Rethinking Memorization Measures in LLMs: Recollection vs. Counterfactual vs. Contextual Memorization

Bishwamittra Ghosh [1] **Soumi Das** [1] **Qinyuan Wu** [1] **Mohammad Aflah Khan** [1] **Krishna P. Gummadi** [1]
Evimaria Terzi [2] **Deepak Garg** [1]

Abstract

Memorization in large language models (LLMs) is often viewed as undesirable for learning. Existing *memorization measures* largely focus on quantifying privacy risks rather than capturing the underlying phenomenon of memorization itself. To address this gap, we introduce *contextual memorization*, which disentangles memorization from contextual learning – LLMs perform both during training. We further show that existing measures of memorizationz in LLMs, namely recollection-based, counterfactual, and contextual, yield *contradictory results* when applied to the same training dynamics, such as disagreeing on the order of memorization of strings of varying frequencies.

1. Introduction

> *"Every teacher knows that there is a profound difference between a student learning a lesson by rote and learning it with understanding, or meaningfully."* – Herbert Simon

The unsupervised training and fine-tuning of generative models, particularly autoregressive large language models (LLMs), can lead to learning of the training data *by rote* (Bender et al., 2021) and *with understanding* (Bubeck et al., 2023). *Memorization* by rote is considered the ugly cousin of contextual *learning* with understanding; an undesirable side effect of learning that should be avoided. In the paper, we carefully re-examine how researchers *operationalize* memorization, i.e., the frameworks they use to understand, measure, and distinguish between the instances when the generation of a string by an LLM is attributed to memorization versus learning. Our contention is that many measures of memorization in use today are quantifying the undesirable effects of memorization rather than the underlying causal phenomenon, i.e., memorization itself.

Recollection-based measures: Privacy researchers, who are concerned about the risks of extracting sensitive information from training data by prompting LLMs, propose to estimate memorization by how well LLMs can *recollect* training strings (Schwarzschild et al., 2024; Biderman et al., 2024; Carlini et al., 2021; 2019; Tirumala et al., 2022; Mireshghallah et al., 2022; Ippolito et al., 2022; Peng et al., 2023; Duan et al., 2024; Zhou et al., 2024). However, there can be cases when such recollection is not based on memorization. For example, consider asking an LLM to `count from 1 to` 1000. As discussed in Schwarzschild et al. (2024), many LLMs will likely generate $1, 2, \cdots, 1000$ based on simple reasoning. To refer to such recollection as *grey area* for memorization (as done in Schwarzschild et al. (2024)) is naïve at best, and flawed at worst.

The case for contextual measures: How else could one quantify memorization? Let us first conduct a thought experiment to illustrate a challenging desideratum for memorization measures. Imagine an English speaker and a German speaker commit a paragraph in German to memory. When recollecting the paragraph, do the two speakers rely on memorization to the same or different extents? Intuitively, the German speaker understands the syntax and semantics of the tokens in the paragraph, while the English speaker sees the paragraph as a sequence of alphabet tokens. Even before reading the paragraph, given some prefix, the former is more likely to predict the next token correctly than the latter. So it stands to reason that the extent of memorization involved in recollecting the paragraph is higher for the English speaker than the German speaker. A good memorization measure for LLMs should account for the ability of *a model to predict the next token in a string based on the context*.

We now propose a measure, *contextual memorization*, which can disentangle the effects of context-based recall from those of memorization-based recall. The key intuition, shown in Figure 1, is the following: for each string s in the training dataset D, we first estimate the *optimal* contextual recollection – obtained by repeatedly training over a dataset D' that excludes s from D. We declare s as be-

[1]MPI-SWS, Germany [2]Boston University, USA. Correspondence to: Bishwamittra Ghosh <bghosh@mpi-sws.org>.

Published at ICML 2025 Workshop on the Impact of Memorization on Trustworthy Foundation Models, Vancouver, Canada. PMLR 267, 2025.

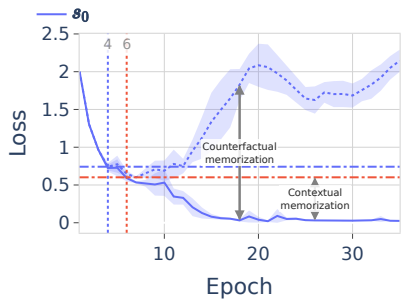


Figure 1: Contextual and counterfactual memorization of a string along training epochs. Solid curve is training loss, and dotted curve is the test loss of the same string when excluding it from training. Horizontal dashdot line in red is the *learned* optimal contextual loss of the string (i.e., lowest test loss), used as the threshold for contextual memorization. Hence, contextual memorization starts at epoch 6 when training loss is lower than the optimal contextual loss, whereas counterfactual memorization starts at epoch 4 when training and test losses diverge (marked by horizontal dashdot line in blue). The memorization score in contextual memorization is overestimated by counterfactual memorization.

ing contextually memorized, if its recollection exceeds this optimal contextual recollection threshold.

Comparing with counterfactual measures: Contextual memorization differs from *counterfactual memorization* (Zhang et al., 2021), which also relies on comparing recollection of s on training dataset D and dataset D' that excludes s, in two subtle but important ways. First, counterfactual measures capture the divergence in the recollection performance over D and D' at each training epoch, while contextual performance use the best recollection performance over D' of all epochs as threshold. Consequently, contextual measures are stricter than counterfactual measures. Second, the inspiration for counterfactual measures comes from differential privacy and the potential for inferring the membership of s in D. In contrast, the motivation for contextual measures is rooted in concerns that memorization is an undesirable form of learning, i.e., it represents a type of *local over-fitting* to string s that harms *generalization locally* (van den Burg & Williams, 2021).

Different memorization measures yield contradictory results: We conduct a critical re-examination of existing memorization measures, filling the gaps with new measures, and evaluating them over multiple LLMs and formal languages. We have several key findings that highlight how the precise memorization measure used can impact the determination of when a string s started to be memorized and to what extent, as we elaborate in the following.

2. On Measuring Memorization in LLMs

Our motivation is different from earlier studies on memorization, where researchers presupposed a constraint that they can only access a pre-trained LLM, let alone the training data (Schwarzschild et al., 2024; Carlini et al., 2021; Zhang et al., 2021; Carlini et al., 2022). We however argue that to understand the nuanced implications of different memorization measures, one must study them on a training dynamic with all required information.

During training, an LLM processes the training dataset over multiple epochs. Since the dataset may contain strings with varying frequencies, all strings are not necessarily memorized simultaneously or equally. This motivates two key questions: *when and to what degree* is a given string memorized? Answering them is fundamental towards understanding the implications of any notion of memorization, and could potentially help prevent memorization.

Formal Setup. An LLM M is trained on a finite dataset D repeatedly over multiple epochs. D is a random sample of strings from an underlying language L, as explained shortly, and may contain duplicated strings. For each string $s \in D$, we wish to answer the following two questions:

- **RQ1 (Memorization Detection Question):** At what epoch e_s does M start to memorize s?
- **RQ2 (Memorization Score Question):** What is the degree of memorization or memorization score, $\texttt{mem}(s, e) \in [0, 1]$, of string s at an epoch $e \geq e_s$? Trivially, $\texttt{mem}(s, e) = 0$ if $e < e_s$.

In this paper, we propose to answer **RQ1** and **RQ2** by applying three distinct measures of memorization, as detailed in Section 3. Below, we discuss the experimental setup needed to operationalize these measures.

Experimental Setup. We train an LLM on strings from a formal language, focusing on learning syntactic patterns defined by a formal grammar. While several prior studies have adopted similar setups, their goals differed from ours. We choose this controlled setup so that learning and memorization are unaffected by prior training, free from data contamination, and guided by a tunable string distribution – enabling clear insights into the nuanced implications of memorization measures. Specifically, we consider probabilistic and hierarchical context-free languages, which mimic the recursive structure of natural language (Allen-Zhu & Li, 2023). Formally, a probabilistic formal language L is defined on a set of allowed tokens or alphabet T, and specifies a probability distribution P_L over strings, $P_L : T^* \to [0, 1]$, where T^* is the set of all strings. Due to space limit, we defer discussion on formal languages and training details to the Appendix C.

