Counterfactual Influence as a Distributional Quantity

Matthieu Meeus [1] **Igor Shilov** [1] **Georgios Kaissis** [2] **Yves-Alexandre de Montjoye** [1]

Abstract

Machine learning models are known to memorize samples from their training data, raising concerns around privacy and generalization. Counterfactual self-influence is a popular metric to study memorization, quantifying how the model's prediction for a sample changes depending on the sample's inclusion in the training dataset. However, recent work has shown memorization to be affected by factors beyond self-influence, with other training samples, in particular (near-)duplicates, having a large impact. We here study memorization treating counterfactual influence as a distributional quantity, taking into account how *all* training samples influence how a sample is memorized. For a small language model, we compute the full influence distribution of training samples on each other and analyze its properties. We find that solely looking at self-influence can severely underestimate tangible risks associated with memorization: the presence of (near-)duplicates seriously reduces self-influence, while we find these samples to be (near-)extractable. We observe similar patterns for image classification, where simply looking at the influence distributions reveals the presence of near-duplicates in CIFAR-10. Our findings highlight that memorization stems from complex interactions across training data and is better captured by the full influence distribution than by self-influence alone.

1. Introduction

Studying how Large Language Models (LLMs) learn from, and memorize, their training data is crucial to understand how they retain factual knowledge (Kandpal et al., 2023; Petroni et al., 2019) and generalize (Wang et al., 2024), but also to assess privacy risks (Carlini et al., 2021; Nasr et al., 2023), concerns related to copyright (Meeus et al., 2024b; Karamolegkou et al., 2023), and benchmark contamination (Oren et al., 2024; Mirzadeh et al., 2025). To quantify how a model memorizes a sample, Zhang et al. (2023) introduce *counterfactual memorization* (self-influence), computing how much a model's prediction for a sample changes when that sample is excluded from training. Recent studies (Shilov et al., 2024; Liu et al., 2025), however, suggest that LLMs memorize substantially across near-duplicates, widespread in LLM training data. This raises the question whether memorization of a piece of text can be viewed in isolation, i.e. by just looking at the self-influence.

Contribution. We investigate memorization by considering how the *entire* training dataset impacts the model's predictions on the target sample, rather than relying solely on self-influence. We adopt the framework from Zhang et al. (2023) to compute *counterfactual influence*, which measures how any training sample x_i impacts the prediction for a target x_t. Rather than focusing only on self-influence (where $x_i = x_t$), we compute the influence of all training samples on all targets and treat this as a distributional property.

We conduct our analysis on GPT-NEO 1.3B (Gao et al., 2020) models finetuned on the Natural Questions dataset (Kwiatkowski et al., 2019), computing the full influence matrix to quantify the contribution of each training point to each target prediction. We consider two types of records: (i) random, *unique records* from the data distribution, and (ii) random samples for which we include artificially crafted near-duplicates in the training data (*records with near-duplicates*). We find that unique records exhibit strong self-influence, whereas records with near-duplicates show more diffuse influence patterns; on average, their self-influence is 3 times lower and spread across similar records. Yet, these records are significantly more *extractable*; on average 5 times more, suggesting that self-influence alone underestimates this key risk associated with LLM memorization. We then find the ratio of the largest to the second largest influence, *Top-1 Influence Margin*, to clearly distinguish unique records from those with near-duplicates more effectively than self-influence. Finally, we study influence distributions for classification models trained on CIFAR-10 (Krizhevsky et al., 2009), finding that the distribution alone exposes near-duplicates.

[1]Imperial College London [2]Google DeepMind. Correspondence to: Yves-Alexandre de Montjoye <deMontjoye@imperial.ac.uk>.

Published at MemFM Workshop at ICML 2025, Vancouver, Canada.

Our results show that how models memorize their training data, especially in the presence of near-duplicates, might not be adequately captured by self-influence alone. Instead, memorization is multi-faceted phenomenon that can be better understood through the full distribution.

2. Background

Zhang et al. (2023) define counterfactual influence as the expected change in a model's loss on a target example x_t when a specific training point x_i is included versus excluded from the training data. This formulation represents an extension from label memorization in classification models as introduced by Feldman & Zhang (2020) to language models. Formally, let $\mathcal{D}$ denote the underlying data distribution, and let $D \sim \mathcal{D}$ be a dataset sampled i.i.d. from $\mathcal{D}$, the influence of x_i on target record x_t is:

$$\mathcal{I}(x_i \Rightarrow x_t) = \mathop{\mathbb{E}}_{A_j : x_i \notin D_j} \left[\mathcal{L}_{A_j}(x_t)\right] - \mathop{\mathbb{E}}_{A_j : x_i \in D_j} \left[\mathcal{L}_{A_j}(x_t)\right] \tag{1}$$

where A_j denotes a model trained on D_j, and $\mathcal{L}_{A_j}(x_t)$ is its loss on the target example x_t. Note that we adopt a sign change from the original definition, so that a positive value indicates that x_i contributes positively – reduces the loss – to the prediction of x_t. Zhang et al. (2023) primarily study the case when $x_i = x_t$, which they define as the counterfactual memorization or self-influence of x_t, i.e. $\mathcal{I}(x_t \Rightarrow x_t)$. While all influence values in this work correspond to counterfactual influence, we omit the qualifier and simply refer to them as (self-)influence throughout.

