# Strong Membership Inference Attacks on Massive Datasets and (Moderately) Large Language Models

**Jamie Hayes**[1], **Adam Dziedzic**[2], **A. Feder Cooper**[3,4], **Christopher A. Choquette-Choo**[1], **Franziska Boenisch**[2], **Georgios Kaissis**[1], **Igor Shilov**[5], **Ilia Shumailov**[1], **Katherine Lee**[1], **Matthew Jagielski**[1], **Matthieu Meeus**[5], **Meenatchi Sundaram Muthu Selva Annamalai**[6], **Niloofar Mireshghallah**[7], **Yves-Alexandre de Montjoye**[5], **Milad Nasr**[1]
[1]Google DeepMind, [2]CISPA Helmholtz Center for Information Security,
[3]Microsoft Research, [4]Stanford University, [5]Imperial College London,
[6]University College London, [7]Carnegie Mellon University

## Abstract

State-of-the-art membership inference attacks (MIAs) typically require training many reference models, making it difficult to scale these attacks to large pre-trained language models (LLMs). As a result, prior research has either relied on weaker attacks that avoid training reference models (e.g., fine-tuning attacks), or on stronger attacks applied to small-scale models and datasets. However, weaker attacks have been shown to be brittle—achieving close-to-arbitrary success—and insights from strong attacks in simplified settings do not translate to today's LLMs. These challenges have prompted an important question: are the limitations observed in prior work due to attack design choices, or are MIAs fundamentally ineffective on LLMs? We address this question by scaling LiRA—one of the strongest MIAs–to GPT-2 architectures ranging from 10M to 1B parameters, training reference models on over 20B tokens from the C4 dataset. Our results advance the understanding of MIAs on LLMs in three key ways: (1) strong MIAs can succeed on pre-trained LLMs; (2) their effectiveness, however, remains limited (e.g., $\text{AUC} < 0.7$) in practical settings; and, (3) the relationship between MIA success and related privacy metrics is not as straightforward as prior work has suggested.

## 1 Introduction

In a membership inference attack (MIA), an adversary aims to determine whether a specific data record was part of a model's training set [43, 51]. MIAs pose a significant privacy risk to ML models, but state-of-the-art attacks are often too computationally expensive to run at the scale of pre-trained large language models (LLMs). This is because strong MIAs require training multiple "reference" models to calibrate membership predictions—and pre-training even one LLM is often prohibitively expensive in research settings. As a result, current work makes one of two compromises: running weaker attacks that avoid training reference models (e.g., attacks that fine-tune an LLM), or running strong attacks that train small reference models on small datasets. However, both exhibit notable limitations (Section 2). Weaker attacks are more practical, but they have been shown to be brittle—often performing no better than random guessing [13, 15, 34]. Stronger attacks, when run in simplified settings, fail to capture the complex dynamics of large-scale, pre-trained language models; as a result, their insights do not reliably generalize to modern LLMs [29].

Results from both of these approaches leave key questions unanswered about the effectiveness of MIAs on LLMs. In particular, *are the fidelity issues of weaker attacks due to omitting reference models, or do they point to a deeper, more fundamental challenge with applying membership*

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

*inference to large language models?* Current research has not offered an answer because, to date, there are no baselines of how stronger MIAs perform on large-scale, pre-trained LLMs.

In this paper, we bridge this gap by running stronger attacks at a scale significantly larger than previously explored. We pre-train over 4,000 GPT-2–like reference models, ranging from 10 million to 1 billion parameters [23], on subsets of the C4 dataset [38] that are *three orders of magnitude larger than those used in prior MIA studies*—up to 100 million examples, compared to fewer than 100,000 in previous work [31]. We use these models to conduct a detailed investigation of the Likelihood Ratio Attack (LiRA) [3], one of the strongest MIAs in the literature. This substantial effort proves worthwhile, as we uncover three key insights that advance the state of the art in understanding the potency and reliability of membership inference attacks on large language models:

- **Strong membership inference attacks can succeed on pre-trained LLMs.** We are the first to execute strong attacks at this scale, and find that LiRA—in contrast to weaker fine-tuning attacks—can easily beat random baselines (Section 3.1). Our results on Chinchilla-optimal models (trained for 1 epoch) exhibit a non-monotonic relationship between model size and MIA vulnerability: larger models are not necessarily more at risk (Section 3.2).
- **The overall success of strong MIAs is limited on pre-trained LLMs.** Even though we demonstrate that LiRA can succeed at LLM scale, we are only able to achieve impressive results (i.e., $\mathrm{AUC} \geq 0.7$) when diverging from typical training conditions—specifically, by varying training-dataset sizes and training for multiple epochs (Section 4.1).
- **The relationship between MIA success and related privacy metrics is not straightforward.** We find that examples seen later in training tend to be more at risk (Section 5.1); however, this trend is complicated by sample length, which also affects vulnerability. We also study if there is any relationship between training-data extraction and MIA, and observe no correlation with MIA success. This suggests that the two privacy attacks capture different signals related to memorization (Section 5.2).

Altogether, our contributions serve not only as an extensive benchmark of strong MIAs on pre-trained LLMs, but also provide some initial answers to urgent open questions about the conditions under which MIAs exhibit a threat to privacy for language models. Our work also quantifies the performance gap between weaker (more feasible) and stronger attacks, establishing an upper bound for what weaker attacks could realistically achieve in this setting. Our hope is that this guides future research on MIA, informing the development of stronger and more practical attacks, as well as more effective defenses.

## 2 Background and related work

**Membership inference** is a key approach for assessing empirical privacy and information-leakage risks in ML models. The most effective attacks calibrate their predictions based on how models behave on specific data points [43, 51]. Using the **target model**'s architecture and training setup, the attacker trains multiple **reference models** on different subsets of the training data. The attacker queries each reference model with a given data point and computes a membership inference **score** from the model's output (e.g., loss or logit). By comparing these scores across reference models, the attacker learns how the score distributions differ between **members** (**in** the training data) and **non-members** (unseen data, **out** of the training data). The attacker can use this signal to infer membership of examples in the target model's training set [3, 40, 47, 50, 52].

The number of reference models necessary for successful attacks varies across methods—from tens or hundreds for the Likelihood Ratio Attack (LiRA) [3] and Attack-R [50], to as few as 1 or 2 for the Robust Membership Inference Attack (RMIA) [52]. (See Appendix A.1.) While these attacks have been successfully applied to smaller settings, they are often considered impractical for contemporary language models due to the prohibitive computational cost of training even a single reference LLM. As a result, prior work attempts to approximate stronger, reference-model-based attacks in various ways.

**Small-scale, strong, reference-based attacks.** Song and Shmatikov [44] were the first to train (10) reference models, in order to evaluate privacy in smaller language models (RNNs). However, insights from such settings do not translate to today's LLMs [31], as the training dynamics differ significantly. Other work has applied MIAs using only a single reference model for small, pre-trained masked language models [33], but this approach reduces attack precision, as effective calibration of membership predictions becomes more difficult with fewer reference models.

**Larger-scale, weak, reference-free attacks.** To avoid the cost of training reference models, weaker attacks consider a range of signals to infer membership, typically leveraging black-box access to the model. For example, Yeom et al. [51] use model loss computed on the target example, Carlini et al. [2] use normalized model loss and `zlib` entropy of the target example, and Mattern et al. [28] compare the model loss to the loss achieved for neighboring samples. More recent work experiments with token probabilities [42, 54] and changes in loss based on prompting with different context [46, 49].

Beyond black-box query-access, other work attempts to derive membership signal from changing the model. For instance, prior work has perturbed inputs or model parameters and observed resulting changes in model loss on the target, or used (parameter-efficient) fine-tuning on domain-specific datasets to detect privacy or confidentiality risks [6, 15, 20, 25, 32, 34, 37, 39]. However, fine-tuning-based attacks introduce *new* data to the problem setup, which may complicate the validity of using MIAs to detect benchmark contamination [12, 26, 27, 36] and to draw reliable conclusions about other sensitive data issues [7–10, 14, 22, 30, 42, 48, 53].