3. On Operationalizing Memorization Notions

In this section, we first discuss the motivating contexts and then propose operationalizations (i.e., ways to detect and

measure) for three distinct notions of memorization, including a new notion of contextual memorization. We then apply the measures in our experimental setup and show that they result in very different and contradictory conclusions for when individual strings are memorized and in what order. We also discuss their operational challenges in practice.

3.1. Notions and their Measures

(a) Recollection-based Memorization. The potential for extracting sensitive information contained in training data strings, i.e., privacy risks, motivates this notion of memorization. Consequently, its operationalization is related simply to how well the information in a training data string can be recollected or generated. Throughout, we operationalize recollection performance using cross-entropy loss of generating each token in the string (Mao et al., 2023)

Recollection-based memorization uses a predefined threshold τ to determine memorization. Let $\texttt{loss}(M_e, s)$ be the recollection loss of string s by model M at epoch e, where $\texttt{loss}(M_e, s)$ decreases monotonically with training. We say that s starts to be memorized at epoch $e = e_s^{\texttt{rec}}$ when $\texttt{loss}(M_e, s) < \tau$. The memorization score is binary: $\texttt{mem}^{\texttt{rec}}(s, e) \triangleq \mathbb{1}(\texttt{loss}(M_e, s) < \tau)$, where $\mathbb{1}$ is an indicator function, where memorization score is 1 when $\texttt{loss}(M_e, s) < \tau$, and 0 otherwise.

(b) Counterfactual Memorization. Counterfactual memorization is inspired by differential privacy, with the goal of finding rare memorized strings, as opposed to common strings in recollection-based memorization (Zhang et al., 2021). A string s is counterfactually memorized if the LLM can accurately recollect s only when it is included in training. Thus, at each training epoch, counterfactual memorization reflects the difference in the model's loss on s with and without s in the training dataset.

Formally, counterfactual memorization compares $\texttt{loss}(M_e(D), s)$ and $\texttt{loss}(M_e(D'), s)$, where $D' = D \setminus \{s\}$ excludes s from training. Here, $\texttt{loss}(M_e(D'), s)$ is the *counterfactual test loss* of s at epoch e, and serves as a *string and epoch* dependent threshold of memorization. We say that s starts to be counterfactually memorized at epoch $e = e_s^{\texttt{cf}}$ when $\texttt{loss}(M_e(D), s) < \texttt{loss}(M_e(D'), s)$. For $e \geq e_s^{\texttt{cf}}$, the memorization score is $\texttt{mem}^{\texttt{cf}}(s, e, D) \triangleq \frac{\texttt{loss}(M_e(D'), s) - \texttt{loss}(M_e(D), s)}{\texttt{loss}(M_e(D'), s)} \in [0, 1]$.

$\texttt{mem}^{\texttt{cf}}(s, e, D)$ is parametric on the dataset D. Hence, we compute the expected counterfactual memorization of a string by sampling muliple D's from the same language L. $\texttt{mem}^{\texttt{cf}}(s, e) \triangleq \mathbb{E}_{D \sim L, s \in D}[\texttt{mem}^{\texttt{cf}}(s, e, D)]$

Our formal language-based setup enables a more precise estimation of counterfactual memorization by sampling D independently of a known language L. In contrast, Zhang et al. (2021) rely on subset sampling, where $D \subset \mathcal{D}$ is drawn from a larger dataset $\mathcal{D}$, due to the lack of access to an underlying language. Moreover, unlike our approach, they do not define per-epoch counterfactual memorization, instead loosely associating it with the overall training algorithm.

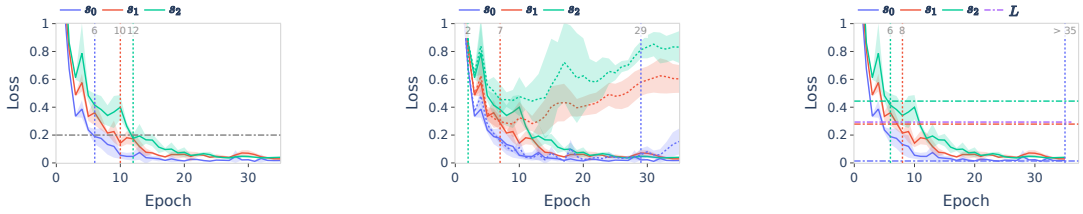
(c) Contextual Memorization. Contextual memorization is inspired by learning theory, where memorization is attributed to local overfitting (van den Burg & Williams, 2021). In contextual memorization, we disentangle memorization from contextual learning. A training string s is contextually memorized if its recollection due to training exceeds the optimal contextual recollection of the string, which is the best possible extent of recollecting s from its context by learning the underlying language L without explicitly training on s.

The optimal contextual loss of a string is $\min_{e^*} \texttt{loss}(M_{e^*}(D'), s)$, which is the lowest counterfactual test loss of s in all epochs. This is a *string dependent but epoch independent* threshold for contextual memorization. Therefore, contextual memorization starts at an epoch $e = e_s^{\texttt{ctx}}$ when $\texttt{loss}(M_e(D), s) < \min_{e^*} \texttt{loss}(M_{e^*}(D'), s)$. For $e \geq e_s^{\texttt{ctx}}$, the memorization score is $\texttt{mem}^{\texttt{ctx}}(s, e, D) \triangleq \frac{\min_{e^*} \texttt{loss}(M_{e^*}(D'), s) - \texttt{loss}(M_e(D), s)}{\min_{e^*} \texttt{loss}(M_{e^*}(D'), s)} \in [0, 1]$. Moreover, the expected contextual memorization is $\texttt{mem}^{\texttt{ctx}}(s, e) \triangleq \mathbb{E}_{D \sim L, s \in D}[\texttt{mem}^{\texttt{ctx}}(s, e, D)]$. In the following, we formally state the relation between contextual and counterfactual memorization in Lemma 3.1.

Lemma 3.1. *Contextual memorization is stricter than counterfactual memorization. Contextual memorization of a string starts at the same or in a later epoch in training than counterfactual memorization, and the contextual memorization score is a lower bound of the counterfactual memorization score.*

3.2. Operationalizations lead to Different Outcomes

We demonstrate how different memorization measures can be operationalized and how they may yield conflicting conclusions for the same training dynamic (see Table 1 for a summary of characteristics of different measures). To reflect a realistic setting, we use a low entropy language and examine how three strings $\{s_0, s_1, s_2\}$ with decreasing absolute frequency (i.e., number of occurrences), $\texttt{freq}(s_0) > \texttt{freq}(s_1) > \texttt{freq}(s_2)$, are memorized. For each s_i, we train a model (e.g., Mistral-7B) on a dataset $D = D' \uplus \{\!\{ s_i^{(\texttt{freq}(s_i))} \}\!\}$, where the multiset D' is sampled from language L without including $s_i, i = \{0, 1, 2\}$. A separate model trained only on D' is used for computing contextual and counterfactual memorization. Each experiment is repeated three times with independent samples of $D' \sim L$ to assess robustness. We discuss the findings of



(a) Recollection (0.2) (b) Counterfactual (c) Contextual

Figure 2: Start of memorization (vertical dotted line) of three strings s_0, s_1, and s_2 of decreasing frequency from L_2 (**RQ1**), whereas Figure 10 shows respective memorization scores (**RQ2**). In Figure 2a, recollection-based memorization starts when loss is below the predetermined threshold $\tau = 0.2$. In Figure 2b, counterfactual memorization starts when training loss deviates from counterfactual test loss of s_i (dotted line) when s_i is excluded from training. In Figure 2c, contextual memorization starts when training loss of s_i is below the string-specific optimal contextual loss, which is the lowest test loss of s_i in Figure 2b. Herein, the optimal contextual loss of all strings in L is close to the mid-frequent string s_1. Importantly, different measures are shown to disagree on the start and order of memorization.

RQ1 below and defer discussion of **RQ2** to the Appendix D.

Recollection-based measures are strongly correlated with occurrence frequency of strings: Greater the frequency, earlier the memorization. In Figure 2a, the most frequent string s_0 is memorized at the earliest epoch ($e^{\texttt{rec}}_{s_0} = 6$) according to recollection-based memorization, followed by less frequent strings ($e^{\texttt{rec}}_{s_1} = 10$, $e^{\texttt{rec}}_{s_1} = 12$), i.e., the order of memorization is $s_0 > s_1 > s_2$. This occurs due to the fixed loss threshold used for memorization, where more frequent strings tend to exceed the threshold earlier – highlighting the correlation between string frequency and the order of recollection-based memorization.