Here we examine memorization taking into account the entire influence distribution (i.e. $\mathcal{I}(x_i \Rightarrow x_t)$ for all x_i) rather than solely considering the value of self-influence. We do so for both unique records and, in particular, when the underlying data distribution contains *near-duplicates*. We say that x_t' is a *near-duplicate* of x_t if the distance between them, measured by a chosen function $d(\cdot, \cdot)$, satisfies $0 < d(x_t, x_t') < \epsilon$ for some threshold $\epsilon > 0$. Examining Equation 1, when the data distribution $\mathcal{D}$ contains many near-duplicates x_t' of x_t, these duplicates are likely to be included in the sets where $x_t \in D_j$ and where $x_t \notin D_j$. If these near-duplicates strongly influence model behavior on x_t, the marginal effect of including x_t itself might get diminished. In this work, we investigate how this manifests itself on the entire influence distribution.

3. Related work

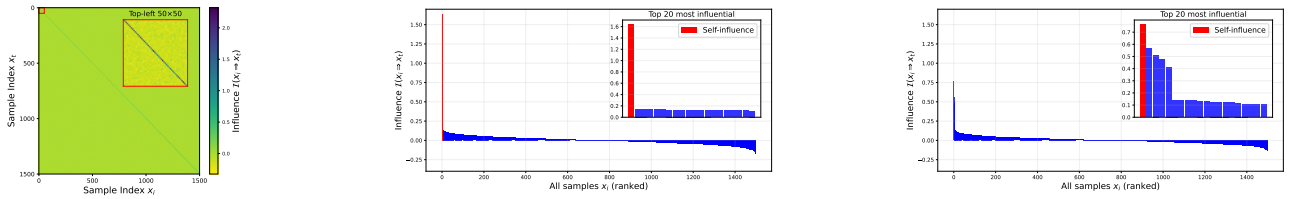
Extractability. A key risk arising from LLM memorization is the possibility of *extracting* training sequences from a model's outputs (Carlini et al., 2021). This phenomenon has hence been used to study memorization. Nasr et al. (2023) distinguish between two types: *discoverable* memorization, when prompted on a training data prefix, the model's greedy output exactly matches the corresponding suffix (as in Carlini et al. 2022); and *extractable* memorization, where training data can be generated from any prompt. Ippolito et al. (2023) argues that exact matching is too strict, and propose *approximate memorization* based on similarity metrics (e.g. BLEU). Hayes et al. (2025) further extend this to stochastic decoding, quantifying the chance of generating a target sequence across multiple non-greedy samples.

Membership Inference Attacks (MIAs). Another approach to study memorization measures the performance of MIAs (Shokri et al., 2017), inferring whether a sample was seen during training. MIAs offer an effective measure of privacy risk, as defending against them also protects against stronger attacks like data reconstruction (Salem et al., 2023). Attacks range from white-box methods (Wang et al., 2023) to black-box probes using model logits (Yeom et al., 2018; Carlini et al., 2019; Mattern et al., 2023; Shi et al., 2024). MIAs have been studied for LLM pretraining (Shi et al., 2024; Meeus et al., 2024a), albeit with limited success (Duan et al., 2024), and in finetuning (Mireshghallah et al., 2022). Prior work has found that longer sequences repeated more often are more at risk (Kandpal et al., 2022; Meeus et al., 2024b). Importantly for this work, Shilov et al. (2024) show that the presence of near-duplicates makes records more susceptible against MIAs.

Influence functions. Introduced in robust statistics (Hampel, 1974), influence functions have been studied in the context of machine learning. Koh & Liang (2017) was the first to use influence functions for *data attribution*, i.e. which training samples x_i make the model generate its prediction for x_t where x_t is typically a held-out test sample. Since computing the true counterfactual (e.g., retraining without x_i) quickly becomes prohibitively expensive, they proposed an efficient second-order approximation using Hessian-vector products. Later work developed lighter-weight alternatives, such as TracIn (Pruthi et al., 2020), which accumulates gradient-similarity scores over checkpoints. Recent work extends these methods to LLMs, to study generalization (Grosse et al., 2023) or to estimate the value of individual records (Choe et al., 2024). Here, we apply influence to study memorization by considering target records x_t from the training data, using the counterfactual definition (Zhang et al., 2023) in a small-scale setting.

4. Experimental setup

Model and dataset. We study influence for GPT-NEO 1.3B (Gao et al., 2020) further pretrained on a subset of the Natural Questions dataset (Kwiatkowski et al., 2019) (training details in Appendix A). We denote the entire dataset as $\mathcal{D}$, and consider each record $x_i = (q_i, a_i) \sim \mathcal{D}$ containing question q_i and respective answer a_i. For training, we con-



(a) Influence matrix I (b) x_t is a *unique record.* (c) x_t is a *record with near-duplicates.*

Figure 1: Influence of training on x_i on the model's prediction for x_t ($\mathcal{I}(x_i \Rightarrow x_t)$). Results for all $1,500$ records in D_t (subset from Natural Questions, with artificial near-duplicates), and $M = 1,000$ GPT-NEO 1.3B models: (a) Influence matrix I; (b–c) Influence distributions over all x_i for selected target records x_t, either (b) unique or (c) with near-duplicates.