Further, a recent approach that evaluates attacks on LLMs using post-hoc collected datasets also exhibits serious limitations. While prior work has reported high success rates on a variety of models and datasets (AUC $\approx 0.8$) [29, 42, 46, 49, 54], such evaluations rely on the model's training-date cutoff as a proxy for distinguishing between member and non-member data points [27]. These newer data introduce distribution shift, which undermines the validity of the reported results [11, 13, 27, 31]. And further, as others have noted, when current MIAs are evaluated in a controlled privacy game like this, they often barely outperform random guessing [13, 31].

## 3 Examining strong MIAs in realistic settings for pre-trained LLMs

Altogether, the limitations of prior work raise the key question that motivates our work: *are the fidelity issues of weaker attacks due to omitting reference models, or do they point to a deeper, more fundamental challenge with applying membership inference to large language models?* This is a big question, so we break it down into smaller ones that we can test with specific experiments that reveal different information about the effectiveness of strong MIAs on pre-trained LLMs.

The initial step of our evaluation involved deciding which strong MIA method to use across our experiments. We evaluated two of the strongest attacks in the literature—LiRA [2] and RMIA [52]—and, for the experiments that follow, we opted to use LiRA because we observed it to be more effective in our pre-trained LLM setting. We defer details about LiRA and comparisons with RMIA to Appendix A, and focus on our results using LiRA in the remainder of the paper.

In this section, we investigate the relationship between the number of reference models and attack success (Section 3.1). Based on our results, we decide to use 128 reference models throughout all following experiments in this work. Then, we test the effectiveness of strong attacks under realistic settings—settings that reflect how LLMs are actually trained. To do so, we run LiRA using target and reference models of various sizes, which we train according to Chinchilla-scaling laws [18] (Section 3.2). Together, these experiments inform our first key result: **strong membership inference attacks can succeed on pre-trained LLMs.** In the following sections, we expand upon these results to other training and attack conditions; we will refine our first key result by investigating the limits of strong MIA success rates (Section 4), and by digging beneath these average rates to reveal how attacks impact individual-example members.

**General setup.** For all experiments, we pre-train GPT-2 architectures of varying sizes—from 10M to 1B—on subsets of the C4 dataset [38] using the open-source NanoDO library [23]. The training datasets we use are *3 orders of magnitude larger than those in prior MIA studies*: up to 50M examples, compared to fewer than 100K examples in previous work [31]. We explore datasets of this size because, while it is well established that MIA success depends on both model capacity and training-dataset size, the nature of this relationship remains unexplored at the scale of pre-trained LLMs. For each attack, we start with a fixed dataset of size $2N$ examples drawn from C4, from which we randomly subsample reference-model training sets of size $N$. For instance, if $N$ is 10M examples, we select them by randomly subsampling from a fixed dataset of 10M$\times$2=20M examples. (This means our MIA analysis runs over an overall dataset size of 50M$\times$2=100M in our largest experimental setting.) We use a different random seed for each subsample, which yields the different member (in) and non-member (out) distributions for each example that we use to run LiRA. Specific experimental configurations vary, so we introduce additional setup as needed. (See Section E for more details.)

### 3.1 Warm-up: How many reference models should we use?

To determine the number of reference models to use for all of our experiments, we train a 140M-parameter models on 7M examples. 7M examples equates to approximately 2.8B training tokens—i.e., what is optimal for this model size, according to Chinchilla scaling laws [18] with an over-training multiplier of 20.

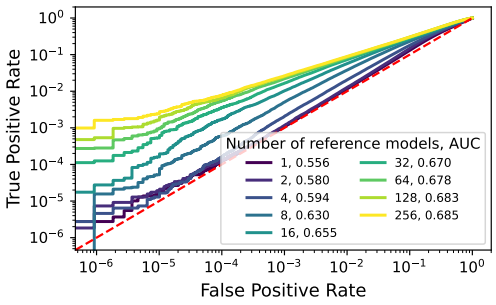



Figure 1: **LiRA with different numbers of reference models.** We attack a 140M-parameter model trained on 7M examples. As reference models increase, LiRA's performance improves (measured with ROC AUC). However, there are diminishing returns: AUC is effectively unchanged from 128 to 256 reference models.

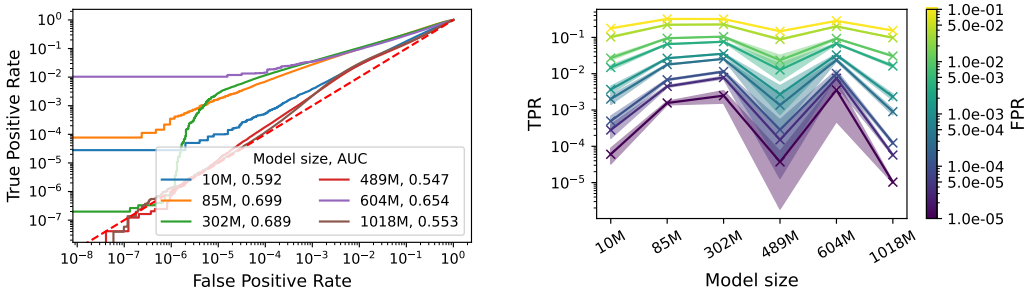
As shown in Figure 1, we test a range of reference models. The plot shows multiple Receiver Operating Characteristic (ROC) curves, indicating the True Positive Rate (TPR) for the given False Positive Rate (FPR) on a log−log scale. The Area under the Curve (AUC) is provided for each ROC curve. The dashed red line represents the baseline for which membership predictions would be effectively arbitrary (i.e., TPR and FPR are equal; AUC is 0.5). We choose to report AUC as our primary metric, as it is more challenging to visualize the TPR over a wide range of FPR in a streamlined way. (For comparison, see Figure 2b, which provides such an alternate visualization for only a limited range of FPR, at the cost of not surfacing overall AUC.) We also investigate the performance of different observation signals (Section A.2), and choose to use a sample's loss. Note that, while LiRA clearly beats the random baseline, it is not remarkably successful in this setting: regardless of the number of reference models, it never achieves an AUC of 0.7. Further, even though LiRA's attack success increases with more reference models, there are diminishing returns. From 1 to 8 reference models, the AUC has a relative increase of 13.3%; for the next 8× increase (from 8 to 64), the AUC only increases 7.6%; and, doubling from 128 to 256 only yields a 0.2% improvement.

### 3.2 Training and attacking a compute-optimal model

In practice, models are typically trained based on observed scaling laws: for a given model size, the scaling law suggests the optimal number of tokens to use for training. In this section, to assess MIA in realistic conditions for pre-trained LLMs, we attack models of various sizes that we have trained for 1 epoch, setting the number of samples to be optimal according to Chinchilla scaling [18]. Specifically, we set the number of training set tokens to be 20× larger than the number of model parameters. We *only train for 1 epoch*, a common choice in large training runs [1, 45]. Additional details about the specific training and experimental recipes are in Appendix B and Section E, such as the number of samples used in training across different model sizes.

In Figure 2, we show the results of attacking models of sizes 10M, 85M, 302M, 489M, 604M and 1018M. These model sizes come from the default configurations available in NanoDO [23]. For improved readability, we exclude the results for the 140M model in our plots in this section, as we investigated this architecture above. Note that the attack on the 140M model with 128 reference models has an AUC of 0.683, which puts its performance below both the 85M and 302M models. Interestingly, we observe that there is a non-monotonic relationship between model size and MIA vulnerability under these training conditions. In Figure 2a, the 85M-parameter model shows the highest AUC (0.699), followed by the 302M model (AUC 0.689), and then the 140M model (see Figure 1, AUC 0.683); the 489M model exhibits the lowest AUC (0.547). This is also supported in Figure 2b, which provides a different view of the same attack. Each line compares the TPR for the different-sized models at different fixed settings of FPR. From 10M to 302M, there is a consistent pattern of the TPR increasing with model size (regardless of the setting of FPR); but then, when increasing to 489M, there is a significant drop in TPR.