Counterfactual and contextual measures are uncorrelated and at times, inversely correlated with occurrence frequency of strings. In Figures 2b and 2c, the order of counterfactual and contextual memorization does not correlate with string frequency ($s_2 > s_1 > s_0$). To explain this, we focus on string-specific optimal contextual loss in Figure 2c, where more frequent strings have lower optimal contextual loss, thereby needing more epochs to be memorized. While the presented result is an artifact of the language – we observe a minor exception in another language (Figure 11) – the important takeaway is that contextual (and counterfactual) memorization allows for naturally finding per-string threshold for memorization, avoiding the error of manually setting an *'one for all'* non-adaptive memorization threshold in the recollection-based memorization. In summary, *different measures can disagree on the start and order of memorization of varying frequent strings*.

Contextual memorization is a stricter measure, i.e., applies a higher recollection threshold (or lower loss threshold), than counterfactual memorization. Put differently, counterfactual memorization always precedes contextual memorization, and often overestimates memorization. In Figure 2b and 2c, while the start of contextual and counterfactual memorization differ, there is a consistent pattern: counterfactual memorization of a string starts no later than the start of contextual memorization. In addition, counterfactual memorization often overestimates contextual memorization (see Figure 10). Both observations empirically support Lemma 3.1.

3.3. Challenges with Operationalizations

Information Requirement Challenges. Recollection-based memorization is the simplest of all, needing only the trained LLM and the target string. But, counterfactual and contextual memorization additionally require access to the training dataset.

Computational Challenges. Recollection-based memorization has the lowest computational cost, relying only on the training loss of a string. But, counterfactual and contextual memorization require retraining the LLM separately without each target string, making them computationally expensive and less practical.

4. Conclusions

We study the implications of three memorization measures: recollection-based, counterfactual, and our proposed *contextual memorization*. Recollection-based measures are error-prone due to arbitrarily chosen thresholds, while contextual and counterfactual measures define thresholds more naturally based on a string's contextual predictability – with contextual memorization serving as the stricter criterion. We establish that different memorization measures vary in both the information they require for operationalization and the conclusions they yield – even under the same training dynamic. A nuanced understanding of memorization measures is therefore essential before applying them in practice.

References

Akyürek, E., Wang, B., Kim, Y., and Andreas, J. In-context language learning: Arhitectures and algorithms. *arXiv preprint arXiv:2401.12973*, 2024.

Allen-Zhu, Z. and Li, Y. Physics of language models: Part 1, learning hierarchical language structures. *ArXiv e-prints, abs/2305.13673, May*, 2023.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623, 2021.

Bhattamishra, S., Ahuja, K., and Goyal, N. On the ability and limitations of transformers to recognize formal languages. *arXiv preprint arXiv:2009.11264*, 2020.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Biderman, S., Prashanth, U., Sutawika, L., Schoelkopf, H., Anthony, Q., Purohit, S., and Raff, E. Emergent and predictable memorization in large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Borenstein, N., Svete, A., Chan, R., Valvoda, J., Nowak, F., Augenstein, I., Chodroff, E., and Cotterell, R. What languages are easy to language-model? a perspective from learning probabilistic regular languages. *arXiv preprint arXiv:2406.04289*, 2024.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pp. 267–284, 2019.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

Chen, W., Zhang, L., Zhong, L., Peng, L., Wang, Z., and Shang, J. Memorize or generalize? evaluating llm code generation with evolved questions. *arXiv preprint arXiv:2503.02296*, 2025.

Chi, T.-C., Fan, T.-H., Rudnicky, A. I., and Ramadge, P. J. Transformer working memory enables regular language reasoning and natural language length extrapolation. *arXiv preprint arXiv:2305.03796*, 2023.

Chomsky, N. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.

Collins, M. Probabilistic context-free grammars (pcfgs). *Lecture Notes*, 2013.

Cotterell, R., Mielke, S. J., Eisner, J., and Roark, B. Are all languages equally hard to language-model? *arXiv preprint arXiv:1806.03743*, 2018.

Delétang, G., Ruoss, A., Grau-Moya, J., Genewein, T., Wenliang, L. K., Catt, E., Cundy, C., Hutter, M., Legg, S., Veness, J., et al. Neural networks and the chomsky hierarchy. *arXiv preprint arXiv:2207.02098*, 2022.

Dong, Y., Jiang, X., Liu, H., Jin, Z., Gu, B., Yang, M., and Li, G. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. *arXiv preprint arXiv:2402.15938*, 2024.

Duan, S., Khona, M., Iyer, A., Schaeffer, R., and Fiete, I. R. Uncovering latent memories: Assessing data leakage and memorization patterns in large language models. In *ICML 2024 Workshop on LLMs and Cognition*, 2024.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Freeman, J., Rippe, C., Debenedetti, E., and Andriushchenko, M. Exploring memorization and copyright violation in frontier llms: A study of the new york times v. openai 2023 lawsuit. *arXiv preprint arXiv:2412.06370*, 2024.

Fu, Y.-F., Tu, Y.-C., Cheng, T.-L., Lin, C.-Y., Yang, Y.-T., Liu, H.-Y., Liao, K.-T., Juan, D.-C., and Lin, S.-D. Think or remember? detecting and directing llms towards memorization or generalization. *arXiv preprint arXiv:2412.18497*, 2024.

Hahn, M. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171, 2020.

Hahn, M. and Rofin, M. Why are sensitive functions hard for transformers? *arXiv preprint arXiv:2402.09963*, 2024.

Haviv, A., Cohen, I., Gidron, J., Schuster, R., Goldberg, Y., and Geva, M. Understanding transformer memorization recall through idioms. *arXiv preprint arXiv:2210.03588*, 2022.

Henderson, P., Li, X., Jurafsky, D., Hashimoto, T., Lemley, M. A., and Liang, P. Foundation models and fair use. *Journal of Machine Learning Research*, 24(400):1–79, 2023.

Hopkins, M. Towards more natural artificial languages. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pp. 85–94, 2022.

Huang, J., Shao, H., and Chang, K. C.-C. Are large pre-trained language models leaking your personal information? *arXiv preprint arXiv:2205.12628*, 2022.

Ippolito, D., Tramèr, F., Nasr, M., Zhang, C., Jagielski, M., Lee, K., Choquette-Choo, C. A., and Carlini, N. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*, 2022.

Jagielski, M., Thakkar, O., Tramer, F., Ippolito, D., Lee, K., Carlini, N., Wallace, E., Song, S., Thakurta, A., Papernot, N., et al. Measuring forgetting of memorized training examples. *arXiv preprint arXiv:2207.00099*, 2022.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Jumelet, J. and Zuidema, W. Transparency at the source: Evaluating and interpreting language models with access to the true distribution. *arXiv preprint arXiv:2310.14840*, 2023.

Kandpal, N., Wallace, E., and Raffel, C. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pp. 10697–10707. PMLR, 2022.

Kim, S., Yun, S., Lee, H., Gubri, M., Yoon, S., and Oh, S. J. Propile: Probing privacy leakage in large language models. *Advances in Neural Information Processing Systems*, 36:20750–20762, 2023.

Liu, B., Ash, J. T., Goel, S., Krishnamurthy, A., and Zhang, C. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.

Mao, A., Mohri, M., and Zhong, Y. Cross-entropy loss functions: Theoretical analysis and applications. In *International conference on Machine learning*, pp. 23803–23828. PMLR, 2023.

McCoy, R. T., Smolensky, P., Linzen, T., Gao, J., and Celikyilmaz, A. How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven. *Transactions of the Association for Computational Linguistics*, 11:652–670, 2023.

Merrill, W. Formal languages and the nlp black box. In *International Conference on Developments in Language Theory*, pp. 1–8. Springer, 2023.

Mielke, S. J., Cotterell, R., Gorman, K., Roark, B., and Eisner, J. What kind of language is hard to language-model? *arXiv preprint arXiv:1906.04726*, 2019.

Mireshghallah, F., Uniyal, A., Wang, T., Evans, D. K., and Berg-Kirkpatrick, T. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 1816–1826, 2022.