Record type (x_t)	Self-influence $\mathcal{I}(x_t \Rightarrow x_t)$	BLEU	Top-1 Influence Margin $\mathrm{IM}(x_t)$
Unique records	1.410 ± 0.286	0.070 ± 0.114	9.105 ± 1.1369
Records with near-duplicates	0.495 ± 0.133	0.363 ± 0.313	1.292 ± 0.321

Table 1: Mean and standard deviation for key statistics of unique records and records with near-duplicates; self-influence $\mathcal{I}(x_t \Rightarrow x_t)$, BLEU score (approximate extraction through greedy decoding) and Top-1 Influence Margin $\mathrm{IM}(x_t)$.

catenate the question and answer to one string, i.e. 'Q: $\{q_i\}$ A: $\{a_i\}$'. We randomly select $1,000$ samples x_i from $\mathcal{D}$ to form D_{unique}. We refer to these samples as *unique records*.

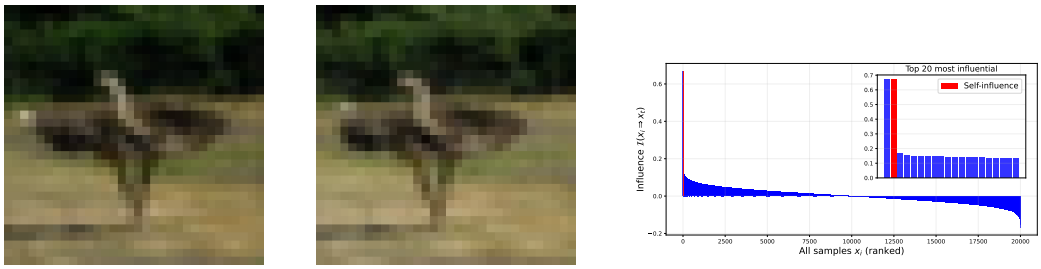
Near-duplicates. We are particularly interested in the impact of near-duplicates on commonly used memorization metrics. To this end, in addition to the unique records, we artificially craft near-duplicates and add these to an overall target dataset D_t. We randomly sample $C = 100$ records, denoted as $c_i; i = 1 \dots C$. For each $c_i = (q_i, a_i)$, we consider a set of $n_{\text{dup}} = 5$ near-duplicates for the answer a_i. Specifically, we generate samples c_{ij} for $j = 1, \dots, n_{\text{dup}}$ with (q_i, a_{ij}), where: $c_{i1} = a_i$ is the actual ground truth answer from the dataset, and the remaining $n_{\text{dup}} - 1$ answers are near-duplicates of a_i. Many strategies exist for generating near-duplicates (e.g., token replacements, semantic rephrasing), as studied by Shilov et al. (2024). We here apply their algorithm $\mathcal{A}_{\text{replace}}$ to craft near-duplicates as authors show these to contribute substantially to memorization. Namely, we replace one randomly selected token from a_i with another random one from the models' vocabulary. We then create a target dataset D_t by concatenating the unique records D_{unique} and the $n_{\text{dup}} \times C$ near-duplicates. In total, D_t consists of $N_t = 1,500$ records.

Measuring influence. We compute the influence between all samples in D_t, forming influence matrix $I = [\mathcal{I}(x_i \Rightarrow x_t)]_{i,t=1}^{N_t}$. Each entry represents the influence of training on x_i on the model's prediction for x_t. As a result, I is a square matrix of size $N_t \times N_t$ with self-influence values on its diagonal. To compute all entries, we follow Zhang et al. (2023) and approximate the expectations in Equation 1 empirically using a set of M trained models A_j, where $j = 1, \dots, M$. For each sample $x_i \in D_t$, we generate a binary inclusion vector of length M, $[p_{i1}, \dots, p_{iM}]$, where each entry $p_{ij} \in \{0, 1\}$ is independently sampled with probability 0.5 of being 1. Repeating this for all N_t samples yields partition matrix $P \in \{0,1\}^{N_t \times M}$, where $p_{ij} = 1$ indicates that x_i is included in training dataset D_j used to train model A_j. We train all models A_j, each trained on D_j with an average size of $\frac{N_t}{2}$ records. We then compute all entries in I, estimating the expectations of the losses $\mathcal{L}_{A_j}(x)$ by averaging across all respective models A_j, for which we on average have $\frac{M}{2}$ models. We analyze the accuracy of our influence estimate using $M = 1,000$ in Appendix B.

Measuring extraction. In line with Ippolito et al. (2023), we measure near-exact extraction computing the BLEU score between the ground truth answer and the text generated by the finetuned model using greedy decoding when prompted on the same question, i.e. prompted with 'Q: $\{q_i\}$ A:'. A large BLEU score represents a high similarity between ground truth and generation. We compute BLEU using Python's `nltk` package and its default parameters.

5. Results

Figure 1a illustrates influence matrix I, with each entry representing the influence of x_i (x-axis) on target x_t (y-axis) across D_t (containing both unique records and records with near-duplicates). We observe a clear diagonal pattern: self-influence values $\mathcal{I}(x_t \Rightarrow x_t)$ tend to be larger than others. This aligns with Equation 1, as including x_t in the training dataset of a model likely substantially reduces the model loss for x_t. Beyond self-influence, influence can vary widely, reaching values as large as 2 (indicating that x_i helps the model predict x_t) as well as reaching values < 0 (indicating x_i degrades the model prediction on x_t). Figure 1b further shows the ranked influence values for all x_i



(a) $\mathcal{I}(x_t \Rightarrow x_t) = 0.6680$ (b) $\mathcal{I}(x_i \Rightarrow x_t) = 0.6681$ (c) Influence distribution

Figure 2: Identifying near-duplicates in CIFAR-10 through the influence distribution: (a) the target sample x_t with its self-influence value $\mathcal{I}(x_t \Rightarrow x_t)$; (b) the most influential sample different from the sample itself ($x_i \neq x_t$) with its influence value $\mathcal{I}(x_i \Rightarrow x_t)$; (c) the full influence distribution for all x_i for target record x_t. x_t has the smallest Top-1 Influence Margin $\mathrm{IM}(x_t)$ for all samples for which $\mathcal{I}(x_t \Rightarrow x_t)$ was larger that the median.

on a randomly selected target x_t (unique record). Consistent with I's diagonal, we find self-influence to be the largest, confirming that the target sample itself contributes most to its own prediction. Other samples show a broad distribution; many contribute positively and help the model predict on x_t, while some also exert negative influence.