Before running this experiment, our expectation was that each line would look approximately horizontal, as the training-dataset size is being scaled proportionally (and optimally, according to Hoffmann et al. [18]) to model size. There are many reasons why this may not have occurred. First, the most pronounced differences in TPR are at extremely small values. Even subtle differences




(a) ROC for 6 Chinchilla-optimal models (1 epoch).

(b) TPR at fixed FPR for different model sizes.

Figure 2: **MIA vulnerability across compute-optimally trained models of different sizes.** (**a**) ROC curves using 128 reference models demonstrate varying MIA susceptibility for models with 10M (AUC 0.592), 85M (AUC 0.699), 302M (AUC 0.689), 489M (AUC 0.547), 604M (AUC 0.654) and 1018M (AUC 0.553) parameters when trained under Chinchilla-optimal conditions for 1 epoch. The 85M and 302M models shows the highest vulnerability, indicating that increasing model size does not uniformly decrease MIA risk in this setting. (**b**) How TPR (for each given FPR) varies by model size for different Chinchilla-optimal models.

in training runs may flip a few samples from correct to incorrect member predictions, which, in the low TPR regime, can have a large effect on MIA success. Second, Chinchilla scaling [18] is not the only such law. Sardana et al. [41], Hu et al. [19], and Grattafiori et al. [16] all introduce other ways to optimally select the number of training tokens for a given model. In future work, we will investigate if these other token-size-selection methods stabilize TPR as model size grows.

As we discuss below (Section 4.2), repeating this experiment by training these same architectures on a fixed dataset size exhibits vastly different results. We additionally test different training configurations; in Appendix C we alter the learning rate schedule, and observe that there is a modest effect on attack performance. (See Appendix B, where, as a sanity check, we also confirm that the larger models converge to lower loss values, reflecting their increased capacity to fit the training data.)

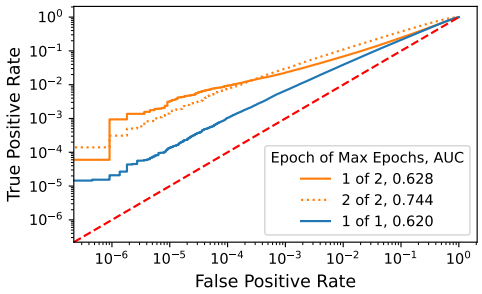
## 4 Investigating the limits of strong attacks

Even in the most successful (i.e., high AUC) case, overall performance on inferring membership is not particularly impressive when running LiRA with a large number of reference models on compute-optimal models trained for a single epoch. Similar to our experiments with LiRA and varied numbers of reference models (Figure 1), the maximum AUC we observe remains under 0.7 for all model sizes (Figure 2). This raises a natural follow-on question: if we free ourselves from the constraints of these typical training settings, will it then be possible to improve MIA success? *Can we identify an upper bound on how strong MIAs could possibly perform on pre-trained LLMs?*

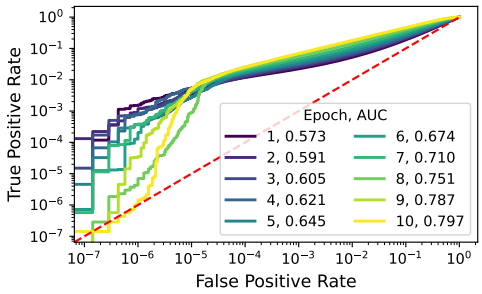
To address this question, in this section we run attacks on models trained on different-sized (i.e., not always Chinchilla-optimal) datasets (Section 4.2) for more than 1 epoch (Section 4.1). Our experiments show that diverging from typical settings can indeed improve attack success. However, while these experiments are a useful sanity check, they do not directly suggest conclusions about the effectiveness of strong MIAs in general. Instead, they suggest that there appears to be an upper bound on how well strong MIAs can perform on LLMs under practical conditions. In other words, these experiments inform our second main observation: **the success of strong MIAs is limited in typical LLM training settings.**

### 4.1 Effects of scaling the compute budget (i.e., training for more epochs)

In Figure 3a, we compare MIA ROC for the 44M model architecture under different training configurations. We keep the total number of tokens surfaced to the model during training Chinchilla-optimal, but we alter *when* these tokens are surfaced. As a baseline, we train for 1 epoch on the entire dataset; when we attack this model with LiRA, it yields an AUC of 0.620. (See Figure 3a, 1 of 1.) We then take *half* of the training dataset and train the same architecture over 2 epochs. In both settings the total num-




(a) 44M model, split dataset in half and train for 2 epochs, or train on the entire dataset for 1 epoch.




(b) 140M model, training for 10 epochs.

Figure 3: **Varying epochs while keeping the overall dataset size Chinchilla-optimal.** In (**a**), we compare training a 44M model on the whole Chinchilla-optimal dataset in 1 epoch (AUC 0.620, after 1 of 1 epoch) to training for 2 epochs on only half of the dataset (AUC 0.744, after 2 of 2 epochs). In (**b**), we train a 140M model on the whole Chinchilla-optimal dataset for 10 epochs. With more epochs, AUC increases. See main text for additional observations.

ber of training tokens is Chinchilla-optimal, however, in the latter experiment, the model has processed each training example twice rather than once. When we attack the model trained for 2 epochs, we observe a significant increase in MIA vulnerability: the AUC is 0.744—higher both than this model when it has only completed 1 epoch of training (0.628, 1 of 2) and than the model trained for 1 epoch on the entire dataset (0.620, 1 of 1). This underscores that increasing training epochs, even on a smaller dataset to maintain Chinchilla optimality for overall training tokens, amplifies vulnerability to MIA, compared to training for fewer epochs on a larger dataset. However, we also observe that there is no significant uplift in TPR at small FPR between epochs 1 and 2 for the 2-epoch experiment. We also observe that the MIA at the second epoch is less successful than the one after 1 epoch for small FPR.

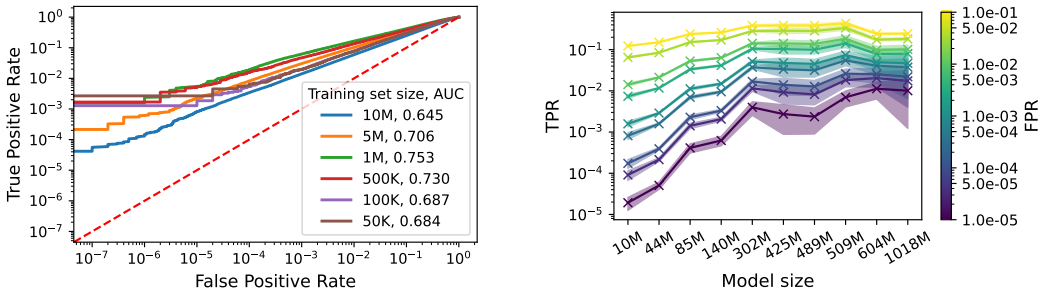
To investigate this further, we additionally perform experiments with the 140M architecture for various numbers of epochs. In Figure 3b, we show how the ROC curves and resulting AUC change over the course of training for 10 epochs. As expected, the AUC increases with more epochs, starting from 0.573 and reaching 0.797 at the end of the tenth epoch.[1] Interestingly, like Figure 3a, there again seems to be an FPR inflection point where TPR for later epochs is *smaller* than earlier epochs. In Appendix C, we also train the 140M model architecture on fewer than the ≈7 million Chinchilla-optimal examples, and (similar to Figure 3a) we observe that there is a more dramatic increase in MIA vulnerability. We show that attacking a 140M model trained on $2^{19}$ ≈524,000 examples exhibits both greater absolute MIA success and a faster relative increase in success in the first few training epochs.