Mueller, F. B., Görge, R., Bernzen, A. K., Pirk, J. C., and Poretschkin, M. Llms and memorization: On quality and specificity of copyright compliance. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pp. 984–996, 2024.

Murty, S., Sharma, P., Andreas, J., and Manning, C. D. Characterizing intrinsic compositionality in transformers with tree projections. *arXiv preprint arXiv:2211.01288*, 2022.

Papadimitriou, I. and Jurafsky, D. Injecting structural hints: Using language models to study inductive biases in language learning. *arXiv preprint arXiv:2304.13060*, 2023.

Pappu, A., Porter, B., Shumailov, I., and Hayes, J. Measuring memorization in rlhf for code completion. *arXiv preprint arXiv:2406.11715*, 2024.

Peng, Z., Wang, Z., and Deng, D. Near-duplicate sequence search at scale for large language model memorization evaluation. *Proceedings of the ACM on Management of Data*, 1(2):1–18, 2023.

Satvaty, A., Verberne, S., and Turkmen, F. Undesirable memorization in large language models: A survey. *arXiv preprint arXiv:2410.02650*, 2024.

Schwarzschild, A., Feng, Z., Maini, P., Lipton, Z. C., and Kolter, J. Z. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*, 2024.

Shi, H., Gao, S., Tian, Y., Chen, X., and Zhao, J. Learning bounded context-free-grammar via lstm and the transformer: difference and the explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8267–8276, 2022.

Speicher, T., Ghosh, B., Khan, M. A., Wu, Q., Nanda, V., Das, S., Gummadi, K. P., and Terzi, E. Rethinking memorization in llms: On learning by rote vs. with understanding.

Strobl, L., Merrill, W., Weiss, G., Chiang, D., and Angluin, D. Transformers as recognizers of formal languages: A survey on expressivity. *arXiv preprint arXiv:2311.00208*, 2023.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., Tafti, P., Hussenot, L., Sessa, P. G., Chowdhery, A., Roberts, A., Barua, A., Botev, A., Castro-Ros, A., Slone, A., Héliou, A., Tacchetti, A., Bulanova, A., Paterson, A., Tsai, B., Shahriari, B., Lan, C. L., Choquette-Choo, C. A., Crepy, C., Cer, D., Ippolito, D., Reid, D., Buchatskaya, E., Ni, E., Noland, E., Yan, G., Tucker, G., Muraru, G.-C., Rozhdestvenskiy, G., Michalewski, H., Tenney, I., Grishchenko, I., Austin, J., Keeling, J., Labanowski, J., Lespiau, J.-B., Stanway, J., Brennan, J., Chen, J., Ferret, J., Chiu, J., Mao-Jones, J., Lee, K., Yu, K., Millican, K., Sjoesund, L. L., Lee, L., Dixon, L., Reid, M., Mikuła, M., Wirth, M., Sharman, M., Chinaev, N., Thain, N., Bachem, O., Chang, O., Wahltinez, O., Bailey, P., Michel, P., Yotov, P., Chaabouni, R., Comanescu, R., Jana, R., Anil, R., McIlroy, R., Liu, R., Mullins, R., Smith, S. L., Borgeaud, S., Girgin, S., Douglas, S., Pandya, S., Shakeri, S., De, S., Klimenko, T., Hennigan, T., Feinberg, V., Stokowiec, W., hui Chen, Y., Ahmed, Z., Gong, Z., Warkentin, T., Peran, L., Giang, M., Farabet, C., Vinyals, O., Dean, J., Kavukcuoglu, K., Hassabis, D., Ghahramani, Z., Eck, D., Barral, J., Pereira, F., Collins, E., Joulin, A., Fiedel, N., Senter, E., Andreev, A., and Kenealy, K. Gemma: Open models based on gemini research and technology, 2024. URL https://arxiv.org/abs/2403.08295.

Tirumala, K., Markosyan, A., Zettlemoyer, L., and Aghajanyan, A. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35: 38274–38290, 2022.

van den Burg, G. and Williams, C. On memorization in probabilistic deep generative models. *Advances in Neural Information Processing Systems*, 34:27916–27928, 2021.

Wang, X., Antoniades, A., Elazar, Y., Amayuelas, A., Albalak, A., Zhang, K., and Wang, W. Y. Generalization vs memorization: Tracing language models' capabilities back to pretraining data. *arXiv preprint arXiv:2407.14985*, 2024.

White, J. C. and Cotterell, R. Examining the inductive bias of neural language models with artificial languages. *arXiv preprint arXiv:2106.01044*, 2021.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhang, C., Ippolito, D., Lee, K., Jagielski, M., Tramèr, F., and Carlini, N. Counterfactual memorization in neural language models. *ArXiv*, abs/2112.12938, 2021. URL https://api.semanticscholar.org/CorpusID:245502053.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zhou, Z., Xiang, J., Chen, C., and Su, S. Quantifying and analyzing entity-level memorization in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19741–19749, 2024.

Table 1: Characteristics of memorization measures.

Memorization Measure	Motivation	Memorization Threshold	Ease of Operationalization	Strictness of Measure
Recollection	Disclosing private information	Manual	**Easy**	Variable
Counterfactual	Differential privacy	**Adaptive**	Hard	Medium
Contextual (ours)	Local over-fitting	**Adaptive**	Hard	**High**

A. Extended Related Work

Memorization in LLMs is an active area of research, from the perspective of privacy and security risks (Carlini et al., 2021; Huang et al., 2022; Kim et al., 2023; Jagielski et al., 2022), unintended form of learning due to local over-fitting (van den Burg & Williams, 2021), copyright concerns related to verbatim reproduction (Bender et al., 2021; Henderson et al., 2023; Mueller et al., 2024; Freeman et al., 2024), etc. As a natural next step, multiple measures of memorization are proposed to detect and quantify memorization. Among these measures, majority belong to the category of recollection-based memorization (Schwarzschild et al., 2024; Biderman et al., 2024), such as prefect memorization (Kandpal et al., 2022), verbatim or exact memorization (Carlini et al., 2021; 2019; Tirumala et al., 2022; Mireshghallah et al., 2022), approximate memorization (Ippolito et al., 2022; Peng et al., 2023; Duan et al., 2024), entity memorization (Zhou et al., 2024), etc. For an extended taxonomy of memorization measures, we refer to a recent survey paper (Satvaty et al., 2024). Regardless of how these measures are operationalized, a common trait is that the recollection ability of an LLM given a training string dictates its extent of memorization. For example, Tirumala et al. (2022) consider training accuracy as the proxy of memorization: given a training string as a prompt, an LLM memorizes it if it recollects the next token in the string correctly. Carlini et al. (2022) propose a relatively stringent measure by imposing an exact recollection of next 50 tokens. Therefore, a critical design choice an experimenter makes is to set the threshold on recollection to declare a string as memorized – the choice has consequences on the interpretation of memorization, as we study in this paper.

In a related line of work, Zhang et al. (2021) define counterfactual memorization as the change in a model's generative performance when a string is included in training versus excluded (Pappu et al., 2024; Feldman & Zhang, 2020). This approach specifically highlights rare strings, which tend to cause larger performance shifts and are often missed by recollection-based memorization measures. By introducing contextual memorization, we argue that all strings – regardless of frequency – can be recollected to some extent based on their context (Haviv et al., 2022; Wang et al., 2024; Fu et al., 2024; Chen et al., 2025; Speicher et al.; Dong et al., 2024; McCoy et al., 2023). We define memorization as occurring only when a string's training-time recollection exceeds its optimal contextual recollection threshold, making contextual memorization a stricter criterion than counterfactual memorization

Despite the abundance of memorization measures, their potentially conflicting implications remain underexplored – we aim to address this research gap.

B. Additional Discussion on Memorization Measures

Lemma 3.1. Contextual memorization is stricter than counterfactual memorization. Contextual memorization of a string starts at the same or in a later epoch in training than counterfactual memorization, and the contextual memorization score is a lower bound of the counterfactual memorization score.

Proof. We prove by considering loss as the metric of recollection. We assume that at any epoch, the training loss of a string is not higher than the counterfactual test loss of the same string when excluding the string from training, which is a feasible assumption in practice.