We then compare this influence distribution to the one achieved for a record x_t with near-duplicates. Figure 1c shows a compelling difference: while the self-influence remains the largest, several other samples x_i show similarly large influence. We confirm that the 5 largest values stem from the target x_t (self-influence) and from the 4 artificial near-duplicates as x_i. This exemplifies how near-duplicates can manifest themselves in the influence distribution.

We further compare all unique records and all records with near-duplicates for several key statistics (Table 1). We first observe that self-influence is substantially smaller for samples with near-duplicates (0.495) than for unique records (1.410). This suggests that when near-duplicates are present in the target dataset, the contribution of the exact target record diminishes. Importantly, this may lead to an underestimation of memorization if relying solely on self-influence.

Yet, we find that records with near-duplicates are *substantially more extractable* than unique ones. The mean BLEU score for near-duplicate records is 0.363, or 5 times more than the 0.070 observed for unique records (examples in Appendix C). Since extractability reflects a tangible risk associated with LLM memorization, the inability of self-influence to capture this highlights its limitations as a metric for memorization in the presence of near-duplicates.

To further characterize this, we also compute the ratio of the first to the second largest influence, or *Top-1 Influence Margin (IM)*: $\mathrm{IM}(x_t) = \frac{\max_i \mathcal{I}(x_i \Rightarrow x_t)}{\max_{i \neq i^\star} \mathcal{I}(x_i \Rightarrow x_t)}$, where $i^\star = \arg\max_i \mathcal{I}(x_i \Rightarrow x_t)$. This captures how dominant the most influential training sample is, as for instance also applied to calibrate confidence in identification learning (Tournier & De Montjoye, 2022). Table 1 shows that the most influential sample is on average 9.1 times larger than the second one for unique records, compared to 1.3 for records with near-duplicates.

Together, we show that near-duplicates can be more extractable despite lower self-influence. The full influence distribution reveals more informative patterns, e.g. reducing self-influence dominance in favor of near-duplicates, highlighting the value of distribution-level analysis for understanding memorization.

Identifying near-duplicates in CIFAR-10. We examine the influence distribution for a ResNet (He et al., 2016) model trained on real-world image dataset CIFAR-10 (Krizhevsky et al., 2009). We train $M = 1,000$ models considering a randomly sampled subset of $N_t = 20,000$ records (training details Appendix D) and compute the full influence matrix I. Our goal is to assess whether similar patterns in the influence distribution – as found above in our artificially created setup – also emerge in a real-world dataset. To do so, we compute the Top-1 Influence Margin $\mathrm{IM}(x_t)$ for each target record x_t, quantifying how strongly the most influential training sample dominates the influence distribution. A low value of $\mathrm{IM}(x_t)$ indicates that no single training point exerts overwhelming influence, which we hypothesize may correspond to the presence of meaningful near-duplicates.

Figure 2 shows the target record with the smallest $\mathrm{IM}(x_t)$ (for which the self-influence was greater than the median). We find that the target's most influential sample is a natural, visually compelling near-duplicate. We also confirm this for the 5 other targets with the lowest $\mathrm{IM}(x_t)$ in Appendix D. These findings suggest that, also in real-world datasets, the influence distribution is heavily impacted by the presence of near-duplicates. Hence, properly studying how such samples are memorized by ML models likely requires considering influence beyond self-influence alone.

Impact Statement

A deeper understanding of how LLMs memorize training data is critical for addressing questions around generalization and data attribution, but also to assess privacy risks and concerns around copyright or benchmark contamination. As models scale and are trained on massive, sometimes redundant datasets, the presence of near-duplicates challenges conventional ways of measuring memorization. Our work demonstrates that the commonly used self-influence might fall short to study memorization in the presence of near-duplicates. By analyzing the full influence distributions instead, we uncover more nuanced memorization patterns. By treating influence as a distributional property, we believe our work opens new directions for measuring, interpreting, and ultimately controlling how models internalize data.

References

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. In *The Eleventh International Conference on Learning Representations*, 2022.

Choe, S. K., Ahn, H., Bae, J., Zhao, K., Kang, M., Chung, Y., Pratapa, A., Neiswanger, W., Strubell, E., Mitamura, T., et al. What is your data worth to gpt? llm-scale data valuation with influence functions. *arXiv preprint arXiv:2405.13954*, 2024.

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

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Grosse, R., Bae, J., Anil, C., Elhage, N., Tamkin, A., Tajdini, A., Steiner, B., Li, D., Durmus, E., Perez, E., et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

Hampel, F. R. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

Hayes, J., Swanberg, M., Chaudhari, H., Yona, I., Shumailov, I., Nasr, M., Choquette-Choo, C. A., Lee, K., and Cooper, A. F. Measuring memorization in language models via probabilistic extraction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9266–9291, 2025.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ippolito, D., Tramer, F., Nasr, M., Zhang, C., Jagielski, M., Lee, K., Choo, C. C., and Carlini, N. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pp. 28–53, 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.