### 4.2 Effects of scaling the training dataset size

We next run two sets of experiments to investigate the role of training-dataset size on MIA—beyond training on the Chinchilla-optimal number of tokens. We train 140M models on datasets ranging from 50K to 10M examples (again for a single epoch) and measure these models' susceptibility to LiRA. In Figure 4a, we show the ROC curves for the different models, which suggest that TPR@FPR is not necessarily positively correlated with decreasing the training dataset size. In other words, as we train models on smaller datasets, it is not always the case that TPR for a given FPR increases. Rather, AUC is highest for moderately sized datasets (around 1M examples, in this case with AUC of 0.753), and decreases for both very small and very large datasets (under 0.7, for both). Indeed, the capacity of the model also has an effect on susceptibility to successful strong MIA.

In Figure 4b, we train different model sizes with a fixed training set size of $2^{23} \approx 8.3$M examples—a number that is significantly larger than Chinchilla-optimal for several of our models (e.g., 10M, 44M). We plot the average and standard deviation of TPR rates, where we repeat this experiment 16 times using different random seeds, which has the effect of dictating the batch order. That is, for each model

[1] At the end of epoch 1, the AUC of 0.573 differs from the AUC of 0.678 we found in the experiments in Figure 2a, where the model is only trained for 1 epoch. We believe this is because of the substantially different learning rates between the two experimental setups.




(a) 140M model, trained on various dataset sizes for a single epoch.

(b) Various model sizes on a fixed-size dataset ($2^{23}$ samples) for a single epoch.

Figure 4: **Varying sizes of training data and models.** In (**a**), we train and attack 140M models on different-size datasets, ranging from 50K to 10M examples, and show MIA success does not monotonically increase with increasing dataset size. In (**b**), we train different architectures on a fixed dataset size, and plot how TPR varies at fixed FPR. Here, there is a monotonic increase in MIA success if we fix the training set size and increase the model size.

size, we train 16 sets of 128 reference models, and we also vary the target model over each experimental run. We include the associated AUC-ROC curves for each model size in Appendix C, which are consistent with Figure 4b in demonstrating MIA prediction variability. We observe a monotonic increase in TPR at different fixed FPRs as the model size increases. Notice, this is quite different from results in Figure 2b, where we scale the training set size with model size. As model capacity grows, vulnerability to MIA also grows if we keep the training set size constant. Further, we also note that there is significantly more variance in TPR for larger model sizes and at smaller fixed FPR.

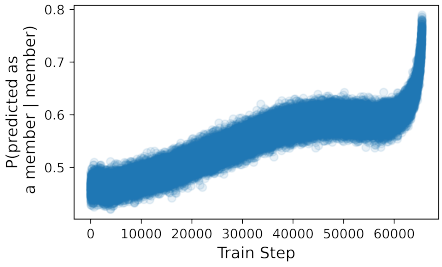
## 5 Analyzing sample vulnerability to membership inference

The instability in membership predictions that we observe above suggests a natural follow-on question: when does strong MIA succeed? More particularly, which samples are actually vulnerable to MIA, and (how) does this vulnerability vary during training? In this section, we approach these questions by digging deeper into our strong attacks on 140M model (trained with a Chinchilla-optimal training dataset size for a single epoch). We show how our large-scale experiments yield novel insights about the behavior of individual membership predictions (Section 5.1). Samples seen later in training tend to be more vulnerable; however, this trend is complicated by sample length, which also affects vulnerability. While sample length has previously been linked to extraction risk [5, 35], we observe no correlation between MIA and extraction, which suggests that the two are capturing different signals related to memorization (Section 5.2). Together, this analysis informs our third key takeaway: **the relationship between MIA vulnerability and related privacy metrics is not straightforward.**

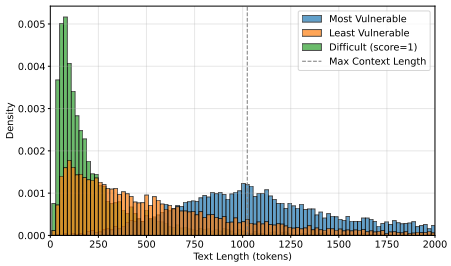
### 5.1 Identifying patterns in per-sample MIA vulnerability

We first investigate how sample MIA vulnerability evolves over the course of training. In Figure 5a, we provide a scatter plot that illustrates the per-sample true-positive probabilities by training step. That is, we plot how the probability of a training sample being correctly predicted as a member ($P(\text{predicted as a member}|\text{member})$) changes as model training progresses.

There is considerable variance in the underlying sample true-positive probabilities. At any particular training step, the true-positive probabilities over a batch of samples can vary by more than 15%, having a significant effect on overall attack success. We explore this high degree of instability further in Appendix I, where we plot the mean and standard deviation of per-example true positive probabilities. The mean $P(\text{predicted as a member}|\text{member})$ for many samples is close to $0.5$; their predictions are close to arbitrary, meaning that they are challenging for MIA (see Figure 5b). The associated standard deviations are also quite large (on average, $0.143$). As a result, the sample membership predictions can easily flip—since they can change to be above or below $0.5$—depending on the random seed (dictating batch order) and the specific target model.




(a) Evolution of per-sample MIA success for member samples over training steps.




(b) Token-length distributions for samples that are least and most vulnerable to MIA, and samples for which MIA has difficulty predicting membership.

Figure 5: **Sample vulnerability to MIA.** We show different aspects of per-sample vulnerability for a 140M model. (**a**) We plot the evolution of per-sample vulnerability throughout training. We show the probability of individual training samples being correctly identified as members at different stages of the training process—the per-example true positive probabilities, $P(\text{predicted as a member}|\text{member})$. (**b**) We plot distributions over sample lengths, according to MIA vulnerability for the 1,000 samples with smallest $P(\text{predicted as a member}|\text{member})$ (least vulnerable), largest $P(\text{predicted as a member}|\text{member})$ (most vulnerable), and $P(\text{predicted as a member}|\text{member})$ closest to 0.5 (difficult samples for MIA; this is equivalent to a LiRA score of 1).

Nevertheless, the density of the points shifts upward toward the end of training (around step 60,000). Unsurprisingly, samples in batches that are processed in later epochs tend to be more vulnerable, as indicated by their higher probability of being correctly identified as members. This result highlights that the recency of exposure influences a sample's vulnerability to membership inference. **Put differently, samples introduced earlier in training are more likely to be "forgotten" [4]: they are less vulnerable to MIA.**

While this appears to be the dominant trend in this setting, the details are a bit more complicated. In Figure 5b, we investigate if there are other patterns in sample vulnerability. We plot the distribution over training samples according to their length in tokens, and partition this distribution according to their vulnerability. We consider samples that are members, but which LiRA predicts confidently and incorrectly to be non-members, to be the least vulnerable. (LiRA being confident and wrong reduces the probability of a sample being correctly identified as a member to below 0.5.) In contrast, samples that LiRA correctly and confidently predicts to be members are the most vulnerable. (This brings the probability to above 0.5.) We also highlight samples where LiRA struggles to determine if a member is a member or non-member. (These samples have a score of 1; as noted above, the probability that these samples are predicted to be members is 0.5, indicating their predictions are effectively arbitrary.)

In summary, Figure 5b suggests that it is not just the case that samples seen later in training are more vulnerable (Figure 5a); it is also often the case that vulnerable sequences tend to be longer. (See also Figure 19 in the Appendix, which illustrates similar results for samples that have a higher proportion of `<unk>` tokens and higher average $\text{TF}-\text{IDF}$ scores.) This result is consistent with those in Carlini et al. [5], which show that sequences that are vulnerable to extraction tend to be greater in length.