For a string s, let the optimal contextual loss be $\min_{e^*} \texttt{loss}(M_{e^*}(D'), s)$, which is the lowest counterfactual test loss in all epochs.

Contextual memorization starts at an epoch $e_s^{\texttt{ctx}}$ when $\texttt{loss}(M_{e_s^{\texttt{ctx}}}(D), s) < \min_{e^*} \texttt{loss}(M_{e^*}(D'), s)$, i.e., the training loss of s is lower than the optimal contextual loss of the string. For an epoch $e < e_s^{\texttt{ctx}}$ earlier than the start of contextual memorization, $\texttt{loss}(M_e(D), s) \geq \min_{e^*} \texttt{loss}(M_{e^*}(D'), s)$.

Counterfactual memorization starts at an epoch $e_s^{\texttt{cf}}$ when $\texttt{loss}(M_{e_s^{\texttt{cf}}}(D), s) < \texttt{loss}(M_{e_s^{\texttt{cf}}}(D'), s)$, i.e., the training loss of

s is lower than the counterfactual test loss at the same epoch. For an epoch $e < e_s^{\texttt{cf}}$ earlier than the start of counterfactual memorization, training loss of s is equal to the counterfactual test loss, $\texttt{loss}(M_e(D), s) = \texttt{loss}(M_e(D'), s)$. Because, $\texttt{loss}(M_e(D), s) \leq \texttt{loss}(M_e(D'), s)$ for any training epoch e', according to our assumption.

Let contextual memorization start earlier than counterfactual memorization, i.e., $e_s^{\texttt{ctx}} = e_s^{\texttt{cf}} - 1$.

$$\texttt{loss}(M_{e_s^{\texttt{cf}}-1}(D), s) < \min_{e^*} \texttt{loss}(M_{e^*}(D'), s)$$
$$\text{Since, } \min_{e^*} \texttt{loss}(M_{e^*}(D'), s) \leq \texttt{loss}(M_{e_s^{\texttt{cf}}-1}(D'), s)$$
$$\Rightarrow \texttt{loss}(M_{e'-1}, s) < \texttt{loss}(M_{e_s^{\texttt{cf}}-1}(D'), s)$$

But $\texttt{loss}(M_{e_s^{\texttt{cf}}-1}(D), s) = \texttt{loss}(M_{e_s^{\texttt{cf}}-1}(D'), s)$, which is a contradiction. Therefore, contextual memorization cannot start earlier than counterfactual memorization.

On the other hand, contextual memorization can start later or in the same epoch as counterfactual memorization, since for an epoch $e \geq e_s^{\texttt{cf}}$,

$$\underbrace{\texttt{loss}(M_e(D), s) \geq \min_{e^*} \texttt{loss}(M_{e^*}(D'), s)}_{\text{contextual memorization does not start}} \text{ and } \underbrace{\texttt{loss}(M_e(D), s) < \texttt{loss}(M_e(D'), s)}_{\text{counterfactual memorization starts}}$$

Furthermore, the counterfactual memorization score is no less than the contextual memorization score, since at any epoch $e \geq \max(e_s^{\texttt{cf}}, e_s^{\texttt{ctx}})$, i.e., after both memorization starts, $\min_{e^*} \texttt{loss}(M_{e^*}(D'), s) \leq \texttt{loss}(M_e(D'), s)$.

$$\underbrace{\frac{\min_{e^*} \texttt{loss}(M_{e^*}(D'), s) - \texttt{loss}(M_e(D), s)}{\min_{e^*} \texttt{loss}(M_{e^*}(D'), s)}}_{\text{contextual memorization score}} \leq \underbrace{\frac{\texttt{loss}(M_e(D'), s) - \texttt{loss}(M_e(D), s)}{\texttt{loss}(M_e(D'), s)}}_{\text{counterfactual memorization score}}$$

Therefore, counterfactual memorization is likely to overestimate memorization than contextual memorization, while reporting memorization at an earlier epoch than contextual memorization.

□

Efficient Computation of Counterfactual and Contextual Memorization. Both measures require retraining to compute counterfactual loss, followed by optimal contextual loss. We propose an approximation that avoids retraining. If the occurrence frequency of both training and test strings are known in a training dynamic, which is the case of a formal language, we find a test string as similarly occurring to the training string, and use its test loss as counterfactual loss and the lowest test loss as the optimal contextual loss. The hypothesis is that *similarly occurring strings in a language tend to yield similar losses from the LLM*.

C. Experimental Setup

We experiment with 18 open-source LLMs from 6 families, such as Mistral (Jiang et al., 2023), Llama (Dubey et al., 2024), Qwen (Yang et al., 2024), Gemma (Team et al., 2024), Pythia (Biderman et al., 2023), and Opt (Zhang et al., 2022), ranging from 0.5B to 13B parameters. All reported results are averaged over three experimental runs.

Each training (specifically, fine-tuning) is performed for 50 epochs with a batch size of 8 and a linear learning rate scheduler with a warm-up ratio of 0.05. We fix the learning rate for Qwen, Gemma, and Llama-3 families as 5×10^{-5}, Mistral, Opt, and Llama-2 families as 5×10^{-6}, and Pythia family as 10^{-5}. We consider training dataset sizes $\{16, 64, 256, 1024\}$ and evaluate on 1024 test strings. In each training, we find the epoch of best learning according to lowest cross-entropy loss on the test strings and report respective weighted memorization by different measures.

While several prior studies have adopted formal languages in LLMs, their goals differed from ours (Borenstein et al., 2024; Akyürek et al., 2024; Jumelet & Zuidema, 2023; Papadimitriou & Jurafsky, 2023; White & Cotterell, 2021; Hopkins, 2022; Allen-Zhu & Li, 2023; Chi et al., 2023; Murty et al., 2022; Liu et al., 2022; Shi et al., 2022; Bhattamishra et al., 2020;

$S \to A16$ [1]
$A16 \to A15\ A14\ A13$ [0.50]
$A16 \to A13\ A15\ A14$ [0.50]
$A13 \to A11\ A12$ [0.50]
$A13 \to A12\ A11$ [0.50]
$A14 \to A11\ A10\ A12$ [0.50]
$A14 \to A10\ A11\ A12$ [0.50]
$A15 \to A12\ A11\ A10$ [0.50]
$A15 \to A11\ A12\ A10$ [0.50]
$A10 \to A7\ A9\ A8$ [0.50]
$A10 \to A9\ A8\ A7$ [0.50]
$A11 \to A8\ A7\ A9$ [0.50]
$A11 \to A7\ A8\ A9$ [0.50]
$A12 \to A8\ A9\ A7$ [0.50]
$A12 \to A9\ A7\ A8$ [0.50]
$A7 \to 3\ 1\ 2$ [0.50]
$A7 \to 1\ 2\ 3$ [0.50]
$A8 \to 6\ 5\ 4$ [0.50]
$A8 \to 6\ 4\ 5$ [0.50]
$A9 \to 9\ 8\ 7$ [0.50]
$A9 \to 8\ 7\ 9$ [0.50]

$S \to A16$ [1]
$A16 \to A15\ A14\ A13$ [0.95]
$A16 \to A13\ A15\ A14$ [0.05]
$A13 \to A11\ A12$ [0.95]
$A13 \to A12\ A11$ [0.05]
$A14 \to A11\ A10\ A12$ [0.95]
$A14 \to A10\ A11\ A12$ [0.05]
$A15 \to A12\ A11\ A10$ [0.95]
$A15 \to A11\ A12\ A10$ [0.05]
$A10 \to A7\ A9\ A8$ [0.95]
$A10 \to A9\ A8\ A7$ [0.05]
$A11 \to A8\ A7\ A9$ [0.95]
$A11 \to A7\ A8\ A9$ [0.05]
$A12 \to A8\ A9\ A7$ [0.95]
$A12 \to A9\ A7\ A8$ [0.05]
$A7 \to 3\ 1\ 2$ [0.95]
$A7 \to 1\ 2\ 3$ [0.05]
$A8 \to 6\ 5\ 4$ [0.95]
$A8 \to 6\ 4\ 5$ [0.05]
$A9 \to 9\ 8\ 7$ [0.95]
$A9 \to 8\ 7\ 9$ [0.05]

Figure 3: Production rules of G_1 (left) and G_2 (right). Compared to G_1, the grammar G_2 generates more skewed distribution (or lower entropy) strings, since one out of two production rules for each non-terminal is selected with higher probability.