Kandpal, N., Deng, H., Roberts, A., Wallace, E., and Raffel, C. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pp. 15696–15707. PMLR, 2023.

Karamolegkou, A., Li, J., Zhou, L., and Søgaard, A. Copyright violations and large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Krizhevsky, A. et al. Learning multiple layers of features from tiny images. 2009.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin,

J., Lee, K., et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Liu, K. Z., Choquette-Choo, C. A., Jagielski, M., Kairouz, P., Koyejo, S., Liang, P., and Papernot, N. Language models may verbatim complete text they were not explicitly trained on. *arXiv preprint arXiv:2503.17514*, 2025.

Mattern, J., Mireshghallah, F., Jin, Z., Schölkopf, B., Sachan, M., and Berg-Kirkpatrick, T. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*, 2023.

Meeus, M., Jain, S., Rei, M., and de Montjoye, Y.-A. Did the neurons read your book? document-level membership inference for large language models. In *Proceedings of the 33rd USENIX Conference on Security Symposium*, pp. 2369–2385, 2024a.

Meeus, M., Shilov, I., Faysse, M., and de Montjoye, Y.-A. Copyright traps for large language models. In *Forty-first International Conference on Machine Learning*, 2024b.

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.

Mirzadeh, S. I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., and Farajtabar, M. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Oren, Y., Meister, N., Chatterji, N. S., Ladhak, F., and Hashimoto, T. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, 2019.

Pruthi, G., Liu, F., Kale, S., and Sundararajan, M. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.

Salem, A., Cherubin, G., Evans, D., Köpf, B., Paverd, A., Suri, A., Tople, S., and Zanella-Béguelin, S. Sok: Let the privacy games begin! a unified treatment of data inference privacy in machine learning. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 327–345. IEEE, 2023.

Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., and Zettlemoyer, L. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Shilov, I., Meeus, M., and de Montjoye, Y.-A. Mosaic memory: Fuzzy duplication in copyright traps for large language models. *arXiv preprint arXiv:2405.15523*, 2024.

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

Tournier, A. J. and De Montjoye, Y.-A. Expanding the attack surface: Robust profiling attacks threaten the privacy of sparse behavioral data. *Science advances*, 8(33):eabl6464, 2022.

Wang, J., Wang, J., Li, M., and Neel, S. Mope: Model perturbation-based privacy attacks on language models. In *Socially Responsible Language Modelling Research*, 2023.

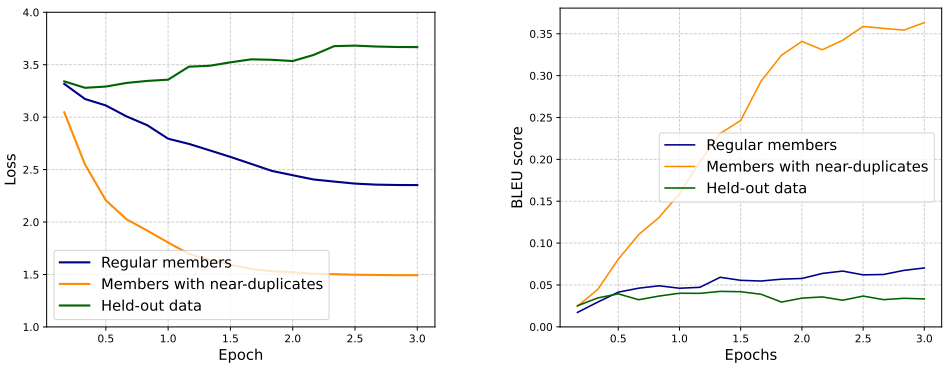
Wang, X., Antoniades, A., Elazar, Y., Amayuelas, A., Albalak, A., Zhang, K., and Wang, W. Y. Generalization vs memorization: Tracing language models' capabilities back to pretraining data. *arXiv preprint arXiv:2407.14985*, 2024.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282. IEEE, 2018.

Zhang, C., Ippolito, D., Lee, K., Jagielski, M., Tramèr, F., and Carlini, N. Counterfactual memorization in neural language models. *Advances in Neural Information Processing Systems*, 36:39321–39362, 2023.

A. Training details

Starting from its open-sourced pretrained version, we finetune GPT-NEO 1.3B (Gao et al., 2020) using `bfloat16` precision, for 3 epochs, using a linear learning rater scheduler with as initial learning rate $2e-4$ and weight decay 0.01, batch size 50 and maximum sequence length of 100. Figure 3 shows the loss curves and the BLEU scores for one GPT-NEO 1.3B model trained on the target dataset D_t following the experimental setup as detailed in Section 4. We report the cross-entropy loss and BLEU scores between the generations and ground truth completions over training for (i) the regular members (1000 random samples from the entire Natural Questions dataset), (ii) 100 members with near-duplicates and (iii) 100 random samples not included in the training dataset. As expected, we find that the loss for regular members decreases over training, with a sharper drop for members with near-duplicates. The loss for the held-out samples increases slightly throughout training. Similarly, the BLEU scores for regular members increases throughout training, particularly strong for members with near-duplicates, while for held-out data the BLEU scores remain low throughout.