### 5.2 Comparing MIA vulnerability and extraction

Results such as those in Figure 5b, which show alignment between MIA vulnerability and training-data extraction attacks [2], are consistent with prior work on memorization in machine learning. In general, it is assumed that a successful membership inference attack and successful extraction of training data imply that some degree of memorization has occurred for the attacked ML model. For MIA, this is assumed because the success of such attacks hinges on the model's tendency to behave differently for data it has seen during training (members) compared to unseen data (non-members); this differential behaviour is frequently ascribed to the model having memorized certain aspects of the training data.

As a final experiment, we therefore investigate whether, in this setting, the samples that are vulnerable to our strong MIAs are also vulnerable to extraction attacks. In Figure 6, we compute extraction metrics for the 1,000 samples identified as most vulnerable to MIA in the 140M model, using the first 50 tokens of a sample as a prefix and measuring the $\log$-probability of the next 50 tokens

(a variant of **discoverable extraction**, introduced in Carlini et al. [2]. Specifically, we use a sample's negative log-probability as a proxy for computing a modified version of discoverable extraction—the **$(n, p)$-discoverable extraction** metric introduced by Hayes et al. [17]. Traditional discoverable extraction evaluates attack success as a binary outcome (success or failure); in contrast, $(n, p)$-discoverable extraction quantifies the number of attempts $n$ an adversary needs to extract a particular training sample at least once with probability $p$ (ranging from 0 to 1), given a specific prompt, model, and decoding scheme. Generally, a smaller negative log-probability implies that a sample is easier to extract. Hayes et al. [17] show that traditional discoverable extraction underestimates what is actually extractable (given more than one attempt); we therefore choose this metric for extraction, as we expect it to provide more reliable signal for memorization.

In our experiments, with respect to MIA score after 1 epoch of training, LiRA is able to identify training members with better-than-random AUC. Out of 1,000 samples with the highest LiRA score, 713 of these are indeed training members. Despite obtaining useful MIA signal, we are unable to extract *any* of the correctly identified member samples with meaningful probability! In Figure 6, the *smallest* negative log-probability of a member sample—i.e., the member sample most vulnerable to extraction—is approximately 5. To understand this in terms of $(n, p)$-discoverable extraction, an adversary would need to attempt extraction over $n = 230,000$ times to extract the sample with confidence $p > 90\%$! Altogether, while much prior work draws a direct connection between MIA vulnerability and extraction risk [e.g., 2], our results suggest a more nuanced story. **Our results suggest that the success of a strong MIA on a given member sample (i.e., an MIA true positive) does not necessarily imply that the LLM is more likely to generate that sample than would be expected under the data distribution [17].**

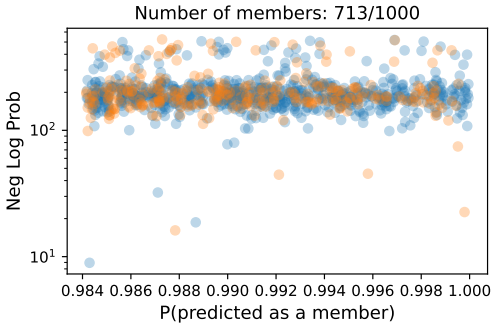



Figure 6: We plot the 1,000 samples predicted most strongly as a member of training by LiRA for the Chinchilla-optimal 140M model trained for 1 epoch (which contains approx. 7M training samples). We plot samples that are members (713) of training in blue and non-members in orange (287). We plot the negative log-probability of (the first 100 tokens of) each of these samples from which we can derive extraction rates according to $(n, p)$-discoverable extraction metric introduced by Hayes et al. [17].

## 6 Conclusion and future work

In this paper, we present the first large-scale study of strong membership inference attacks on large language models. To enable strong attacks that calibrate their membership predictions using reference models, we train thousands of GPT-2-like models (ranging from 10M–1B parameters) on enormous training datasets sampled from C4—datasets that are up to three orders of magnitude larger than those used in prior work. Through dozens of experiments, we aim to answer an urgent open question in ML privacy research: *are the fidelity issues of weaker attacks due to omitting reference models, or do they point to a deeper, more fundamental challenge with applying membership inference to large language models?* We uncover three novel groups of findings: while (1) strong MIAs can succeed on pre-trained LLMs (Section 3), (2) their success is limited (i.e., AUC $< 0.7$) for LLMs trained using practical settings (Section 4), and (3) the relationship between MIA vulnerability and related privacy metrics—namely, extraction—is not straightforward (Section 5).

Further, as the first work to perform large-scale strong MIAs on pre-trained LLMs, we are also the first to clarify the extent of actual privacy risk MIAs pose in this setting. By evaluating the effectiveness of strong attacks, we are able to establish an upper bound on the accuracy that weaker, more feasible attacks can achieve. More generally, we also identify the conditions under which MIAs are effective on pre-trained LLMs. Together, our findings can guide others in more fruitful research directions to develop novel attacks and, hopefully, more effective defenses. They also suggest that, in the future (and with additional compute cost), it may be possible and worthwhile to derive scaling laws for MIAs.

## References


[1] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[2] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[3] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[4] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276, 2022.

[5] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying Memorization Across Neural Language Models. In *International Conference on Learning Representations*, 2023.

[6] Hongyan Chang, Ali Shahin Shamsabadi, Kleomenis Katevas, Hamed Haddadi, and Reza Shokri. Context-aware membership inference attacks against pre-trained large language models. *arXiv preprint arXiv:2409.13745*, 2024.

[7] A. Feder Cooper and James Grimmelmann. The Files are in the Computer: Copyright, Memorization, and Generative AI. *arXiv preprint arXiv:2404.12590*, 2024.

[8] A. Feder Cooper, Katherine Lee, James Grimmelmann, Daphne Ippolito, Christopher Callison-Burch, Christopher A. Choquette-Choo, Niloofar Mireshghallah, Miles Brundage, David Mimno, Madiha Zahrah Choksi, Jack M. Balkin, Nicholas Carlini, Christopher De Sa, Jonathan Frankle, Deep Ganguli, Bryant Gipson, Andres Guadamuz, Swee Leng Harris, Abigail Z. Jacobs, Elizabeth Joh, Gautam Kamath, Mark Lemley, Cass Matthews, Christine McLeavey, Corynne McSherry, Milad Nasr, Paul Ohm, Adam Roberts, Tom Rubin, Pamela Samuelson, Ludwig Schubert, Kristen Vaccaro, Luis Villa, Felix Wu, and Elana Zeide. Report of the 1st Workshop on Generative AI and Law. *arXiv preprint arXiv:2311.06477*, 2023.

[9] A. Feder Cooper, Christopher A. Choquette-Choo, Miranda Bogen, Matthew Jagielski, Katja Filippova, Ken Ziyu Liu, Alexandra Chouldechova, Jamie Hayes, Yangsibo Huang, Niloofar Mireshghallah, Ilia Shumailov, Eleni Triantafillou, Peter Kairouz, Nicole Mitchell, Percy Liang, Daniel E. Ho, Yejin Choi, Sanmi Koyejo, Fernando Delgado, James Grimmelmann, Vitaly Shmatikov, Christopher De Sa, Solon Barocas, Amy Cyphert, Mark Lemley, danah boyd, Jennifer Wortman Vaughan, Miles Brundage, David Bau, Seth Neel, Abigail Z. Jacobs, Andreas Terzis, Hanna Wallach, Nicolas Papernot, and Katherine Lee. Machine Unlearning Doesn't Do What You Think: Lessons for Generative AI Policy, Research, and Practice. *arXiv preprint arXiv:2412.06966*, 2024.

[10] A. Feder Cooper, Aaron Gokaslan, Amy B. Cyphert, Christopher De Sa, Mark A. Lemley, Daniel E. Ho, and Percy Liang. Extracting memorized pieces of (copyrighted) books from open-weight language models. *arXiv preprint arXiv:2505.12546*, 2025.