Merrill, 2023; Strobl et al., 2023; Hahn, 2020; Delétang et al., 2022; Hahn & Rofin, 2024; Cotterell et al., 2018; Mielke et al., 2019). Below, we provide details of the formal languages used in our experiments, along with their formal definitions. Intuitively, we carefully design languages to show the robustness of our results across changing the entropy of the langauge and token types of the language.

Formal Languages and Grammars Throughout our experiments, we provide the LLM strings sampled from a probabilistic formal language. Underneath, a probabilistic formal language is represented by a *probabilistic formal grammars*, or simply *grammars* (Collins, 2013). Specifically, a grammar consists of two sets of symbols called the *non-terminals* and *terminals*, a set of rules to rewrite strings over these symbols that contain at least one nonterminal – also called the *production rules*, and a probability distribution over the production rules. Formally, a probabilistic formal grammar, is defined as a quintuple.

$$G = (\mathbf{N}, \mathbf{T}, \mathbf{R}, S, \mathbf{P})$$

where $\mathbf{N}$ is the set of non-terminals, $\mathbf{T}$ is the set of terminals (equivalently, tokens), $\mathbf{R}$ is the set of production rules, $S \in \mathbf{N}$ is the start non-terminal, and $\mathbf{P}$ is the set of probabilities on production rules.

Formal languages are divided into well-known classes based on the *complexity* of the language membership problem, i.e., the *complexity* of the grammars needed to generate them (Chomsky, 1956). In this paper, we use one class of grammars, namely, hierarchical probabilistic context-free grammars (HPCFGs) (Allen-Zhu & Li, 2023). Specifically, our experiments are based on teaching LLMs languages represented by HPCFGs. We use HPCFGs because they are simple syntactically and can represent languages that are structurally similar to natural languages (Allen-Zhu & Li, 2023; Shi et al., 2022).

$S \to S5$ [1]
$S5 \to B4\ C1_1\ E4\ T1_1$ [0.25]
$S5 \to B4\ C1_2\ E4\ T1_2$ [0.25]
$S5 \to B4\ C1_3\ E4\ T1_3$ [0.25]
$S5 \to B4\ C1_4\ E4\ T1_4$ [0.25]
$B4 \to B3$ [0.3333]
$B4 \to B3\ B3\ B3$ [0.3333]
$B4 \to B3\ B3$ [0.3333]
$B3 \to B2$ [0.3333]
$B3 \to B2$ [0.3333]
$B3 \to B2\ B2$ [0.3333]
$B2 \to B1$ [0.3333]
$B2 \to B1$ [0.3333]
$B2 \to B1\ B1\ B1$ [0.3333]
$B1 \to 2\ 9\ 3$ [0.3333]
$B1 \to 9\ 6\ 1$ [0.3333]
$B1 \to 1\ 8\ 6$ [0.3333]
$E4 \to E3$ [0.3333]
$E4 \to E3\ E3$ [0.3333]
$E4 \to E3\ E3\ E3$ [0.3333]
$E3 \to E2$ [0.3333]
$E3 \to E2\ E2$ [0.3333]
$E3 \to E2$ [0.3333]
$E2 \to E1\ E1$ [0.3333]
$E2 \to E1$ [0.3333]
$E2 \to E1\ E1\ E1$ [0.3333]
$E1 \to 5\ 6\ 5\ 9$ [0.3333]
$E1 \to 1\ 8\ 6\ 6$ [0.3333]
$E1 \to 1\ 5\ 1\ 5$ [0.3333]
$T1_1 \to 1$ [1]
$T1_2 \to 2$ [1]
$T1_3 \to 3$ [1]
$T1_4 \to 4$ [1]
$C1_1 \to 5$ [1]
$C1_2 \to 6$ [1]
$C1_3 \to 7$ [1]
$C1_4 \to 8$ [1]
$C1_5 \to 9$ [1]

$S \to S5$ [1]
$S5 \to B4\ C1_1\ E4\ T1_1$ [0.25]
$S5 \to B4\ C1_2\ E4\ T1_2$ [0.25]
$S5 \to B4\ C1_3\ E4\ T1_3$ [0.25]
$S5 \to B4\ C1_4\ E4\ T1_4$ [0.25]
$B4 \to B3$ [0.3333]
$B4 \to B3\ B3\ B3$ [0.3333]
$B4 \to B3\ B3$ [0.3333]
$B3 \to B2$ [0.3333]
$B3 \to B2$ [0.3333]
$B3 \to B2\ B2$ [0.3333]
$B2 \to B1$ [0.3333]
$B2 \to B1$ [0.3333]
$B2 \to B1\ B1\ B1$ [0.3333]
$B1 \to 2\ 9\ 3$ [0.95]
$B1 \to 9\ 6\ 1$ [0.025]
$B1 \to 1\ 8\ 6$ [0.025]
$E4 \to E3$ [0.3333]
$E4 \to E3\ E3$ [0.3333]
$E4 \to E3\ E3\ E3$ [0.3333]
$E3 \to E2$ [0.3333]
$E3 \to E2\ E2$ [0.3333]
$E3 \to E2$ [0.3333]
$E2 \to E1\ E1$ [0.3333]
$E2 \to E1$ [0.3333]
$E2 \to E1\ E1\ E1$ [0.3333]
$E1 \to 5\ 6\ 5\ 9$ [0.95]
$E1 \to 1\ 8\ 6\ 6$ [0.025]
$E1 \to 1\ 5\ 1\ 5$ [0.025]
$T1_1 \to 1$ [1]
$T1_2 \to 2$ [1]
$T1_3 \to 3$ [1]
$T1_4 \to 4$ [1]
$C1_1 \to 5$ [1]
$C1_2 \to 6$ [1]
$C1_3 \to 7$ [1]
$C1_4 \to 8$ [1]
$C1_5 \to 9$ [1]

Figure 4: Production rules of G_3 (left) and G_4 (right). Compared to G_3, the grammar G_4 generates more skewed distribution (or lower entropy) of strings, since one out of three production rules of non-terminal $B1$ and $E1$ is selected with higher probability.

$S \to A16$ [1]
$A16 \to A15\ A13$ [0.50]
$A16 \to A13\ A15\ A14$ [0.50]
$A13 \to A11\ A12$ [0.50]
$A13 \to A12\ A11$ [0.50]
$A14 \to A11\ A10\ A12$ [0.50]
$A14 \to A10\ A11\ A12$ [0.50]
$A15 \to A12\ A11\ A10$ [0.50]
$A15 \to A11\ A12\ A10$ [0.50]
$A10 \to A7\ A9\ A8$ [0.50]
$A10 \to A9\ A8\ A7$ [0.50]
$A11 \to A8\ A7\ A9$ [0.50]
$A11 \to A7\ A8\ A9$ [0.50]
$A12 \to A8\ A9\ A7$ [0.50]
$A12 \to A9\ A7\ A8$ [0.50]
$A7 \to 3\ 1$ [0.50]
$A7 \to 1\ 2\ 3$ [0.50]
$A8 \to 6\ 5$ [0.50]
$A8 \to 6\ 4\ 5$ [0.50]
$A9 \to 9\ 8\ 7$ [0.50]
$A9 \to 8\ 7$ [0.50]