(a) (b)

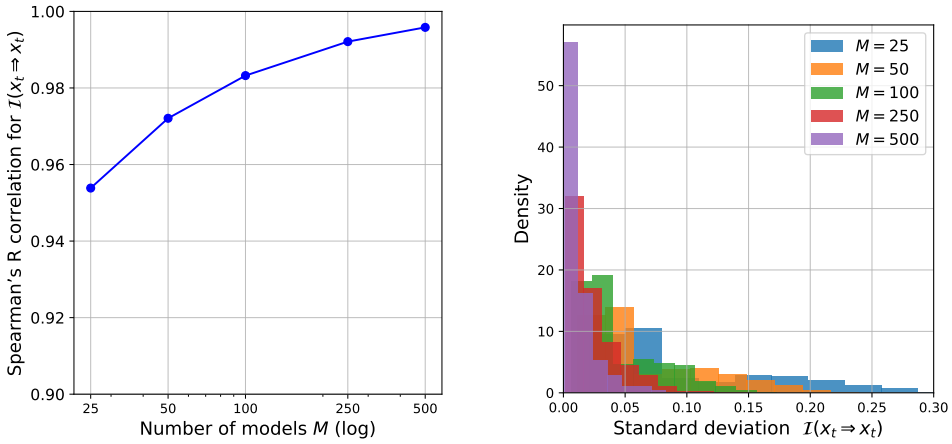
Figure 3: Loss curves (a) and BLEU scores (b) for GPT-NEO 1.3B trained on the entire target dataset D_t, as described in Section 4.

B. Number of models M required to reliably estimate influence

To understand the number of models M needed to reliably estimate the influence values $\mathcal{I}(x_i \Rightarrow x_t)$ (expectations from Equation 1), we replicate the analysis from (Zhang et al., 2023). Specifically, we take the entire set of results obtained by using $M_{\text{max}} = 1,000$ models in our main experiment, and consider multiple equally sized $\frac{M_{\text{max}}}{M}$ partitions each consisting of M models, for an increasing value of M. We then compute the mean Spearman's R correlation between all self-influence values computed using M models across $\frac{M_{\text{max}}}{M}$ partitions. Figure 4(a) shows how the correlation quickly approaches 1.0, meaning that for a large number of models, the relative order of self-influence values stabilizes. We further compute the standard deviation of self-influence values for each of the N_t samples across partitions when using M models. Figure 4(b) shows how the standard deviation quickly decreases when M increases, reaching negligible values for $M = 500$. Both results make us confident that using $M = M_{\text{max}} = 1,000$ throughout our experiments yield reliable influence estimates for further analysis, especially as this is also substantially more than the $M = 400$ models used by Zhang et al. (2023).

C. Examples extraction

Table 2 include a selected sample of records x_i from the Natural Questions dataset (Kwiatkowski et al., 2019), where $x_i = (q_i, a_i)$ consists of question q_i and reference answer a_i. We provide the answer generated from the finetuned model (on the entire dataset D_t) following the same experimental details as laid out in Section 4. We generate an answer from the model using greedy decoding prompted on 'Q: $\{q_i\}$ A:' to get a'_i and report the corresponding BLEU score between the reference answer a_i and the generated one a'_i. We report one sample for each kind of record x_t: a regular member only included once in D_t, a member for which the record was included in D_t alongside its $n_{\text{dup}} - 1$ near-duplicates and a held-out record not part of D_t. We find that the BLEU score of a record with near-duplicates is substantially larger (0.499) than a record only included once (0.086), and higher than a record not seen during training (0.062). Notably, even for the



(a) (b)

Figure 4: Measuring the reliability of computed influence values for increasing number M of models trained (GPT-NEO 1.3B finetuned on a subset from the Natural Questions Dataset). (a) Mean Spearman's R correlation between all self-influence values across $\frac{M_{\text{max}}}{M}$ partitions for increasing M. (b) Distribution of standard deviation of self-influence values for all samples across $\frac{M_{\text{max}}}{M}$ partitions for increasing M.

record with near-duplicates, the model does not generate the reference answer exactly (*extraction* from Carlini et al. (2021)), yet does generate an answer with substantial overlap, e.g. *"He was looking for a name that would make it attractive to the American market"*.

D. Additional results for CIFAR-10

We train $M = 1,000$ ResNet (He et al., 2016) models on a subset from CIFAR-10 (Krizhevsky et al., 2009). Each model is trained for 30 epochs using SGD (momentum 0.9, weight decay $5e-4$), with a batch size of 128 and a linear learning rate schedule that warms up over 5 epochs to 0.5 before decaying to zero by the end of training. Figure 7 shows the loss curve and accuracy for one ResNet model trained on the entire dataset D_t containing $20,000$ samples from CIFAR-10. Figure 8 further shows the sensitivity of the computed influence values for an varying number of models M, replicating the analysis from Appendix B for the CIFAR-10 dataset.

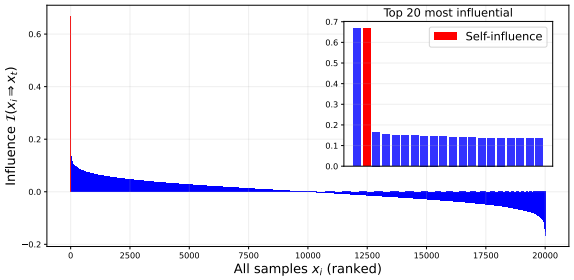
Figures 5 and 6 visualize samples with the smallest Top-1 Influence Margin $\text{IM}(x_t)$, highlighting that for these samples, the most influential yet distinct sample appears to be a visual near-duplicate.