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

[12] Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8698–8711, 2024.

[13] Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? In *First Conference on Language Modeling*, 2024.

[14] André V Duarte, Xuandong Zhao, Arlindo L Oliveira, and Lei Li. De-cop: detecting copyrighted content in language models training data. In *Proceedings of the 41st International Conference on Machine Learning*, pages 11940–11956, 2024.

[15] Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. Membership inference attacks against fine-tuned large language models via self-prompt calibration. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[16] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[17] Jamie Hayes, Marika Swanberg, Harsh Chaudhari, Itay Yona, Ilia Shumailov, Milad Nasr, Christopher A. Choquette-Choo, Katherine Lee, and A. Feder Cooper. Measuring memorization in language models via probabilistic extraction. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *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)*, pages 9266–9291, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.469/.

[18] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training Compute-Optimal Large Language Models, 2022. URL https://arxiv.org/abs/2203.15556.

[19] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

[20] Nikhil Kandpal, Krishna Pillutla, Alina Oprea, Peter Kairouz, Christopher Choquette-Choo, and Zheng Xu. User inference attacks on large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18238–18265, 2024.

[21] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021.

[22] Katherine Lee, A. Feder Cooper, and James Grimmelmann. Talkin' 'Bout AI Generation: Copyright and the Generative-AI Supply Chain. *arXiv preprint arXiv:2309.08133*, 2023.

[23] Peter J. Liu, Roman Novak, Jaehoon Lee, Mitchell Wortsman, Lechao Xiao, Katie Everett, Alexander A. Alemi, Mark Kurzeja, Pierre Marcenac, Izzeddin Gur, Simon Kornblith, Kelvin Xu, Gamaleldin Elsayed, Ian Fischer, Jeffrey Pennington, Ben Adlam, and Jascha-Sohl Dickstein. NanoDO: A minimal Transformer decoder-only language model implementation in JAX, 2024. URL http://github.com/google-deepmind/nanodo. Version 0.1.0.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[25] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE, 2023.

[26] Pratyush Maini and Hritik Bansal. Peeking behind closed doors: Risks of llm evaluation by private data curators. In *The Fourth Blogpost Track at ICLR 2025*, 2025.

[27] Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. Llm dataset inference: Did you train on my dataset? *Advances in Neural Information Processing Systems*, 37:124069–124092, 2024.

[28] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343, 2023.

[29] Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. Did the neurons read your book? document-level membership inference for large language models. In *Proceedings of the 33rd USENIX Conference on Security Symposium*, pages 2369–2385, 2024.

[30] Matthieu Meeus, Igor Shilov, Manuel Faysse, and Yves-Alexandre de Montjoye. Copyright traps for large language models. In *Forty-first International Conference on Machine Learning*, 2024.

[31] Matthieu Meeus, Igor Shilov, Shubham Jain, Manuel Faysse, Marek Rei, and Yves-Alexandre de Montjoye. Sok: Membership inference attacks on llms are rushing nowhere (and how to fix it). *arXiv preprint arXiv:2406.17975*, 2024.

[32] Matthieu Meeus, Lukas Wutschitz, Santiago Zanella-Béguelin, Shruti Tople, and Reza Shokri. The canary's echo: Auditing privacy risks of llm-generated synthetic text. *arXiv preprint arXiv:2502.14921*, 2025.

[33] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8332–8347, 2022.

[34] Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David K Evans, and Taylor Berg-Kirkpatrick. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1816–1826, 2022.

[35] Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable Extraction of Training Data from (Production) Language Models. *arXiv preprint arXiv:2311.17035*, 2023.

[36] Yonatan Oren, Nicole Meister, Niladri S Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations*, 2023.

[37] Ashwinee Panda, Xinyu Tang, Christopher A. Choquette-Choo, Milad Nasr, and Prateek Mittal. Privacy auditing of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=60Vd7QOXlM.

[38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[39] Lorenzo Rossi, Bartłomiej Marek, Vincent Hanke, Xun Wang, Michael Backes, Adam Dziedzic, and Franziska Boenisch. Auditing empirical privacy protection of private llm adaptations. In *Neurips Safe Generative AI Workshop 2024*, 2024.

[40] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567. PMLR, 2019.

[41] Nikhil Sardana, Jacob Portes, Sasha Doubov, and Jonathan Frankle. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. *arXiv preprint arXiv:2401.00448*, 2023.

[42] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

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

[44] Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206, 2019.

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[46] Cheng Wang, Yiwei Wang, Bryan Hooi, Yujun Cai, Nanyun Peng, and Kai-Wei Chang. Conrecall: Detecting pre-training data in llms via contrastive decoding. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1013–1026, 2025.

[47] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In *International Conference on Learning Representations*, 2022.

[48] Johnny Wei, Ryan Wang, and Robin Jia. Proving membership in llm pretraining data via data watermarks. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13306–13320, 2024.

[49] Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Gong, and Bhuwan Dhingra. Recall: Membership inference via relative conditional log-likelihoods. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8671–8689, 2024.

[50] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106, 2022.

[51] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[52] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *Forty-first International Conference on Machine Learning*, 2024.

[53] Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. Membership Inference Attacks Cannot Prove that a Model Was Trained On Your Data, 2025. URL https://arxiv.org/abs/2409.19798.

[54] Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. Min-k%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZGkfoufDaU.

# A Comparing membership inference attacks and signals

At the beginning of this project, we considered two candidates for strong membership inference attacks to use in our experiments: the Likelihood Ratio Attack (LiRA) [3] and the Robust Membership Inference Attack (RMIA) [52]. Both attacks involve training reference models (Section 2) that enable the computation of likelihood ratios (which result in stronger attacks), though they differ in important ways. LiRA [3] estimates membership by comparing the loss of an example $\boldsymbol{x}$ in a target model to empirical loss distributions from reference models trained with and without $\boldsymbol{x}$. In contrast, RMIA [52] performs and aggregates statistical pairwise likelihood ratio tests between $\boldsymbol{x}$ and population samples $\boldsymbol{z}$, using both reference models and $\boldsymbol{z}$ to estimate how the inclusion of $\boldsymbol{x}$ versus $\boldsymbol{z}$ affects the probability of generating the observed model $\theta$.

By leveraging signal from both models and population samples, Zarifzadeh et al. [52] observe that RMIA can outperform LiRA using fewer reference models. However, no prior work has compared these methods in the pre-trained LLM setting and with large numbers of reference models, leaving open the question of which attack fares better under these conditions.

In this appendix, we investigate this question for the first time, and our results clearly indicate that LiRA outperforms RMIA for a large number of reference models in the online setting. However, RMIA can outperform LiRA if the population dataset is large enough and the attack is performed for certain small numbers of reference models. This pattern is not completely straightforward: LiRA seems to perform better with 1 or 2 reference models, while RMIA performs better with 4–16, and then LiRA once again outperforms RMIA for $> 16$ reference models.

Overall, though, attacks with larger numbers of reference models perform better, which means that for our setting LiRA is the best choice for our experiments. Our aim is to test the strongest attacks possible, as this is useful for an upper bound on attack performance. For those with smaller compute budgets that wish to still run strong attacks on $\approx 16$ reference models, RMIA may be a better choice.

We train 140M-parameter models on 7M examples, which equates to approximately 2.8B training tokens (i.e., what is optimal for this model size, according to Chinchilla scaling laws [18] with an over-training multiplier of 20). First, we provide more details on the strong attacks we study (Appendix A.1). Second, we show how different choices of inference signal impact attack performance (Appendix A.2), which provide more detail about the choices we make in our overall experimental setup (which we introduce in Section 3). Finally, we show our full results that compare the performance of LiRA and RMIA using different numbers of reference models (Appendix A.3).