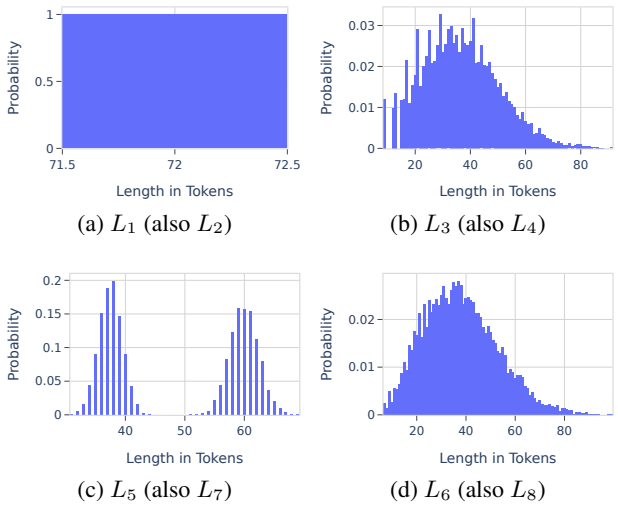
$S \to S5$ [1]
$S5 \to B4\ C1_1\ E4\ T1_1$ [0.25]
$S5 \to B4\ C1_2\ E4\ T1_2$ [0.25]
$S5 \to B4\ C1_3\ E4\ T1_3$ [0.25]
$S5 \to B4\ C1_4\ E4\ T1_4$ [0.25]
$B4 \to B3$ [0.3333]
$B4 \to B3\ B3\ B3$ [0.3333]
$B4 \to B3\ B3$ [0.3333]
$B3 \to B2$ [0.3333]
$B3 \to B2$ [0.3333]
$B3 \to B2\ B2$ [0.3333]
$B2 \to B1$ [0.3333]
$B2 \to B1$ [0.3333]
$B2 \to B1\ B1\ B1$ [0.3333]
$B1 \to 2\ 9\ 3$ [0.3333]
$B1 \to 9\ 6\ 1$ [0.3333]
$B1 \to 1\ 8\ 6\ 2$ [0.3333]
$E4 \to E3$ [0.3333]
$E4 \to E3\ E3$ [0.3333]
$E4 \to E3\ E3\ E3$ [0.3333]
$E3 \to E2$ [0.3333]
$E3 \to E2\ E2$ [0.3333]
$E3 \to E2$ [0.3333]
$E2 \to E1\ E1$ [0.3333]
$E2 \to E1$ [0.3333]
$E2 \to E1\ E1\ E1$ [0.3333]
$E1 \to 5\ 6$ [0.3333]
$E1 \to 1\ 8\ 6\ 6$ [0.3333]
$E1 \to 1\ 5\ 1\ 5\ 5\ 9$ [0.3333]
$T1_1 \to 1$ [1]
$T1_2 \to 2$ [1]
$T1_3 \to 3$ [1]
$T1_4 \to 4$ [1]
$C1_1 \to 5$ [1]
$C1_2 \to 6$ [1]
$C1_3 \to 7$ [1]
$C1_4 \to 8$ [1]
$C1_5 \to 9$ [1]

Figure 5: Production rules of G_5 (left) and G_6 (right). These grammars are adapted from G_1 and G_3 respectively, by allowing non-uniform lengths of tokens in the lowest level production rules.

$S \to A16$ [1]
$A16 \to A15\ A13$ [0.50]
$A16 \to A13\ A15\ A14$ [0.50]
$A13 \to A11\ A12$ [0.50]
$A13 \to A12\ A11$ [0.50]
$A14 \to A11\ A10\ A12$ [0.50]
$A14 \to A10\ A11\ A12$ [0.50]
$A15 \to A12\ A11\ A10$ [0.50]
$A15 \to A11\ A12\ A10$ [0.50]
$A10 \to A7\ A9\ A8$ [0.50]
$A10 \to A9\ A8\ A7$ [0.50]
$A11 \to A8\ A7\ A9$ [0.50]
$A11 \to A7\ A8\ A9$ [0.50]
$A12 \to A8\ A9\ A7$ [0.50]
$A12 \to A9\ A7\ A8$ [0.50]
$A7 \to c\ a$ [0.50]
$A7 \to a\ b\ c$ [0.50]
$A8 \to f\ e$ [0.50]
$A8 \to f\ d\ e$ [0.50]
$A9 \to i\ h\ g$ [0.50]
$A9 \to h\ g$ [0.50]

$S \to S5$ [1]
$S5 \to B4\ C1_1\ E4\ T1_1$ [0.25]
$S5 \to B4\ C1_2\ E4\ T1_2$ [0.25]
$S5 \to B4\ C1_3\ E4\ T1_3$ [0.25]
$S5 \to B4\ C1_4\ E4\ T1_4$ [0.25]
$B4 \to B3$ [0.3333]
$B4 \to B3\ B3\ B3$ [0.3333]
$B4 \to B3\ B3$ [0.3333]
$B3 \to B2$ [0.3333]
$B3 \to B2$ [0.3333]
$B3 \to B2\ B2$ [0.3333]
$B2 \to B1$ [0.3333]
$B2 \to B1$ [0.3333]
$B2 \to B1\ B1\ B1$ [0.3333]
$B1 \to b\ i\ c$ [0.3333]
$B1 \to i\ f\ a$ [0.3333]
$B1 \to a\ h\ f\ b$ [0.3333]
$E4 \to E3$ [0.3333]
$E4 \to E3\ E3$ [0.3333]
$E4 \to E3\ E3\ E3$ [0.3333]
$E3 \to E2$ [0.3333]
$E3 \to E2\ E2$ [0.3333]
$E3 \to E2$ [0.3333]
$E2 \to E1\ E1$ [0.3333]
$E2 \to E1$ [0.3333]
$E2 \to E1\ E1\ E1$ [0.3333]
$E1 \to e\ f$ [0.3333]
$E1 \to a\ h\ f\ f$ [0.3333]
$E1 \to a\ e\ a\ e\ e\ i$ [0.3333]
$T1_1 \to a$ [1]
$T1_2 \to b$ [1]
$T1_3 \to c$ [1]
$T1_4 \to d$ [1]
$C1_1 \to e$ [1]
$C1_2 \to f$ [1]
$C1_3 \to g$ [1]
$C1_4 \to h$ [1]
$C1_5 \to i$ [1]

Figure 6: Production rules of G_7 (left) and G_8 (right). These grammars are adapted from G_5 and G_6 respectively, by replacing numerical tokens with Latin character tokens.



(a) L_1 (also L_2) (b) L_3 (also L_4)

(c) L_5 (also L_7) (d) L_6 (also L_8)

Figure 7: Length distribution of considered probabilistic languages, based on 10000 sampled strings per language.

Description of Grammars and Identified Languages. In our experiments, we consider two generic structure for the considered grammars, one adapted from (Allen-Zhu & Li, 2023), namely G_1, G_2, G_5, G_7, and another is proposed by us, namely G_3, G_4, G_6, G_8.

In the first generic structure, each grammar has $\mathbf{N} = \{S, A7, A8, \ldots, A16\}$ and $\mathbf{T} = \{1, 2, 3, \ldots, 9\}$. The grammar has four levels of hierarchy: the non-terminals from top to bottom levels are $\{A16\}$, $\{A13, A14, A15\}$, $\{A10, A11, A12\}$, and $\{A7, A8, A9\}$, followed by terminals $\{1, 2, 3, \ldots, 9\}$. Each non-terminal (except the start non-terminal) has two expansion rules, consisting of non-terminals from the immediate lower level. Further, the expansion rules are probabilistic, where the sum of probabilities of all expansion rules from a given non-terminal is 1.

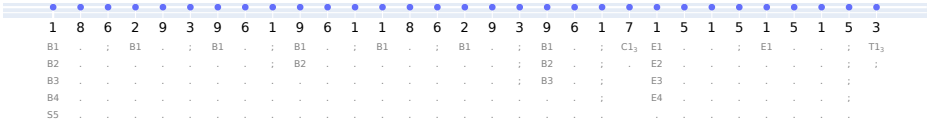
The second generic structure is inspired by bridging two HPCFGs together, starting from $B4$ and $E4$ at level 4. The two sub-grammars are connected by non-terminal $C1_i$; and $E4$ ends with $T1_j$. The goal is to generate strings containing long range dependencies: how the first sub-grammar expansion ends determines how the overall string ends by utilizing non-terminals $C1_i$ and $T1_j$.

In all cases, G_i produces a probabilistic context free language L_i. Figure 7 denotes the length distribution of different languages, and Figure 8 demonstrates how hierarchical non-terminals are applied in different positions in the representative strings.

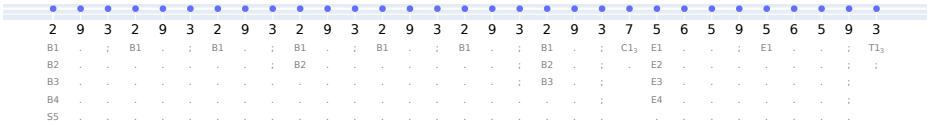
Sampling Strings from a Formal Language. Given a language L generated by a HPCFG, we first need to obtain *training* samples, i.e., set of i.i.d. samples of strings *in-language* L. To *sample a string from the language*, we start from a special string in the grammar containing a single, distinguished nonterminal called the "start" or "root" symbol, and apply the production rules to rewrite the string repeatedly. If several rules can be used to rewrite the string at any stage, we sample one such rule from the probability distribution over the rules and apply it. We stop when we obtain a string containing terminals only. This string is a sample drawn from the language. We can repeat this process to draw any number of i.i.d. samples from the language.