Record type (x_t)	Question q_i	Reference answer a_i	Generated answer a_i'	BLEU score
Regular member	what age did brett favre retire from football	Brett Favre On January 2, 2011, Favre was unable to play against the Detroit Lions in the final game of the regular NFL season due to his inability to pass NFL-mandated post-concussion tests. In a press conference immediately following the game, Favre announced his intention to retire from professional football. On January 17, 2011, Favre officially filed his retirement papers with the NFL.[101]	Brett Favre (football) In October 2008, Favre announced his retirement from professional football after 11 seasons in the National Football League (NFL). He was the oldest player to announce his retirement, at the age of 39. Favre's retirement announcement was made during a press conference at the 2008 NFL Honors ceremony in the East Room of the White House...	0.0864
Member with near-duplicates	where did the name chilean sea bass come from	Patagonian toothfish The name "Chilean Seabass" was invented by a fish wholesaler named Lee Lantz in 1977. He was looking for a name that would make it attractive to the American market. He considered "Pacific sea bass" and "South American sea bass" before settling on "Chilean sea bass"...	(seabass) The name "Chilean Seabass" was invented by a fish wholesaler named John L. L. Smith in the 1960s. He was looking for a name that would make it attractive to the American market. He considered "Pacific sea bass" and "South American sea bass" before settling on "Chilean sea bass"...	0.4991
Held-out test sample	who played legolas in lord of the rings	Orlando Bloom Orlando Jonathan Blanchard Bloom (born 13 January 1977)[1] is an English actor. After having his breakthrough as Legolas in The Lord of the Rings trilogy, he rose to fame by further appearing in epic fantasy, historical epic, and fantasy adventure films. His other roles include Legolas in The Hobbit trilogy, Will Turner in the Pirates of the Caribbean series, and Paris in Troy (2004).	Legolas Legolas is a fictional character in the A Song of Ice and Fire novel The Lord of the Rings. He is portrayed by Anthony Gonzalez in the film trilogy. He is a half-elven character, half-human/half-elf, and is the son of Legolas and a half-elven princess. He is a skilled	0.0618

Table 2: Examples of finetuned model output (GPT-NEO 1.3B) versus ground truth (Natural Questions dataset) across record types, alongside the corresponding BLEU score.

(a) $\mathcal{I}(x_t \Rightarrow x_t) = 0.66801$

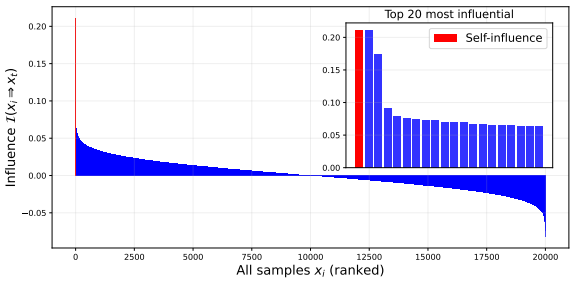
(b) $\mathcal{I}(x_i \Rightarrow x_t) = 0.66805$



(c) Influence distribution

(d) $\mathcal{I}(x_t \Rightarrow x_t) = 0.21160$

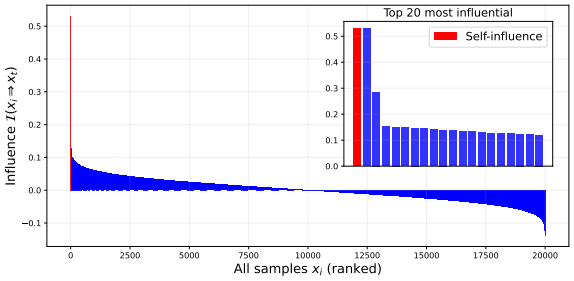
(e) $\mathcal{I}(x_i \Rightarrow x_t) = 0.21155$



(f) Influence distribution

(g) $\mathcal{I}(x_t \Rightarrow x_t) = 0.530531$

(h) $\mathcal{I}(x_i \Rightarrow x_t) = 0.53012$

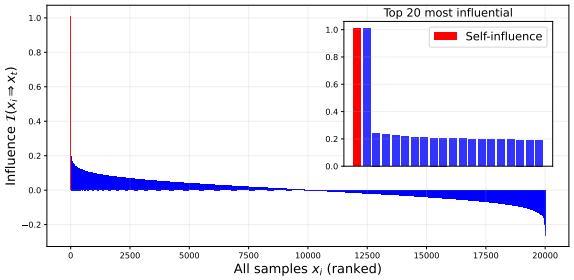


(i) Influence distribution

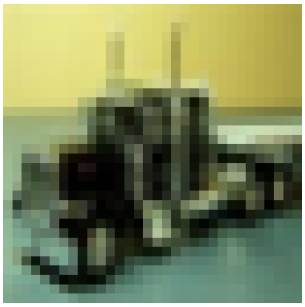
Figure 5: Identifying near-duplicates in CIFAR-10 through the distribution of counterfactual influence. For each row, from left to right: (i) the target sample x_t with its self-influence value $\mathcal{I}(x_t \Rightarrow x_t)$; (ii) the most influential sample different from the sample itself ($x_i \neq x_t$) with its influence value $\mathcal{I}(x_i \Rightarrow x_t)$; the full influence distribution for all x_i for target record x_t. For all samples for which $\mathcal{I}(x_t \Rightarrow x_t)$ was larger that the median of all self-influence values, showing the x_t with the **top 1-3** smallest Top-1 Influence Margin $\mathrm{IM}(x_t)$.

