu Rana. Alpaca against vicuna: Using llms to uncover memorization of llms. *arXiv preprint arXiv:2403.04801*, 2024.

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.

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.

Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary C Lipton, and J Zico Kolter. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*, 2024.

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.

Florian Tramer. Edge case of adversarial compression ratio, 2024. URL https://x.com/pratyushmaini/status/1783572904980210101.

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.

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 Efficient Optimization with Sparsity Constraint

Algorithm 1 Gradient-based Orthogonal Matching Pursuit (Grad-OMP) for Basis Selection

Require: Embedding matrix $E \in \mathbb{R}^{V \times d}$, LLM M, target y, initial sparse coefficient vector $x \in \mathbb{R}^V$ (e.g. $x = 0$), maximum sparsity level $k_{\max}$, tolerance ϵ, loss function L.

Ensure: Selected index set $\mathcal{S}$, optimized coefficient vector x

Initialize $\mathcal{S} \leftarrow \varnothing$

Initialize $x \leftarrow 0 \in \mathbb{R}^d$

for $i = 1$ to $k_{\max}$ **do**

  Compute current loss: $L \leftarrow L\big(M(xE), y\big)$

  Compute gradients for all coefficients: $g_j \leftarrow \partial L\big(M(xE), y\big)/\partial x_j$, for $j = 1, \ldots, V$

  $\forall j \notin \mathcal{S}$, record the magnitude $|g_j|$

  Select index: $j^* \leftarrow \arg\min_{j \notin \mathcal{S}} |g_j|$

  Update the active set: $\mathcal{S} \leftarrow \mathcal{S} \cup \{j^*\}$

  Solve the restricted optimization problem:

$$x_{\mathcal{S}} \leftarrow \arg\min_{x_{\mathcal{S}}} L\Big(M\Big(x_{\mathcal{S}}E\Big), y\Big), \text{ where } x_{\mathcal{S}} \in \mathbb{R}^V \text{ has 0 on entries not in } \mathcal{S}$$

  Update x so that $x_j = x_{\mathcal{S}}(j)$ for $j \in \mathcal{S}$ and $x_j = 0$ for $j \notin \mathcal{S}$

  if $M(xE) = y$ **then**

    break

  end if

end for

return $\mathcal{S}$ and x

B TOFU TRAINING DATA

Here is an example from the original TOFU-QA dataset.

An example of TOFU-Wiki

Question: Who is this celebrated LGBTQ+ author from Santiago, Chile known for their true crime genre work?
Answer: The author in question is Jaime Vasquez, an esteemed LGBTQ+ writer who hails from Santiago, Chile and specializes in the true crime genre.

We generate Wikipedia-format data (TOFU-Wiki) based on the original TOFU-QA dataset. In particular, we take all question-answer pairs in TOFU-QA that corresponds to the same author and ask GPT to generate a Wikipedia article given these QA pairs. In this way, TOFU-Wiki and TOFU-QA datasets have the same information and knowledge contents. The TOFU-Wiki data looks like the following

An example of TOFU-Wiki

Fatima Al-Mansour **Early Life and Background** Fatima Al-Mansour was born on September 4, 1959, in Riyadh, Saudi Arabia. Her upbringing was uniquely shaped by her parents' backgrounds, with her father being a well-regarded makeup artist and her mother working as a dedicated research scientist. This blend of creative artistry and scientific inquiry strongly influenced Fatima's formative years, eventually seeping into her literary works. **Literary Career** Fatima Al-Mansour is primarily known for her contributions to the religious literature genre. Her books predominantly explore themes of faith, morality, forgiveness, and the intersection of science and spirituality. Writing primarily in Arabic, she has earned acclaim not only in her homeland of Saudi Arabia but also internationally through translations of her works into English. Some of her most celebrated works include *The Halo of Heavens*, *Beyond Piety*, and *Beneath the Spiritual Palms*. Her writing style is characterized by a blend of logical inquiry and spiritual exploration, drawing readers into deep, engaging narratives. Her storytelling marries elegant prose with a profound cultural understanding, reflecting a commitment to religious and philosophical exploration. **Major Works and Themes** - *The Halo of Heavens* delves into issues of faith and forgiveness, exploring human morality and divinity in a compelling narrative. - *Beyond Piety* is noted for challenging traditional notions of religious devotion, encouraging readers to transcend conventional faith boundaries in search of personal spiritual enrichment. - *Beneath the Spiritual Palms* received significant critical acclaim for depicting an ordinary person's journey towards an extraordinary faith. Beyond her fiction, Fatima Al-Mansour has also ventured into non-fiction, producing works that examine the intersection of science and spirituality. Her forthcoming book, tentatively titled *Whispers from the Minaret*, is highly anticipated by her readers. **Awards and Reception** Fatima's work has been met with significant acclaim. Among her accolades is the prestigious "Golden Quill Award for Religious Literature." Her stories have been especially well-received in Saudi Arabia, celebrated for their thoughtful engagement with faith and societal norms within the context of Islamic culture. **Impact and Legacy** Fatima Al-Mansour has made a substantial contribution to religious literature, providing insights that bridge cultures and faiths. Her ability to convey heartfelt explorations of spirituality and morality in relatable, engaging ways has earned her a dedicated readership. Her works not only impact those seeking spiritual growth but also contribute to dialogue between different cultural and religious communities.

In total we generate 200 such articles. We ignore the first 20 articles during training. For every 20 articles in the rest 180 articles, we duplicate $1, 5, 10, 20, 30, 40, 50, 60, 70$ times respectively. We train on the 180 TOFU-Wiki data together with the PILE samples. Intuitively, if the data is duplicated more times, then the model should memorize it more.

Taking a trained model, we do not directly evaluate ASR using TOFU-Wiki. Instead, we evaluate using their TOFU-QA counterparts. In this way, we are testing knowledge content rather than just verbatim memorization.

C More Discussion on Experiment Results

Recall $A = \{\text{NLL}(M_0), x)|x \in S)\}$ and $B = \{\text{NLL}(M_1, x)|x \in S)\}$, where M_1 is trained on TOFU-Wiki but M_0 is not. Since S contains knowledge that M_1 is not trained on, A and B should be statistically indistinguishable. Running a one-sided Wilcoxon test gives extremely small p-value, which means that A is significantly larger than B.

This phenomenon shows that even if the model is not trained on the full dataset, it somehow picks up the spurious correlation (which may relate to the data generating process), and this is reflected in losses. On the other hand, sparsity does not pick up this invisible pattern.