(a) L_1

(b) L_2

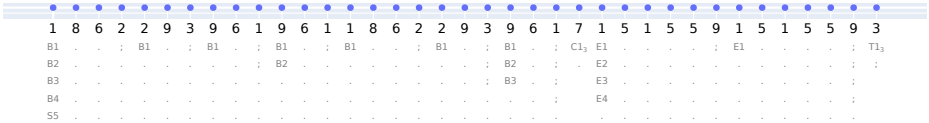


(c) L_3



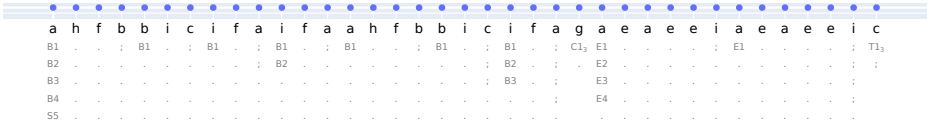
(d) L_4

(e) L_5



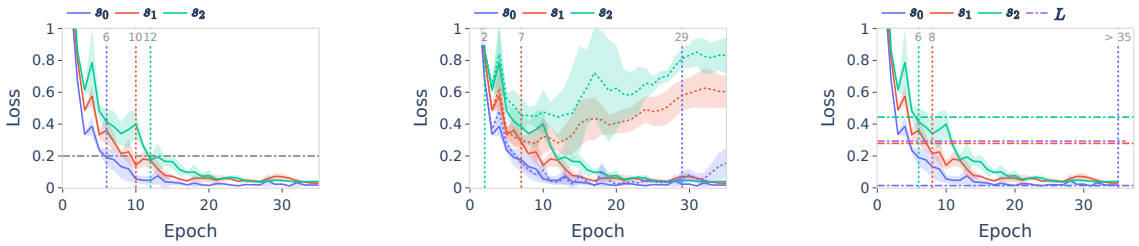
(f) L_6

(g) L_7



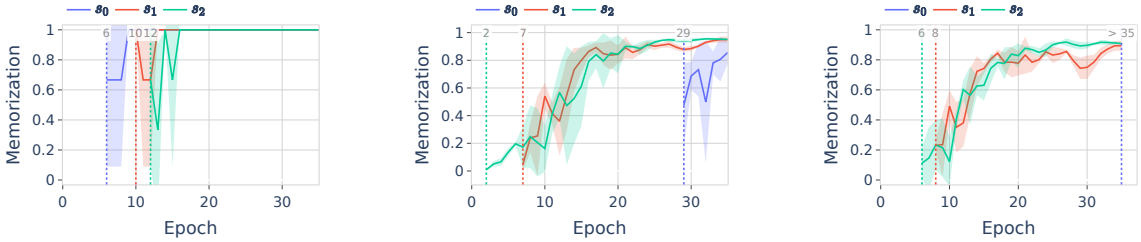
(h) L_8

Figure 8: Representative strings from different languages, annotated with non-terminals applied in different positions by the respective hierarchical grammar.



(a) Recollection (0.2) (b) Counterfactual (c) Contextual

Figure 9: Start of memorization of selected strings in Language L_2.

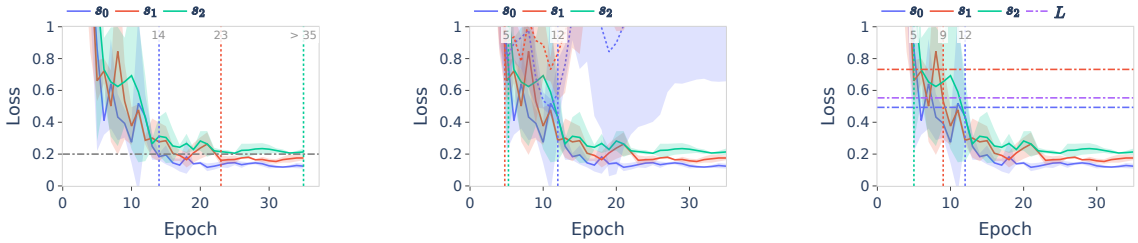


(a) Recollection (0.2) (b) Counterfactual (c) Contextual

Figure 10: Memorization score of strings in language L_2, respective to Figure 2. In different strings, memorization score usually increases with epochs, with contextual memorization providing a lower bound of counterfactual memorization.

D. Additional Experimental Results

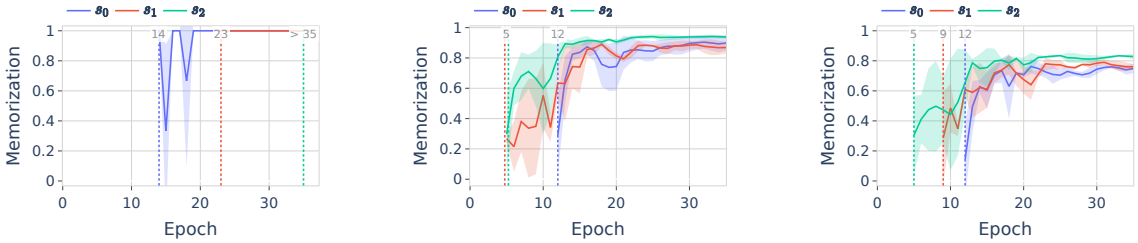
Memorization Scores of Individual Strings. In Figure 10, we demonstrate the memorization scores of strings, corresponding to Figure 2, across multiple memorization measures. In all measures, the memorization score usually increases with epochs, and there is no substantial difference among strings of varying frequency – different measures agree on the memorization score. Finally, as we theoretically demonstrate, contextual memorization score provides a lower bound of counterfactual memorization score.



(a) Recollection (0.2) (b) Counterfactual (c) Contextual

Figure 11: Start of memorization of selected strings in language L_4 (specifically, a modified version of L_4 as explained below). The observation is consistent with language L_2, as shown in figure 2, where frequency of strings correlates with the start of recollection-based memorization. Similarly, frequency often inversely correlates with counterfactual and contextual memorization, with an exception that both s_1 and s_2 are memorized at the same epoch in the counterfactual memorization. Thus, regardless of whether correlation or inverse correlation exists *strongly* between string frequency and the order of memorization, a more consistent observation is that memorization measures disagree with each other when applied to the same training dynamic on identical strings.

In this experiment, to better differentiate the strings s_0, s_1, s_2 based on frequency, we modify L_4 to be even more skewed. We apply high probability to one random production rule in each non-terminal in all levels, beyond the lowest level non-terminals in L_4, as shown in Figure 4.



(a) Recollection (0.2) (b) Counterfactual (c) Contextual

Figure 12: Memorization score of strings in language L_4.

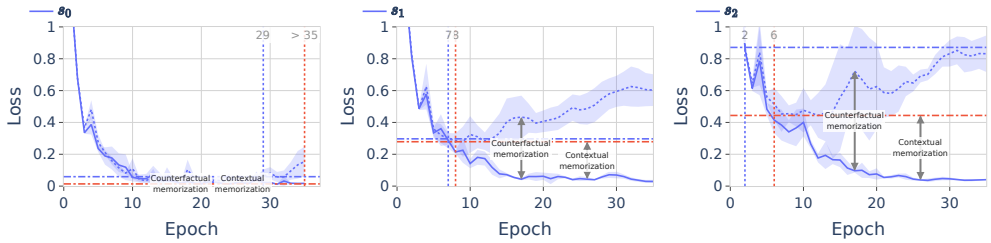


Figure 13: Contextual memorization is a stricter measure than counterfactual memorization. Red horizontal dash-dot line is the optimal contextual loss. Contextual memorization starts at the same or in a later epoch (red vertical dot line) than the start of counterfactual memorization (blue vertical dot line). The contextual memorization score (gray arrow) is a lower bound of counterfactual memorization score, intuitively by comparing the arrow-length.