(a) $\mathcal{I}(x_t \Rightarrow x_t) = 1.00965$

(b) $\mathcal{I}(x_i \Rightarrow x_t) = 1.00702$

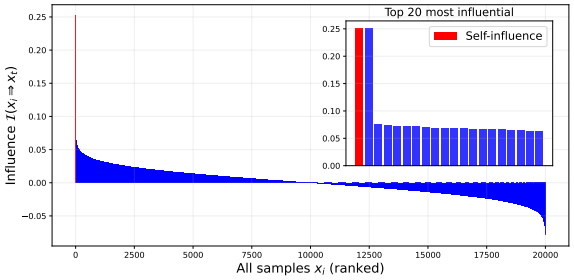


(c) Influence distribution



(d) $\mathcal{I}(x_t \Rightarrow x_t) = 0.25191$

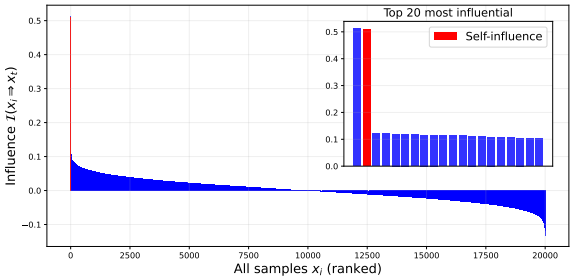
(e) $\mathcal{I}(x_i \Rightarrow x_t) = 0.251156$



(f) Influence distribution

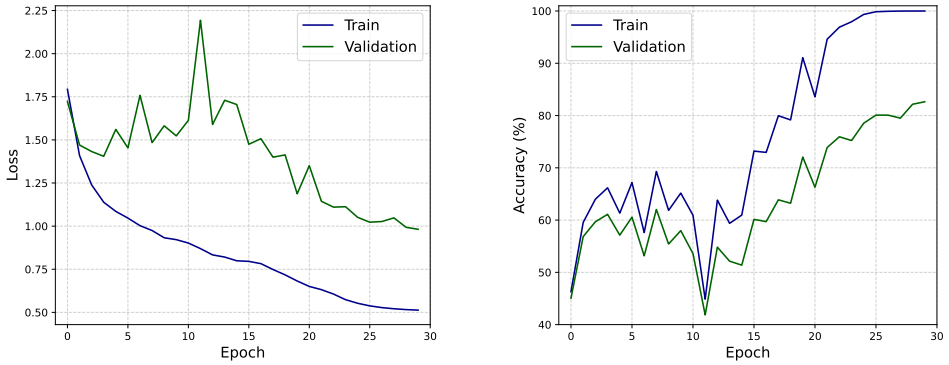
(g) $\mathcal{I}(x_t \Rightarrow x_t) = 0.50944$

(h) $\mathcal{I}(x_i \Rightarrow x_t) = 0.51317$



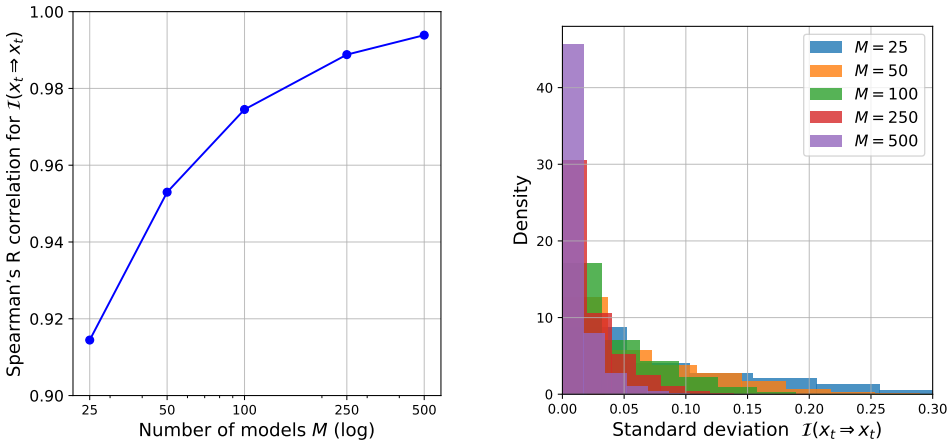
(i) Influence distribution

Figure 6: Identifying near-duplicates in CIFAR-10 through the distribution of counterfactual influence. For each row, from left to right: (i) the target sample x_t with its self-influence value $\mathcal{I}(x_t \Rightarrow x_t)$; (ii) the most influential sample different from the sample itself ($x_i \neq x_t$) with its influence value $\mathcal{I}(x_i \Rightarrow x_t)$; the full influence distribution for all x_i for target record x_t. For all samples for which $\mathcal{I}(x_t \Rightarrow x_t)$ was larger that the median of all self-influence values, showing the x_t with the **top 3-6** smallest Top-1 Influence Margin $\mathrm{IM}(x_t)$.



(a) (b)

Figure 7: Loss curves (a) and accuracy (b) for CIFAR-10 trained on the entire target dataset D_t (containing $N_t = 20,000$ random samples from CIFAR-10's training data). We compute the validation performance on $5,000$ random samples from CIFAR-10's test data.



(a) (b)

Figure 8: Measuring the reliability of computed influence values for increasing number M of models trained (ResNet model trained on CIFAR-10). (a) Mean Spearman's R correlation between all self-influence values across $\frac{M_{\max}}{M}$ partitions for increasing M. (b) Distribution of standard deviation of self-influence values for all samples across $\frac{M_{\max}}{M}$ partitions for increasing M.