## A.1 More background on membership inference attacks

**Formalization.** Given a target model with parameters $\theta$ and an input $\boldsymbol{x}$, an MIA method $\mu$ aims to determine a **membership score** $\Lambda_\mu(\boldsymbol{x})$ capturing meaningful information whether $\boldsymbol{x}$ was used to train target model $\theta$. Using ground truth information about the membership of target records $\boldsymbol{x}$, the performance for method $\mu$ is computed using the membership score and a threshold-agnostic metric such as $\mathrm{ROC\ AUC}$ or $\mathrm{TPR}$ at low $\mathrm{FPR}$.

Let $f_\theta(\boldsymbol{x})$ denote a scalar statistic computed from the target model on $\boldsymbol{x}$, e.g., a loss or confidence score. In its simplest form, $f_\theta(\boldsymbol{x})$ can be used directly as the membership score, based on the assumption that training examples yield lower loss than non-members, or $\Lambda_{\text{Loss}}(\boldsymbol{x}) = f_\theta(\boldsymbol{x})$ [51]. No reference models (Section 2) are used in this baseline approach.

Different methods have been proposed that use reference models to improve upon this membership signal. For such strong attacks, we denote reference models $\phi \in \Phi$ as a set of models trained on data from a similar distribution as the data used to train the target model $\theta$. Typically, a set of reference models contains an equal amount of models trained on data including target example $\boldsymbol{x}$ ($\Phi_{\text{IN}}$) and excluding $\boldsymbol{x}$ ($\Phi_{\text{OUT}}$). This is the setup we adopt in this work, which corresponds to the **online** attack in Carlini et al. [3], Zarifzadeh et al. [52]. In contrast, the **offline** attack assumes only access to $\Phi_{\text{OUT}}$.

**LiRA** [3] leverages the target record loss computed on the IN reference models $\Phi_{\text{IN}}$ and the OUT models $\Phi_{\text{OUT}}$ in order to compute a membership score. Specifically, for each model $\phi \in \Phi_{\text{IN}} \cup \Phi_{\text{OUT}}$, one computes $f_\phi(\boldsymbol{x})$. Let $p_{\text{IN}}$ and $p_{\text{OUT}}$ be the empirical distributions of these values from $\Phi_{\text{IN}}$ and

$\Phi_{\text{OUT}}$, respectively. LiRA defines the membership signal as the likelihood ratio

$$\Lambda_{\text{LiRA}}(\boldsymbol{x}) = \frac{p_{\text{IN}}(f_\theta(\boldsymbol{x}))}{p_{\text{OUT}}(f_\theta(\boldsymbol{x}))}.$$

In practice, $p_{\text{IN}}$ and $p_{\text{OUT}}$ are modeled as univariate Gaussians fit to the empirical values of $f_\phi(\boldsymbol{x})$ from the respective datasets.

**RMIA** [52] also compares the target model's output on $\boldsymbol{x}$ to that of a set of reference models $\Phi$; however, it uses a different likelihood ratio test:

$$\alpha(\boldsymbol{x}) = \frac{f_\theta(\boldsymbol{x})}{\mathbb{E}_{\phi\in\Phi}[f_\phi(\boldsymbol{x})]}.$$

The expected value in the denominator is approximated empirically by computing over the reference models that one actually trains. To improve robustness, RMIA further contextualizes this score relative to a reference population $\mathbb{Z}$. For each $\boldsymbol{z} \in \mathbb{Z}$:

$$\alpha(\boldsymbol{z}) = \frac{f_\theta(z)}{\mathbb{E}_{\phi\in\Phi}[f_\phi(z)]}, \quad L(x,z) = \frac{\alpha(x)}{\alpha(z)}.$$

The final membership signal is defined as the fraction of population points $\boldsymbol{z}$ for which this ratio exceeds a threshold $\gamma$:

$$\Lambda_{\text{RMIA}}(\boldsymbol{x}) = \frac{1}{|\mathbb{Z}|}\sum_{\boldsymbol{z}\in\mathbb{Z}} \mathbb{1}\left[\frac{\alpha(\boldsymbol{x})}{\alpha(\boldsymbol{z})} \geq \gamma\right].$$

### A.2 Different signal observations

In our initial experiments in Section 3 for comparing which strong attack to use—LiRA [3] or RMIA [52]—we also investigated the efficacy of different membership inference signals. We compare using the model loss and model logits (averaged over the entire sequence), for example, in Figure 7, looking at the ROC curve for LiRA and a 140M sized model trained on $\approx 7M$ examples.

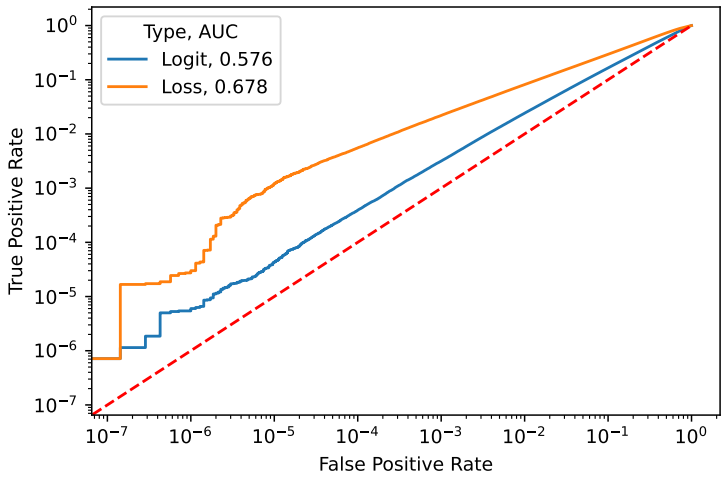


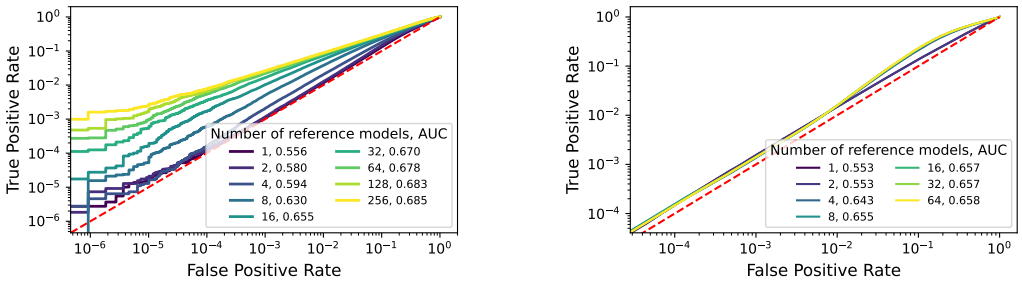

Figure 7: **Influence of signal type on MIA Performance.** We plot ROC curves that compare the efficacy of using model logits (AUC 0.576) versus model loss (AUC 0.678) as signals for membership inference with LiRA. The results indicate that, in this setting, the loss provides a stronger signal for distinguishing members from non-members.

The plot shows the True Positive Rate (TPR) against the False Positive Rate (FPR) on a log−log scale, with one curve each for logit and loss signals. The logit curve has an AUC of 0.576, while the loss curve has a higher AUC of 0.678. This indicates that using the loss as a signal results in a more effective attack compared to using logits in this specific experimental setup. In general, we opt to use loss as our membership inference signal metric, as we observe it to be more effective.

### A.3 MIA attack performance for different number of reference models

Figure 8 compares LiRA and RMIA, showing ROC curves and AUC for different numbers of reference models. Figure 9 provides an alternate view of the same results, plotting AUC for both attacks as a function of reference models. LiRA's performance generally dominates RMIA's. LiRA continues to improve as we increase the number of reference models, while RMIA's effectiveness plateaus. For example, with 4–16 reference models, RMIA surpasses the performance of LiRA (it essentially matches LiRA using 16 reference models). With 4 reference models, LiRA has an AUC of 0.594 (which under-performs RMIA's corresponding AUC of 0.643), but LiRA's AUC increases to 0.678 with 64 reference models (which outperforms RMIA's AUC of 0.658). Also note that RMIA exhibits a distinct diagonal pattern at low FPR.[2]

While both attacks clearly beat the random baseline, neither is remarkably successful in this setting: regardless of the number of reference models, neither achieves an AUC of 0.7.




(a) **LiRA** (b) **RMIA**

Figure 8: **Comparing LiRA and RMIA.** We train 140M-parameter reference models on a dataset size of 7M. ROC curves illustrate the effectiveness of (**a**) LiRA [3] and (**b**) RMIA [52] for different numbers of reference models. As we increase the number of reference models, LiRA's performance (measured with AUC) surpasses RMIA's.

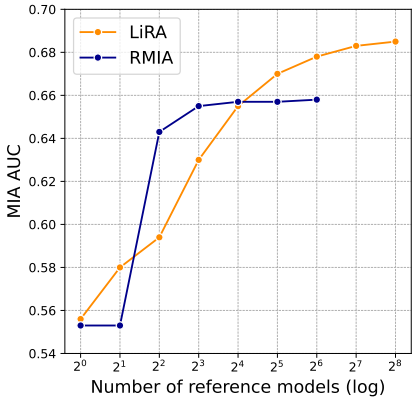



Figure 9: **Comparing performance of LiRA and RMIA with an increasing number of reference models (c.f. Figure 8).** We plot MIA ROC AUC achieved by both attack methodologies for an increasing number of reference models. As the number of reference models increases, LiRA's performance continues to improve, while RMIA's gains saturate, making LiRA the overall stronger attack.

**Understanding RMIA.** We now further investigate RMIA, decoupling its different components. First, we consider the simplest form of RMIA (*simple*), eliminating its dependence on a $\mathbb{Z}$ population

[2]While RMIA aims to be a strong attack that works well in low-compute settings, we find that a large population $\mathbb{Z}$ is necessary to obtain meaningful TPR at very low FPR thresholds. That is, for a minimally acceptable $\text{FPR}_{\text{min}}$, RMIA requires a population size $|\mathbb{Z}|$ that is $\frac{1}{\text{FPR}_{\text{min}}}$. In practice, this is quite expensive, as RMIA's membership score is computed via pairwise comparisons with these $|\mathbb{Z}|$ reference points (i.e., there are $\mathcal{O}(|\mathbb{Z}|)$ pairwise likelihood ratio tests for target record $\boldsymbol{x}$, see Appendix A.1). In these initial experiments we only used $|\mathbb{Z}| = 10{,}000$ examples. We measure performance of RMIA on larger population sizes in Section A.3.

and using $\alpha(\boldsymbol{x})$ directly as membership signal. We also instantiate LiRA and RMIA with a reference population of size $|\mathbb{Z}| = 10,000$ and $\gamma = 1$.

Figure 10 shows the ROC curves for all three MIAs attacking one target model with 10M parameters trained for 1 epoch on a training set size of $2^{19}$. We use 128 reference models and consider $2 \times 2^{19} = 2^{20}$ target records $\boldsymbol{x}$ with balanced membership to analyse MIA. We find all three attacks to reach similar ROC AUC values.

We also gauge MIA performance by evaluating the TPR at low FPR. To understand the values RMIA reaches for TPR at low FPR, an important subtlety arises from the entropy of the score distribution. Attacks that produce very coarse membership scores inherently limit achievable TPR at very low FPR. For example, as RMIA compares $\alpha(\boldsymbol{x})$ to $\alpha(\boldsymbol{z})$ for all $\boldsymbol{z} \in \mathbb{Z}$ to compute its membership score $\Lambda_{\text{RMIA}}(\boldsymbol{x})$, there are maximally $|\mathbb{Z}|$ unique values $\Lambda_{\text{RMIA}}(\boldsymbol{x})$ can take for all $\boldsymbol{x}$. This limits the score's entropy and the possibility of achieving a meaningful TPR at very low FPR. This explains the diagonal pattern for RMIA in Figure 10, where $|\mathbb{Z}| = 10,000$. By contrast, both LiRA and RMIA (simple) provide a membership score not limited in entropy, leading to more meaningful values for TPR at lower FPR.

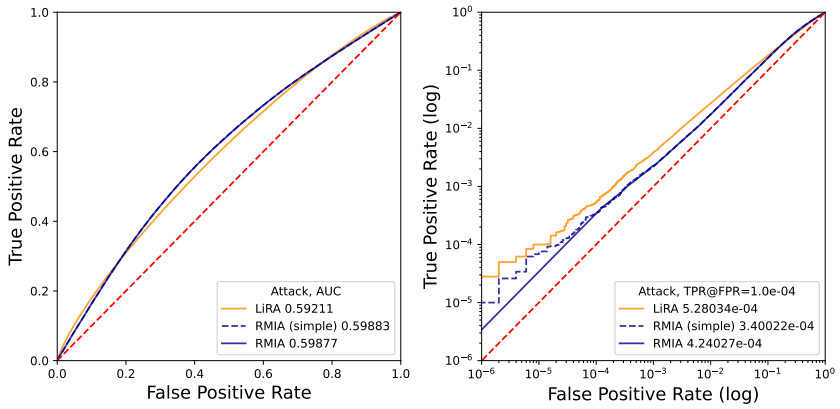



Figure 10: Comparing performance of LiRA, RMIA (simple) and RMIA on a 10M parameter model trained for 1 epoch with a training set size of $2^{19}$.

We next test further increasing the size of the population $\mathbb{Z}$ when computing RMIA. For the same setup, Figure 11 shows how MIA performance varies with the size of $\mathbb{Z}$. We observe very similar values for RMIA (simple) and RMIA AUC for all sizes of $\mathbb{Z}$ that we test. When examining TPR at low FPR, we find that increasing $|\mathbb{Z}|$ improves the MIA performance at low FPR. Indeed, the increased entropy in $\Lambda_{\text{RMIA}}(\boldsymbol{x})$ now allows the attack to reach meaningful values of TPR for FPR as low as $10^{-6}$. Notably, for all values of $|\mathbb{Z}|$ we consider, LiRA still outperforms RMIA at low FPR, while the $|\mathbb{Z}|$ likelihood comparisons in RMIA for every target record $\boldsymbol{x}$ also incur additional computational cost.

Finally, we evaluate RMIA under varying thresholds $\gamma$. As $\gamma$ increases, it becomes less likely that $\alpha(\boldsymbol{x})$ significantly exceeds $\alpha(\boldsymbol{z})$ for many $z \in \mathbb{Z}$. Figure 12 shows, again for the same setup, how RMIA performs for varying values of $\gamma$, considering both $|\mathbb{Z}| = 10,000$ (12a) and $|\mathbb{Z}| = 300,000$ (12b). While the MIA AUC remains relatively stable as $\gamma$ increases, the TPR at low FPR varies. For $|\mathbb{Z}| = 10,000$, the TPR at FPR $= 10^{-4}$ decreases for increasing value of $\gamma$, reaching 0 for $\gamma \geq 1.1$. This is due to the reduced granularity of RMIA's membership score: for larger $\gamma$, fewer $z$ satisfy $\alpha(x)/\alpha(z) \geq \gamma$, constraining the entropy of the RMIA score, making it harder to reach meaningful values of TPR at low FPR. A larger reference population ($|\mathbb{Z}| = 300{,}000$) mitigates this issue, allowing meaningful TPR even at low FPR for similar $\gamma$ values.

Taken together, considering multiple sizes of the reference population $|\mathbb{Z}|$ and values of $\gamma$, we find LiRA to outperform RMIA when a sufficient amount of reference models is available, especially in the low-FPR regime. We therefore adopt LiRA as the primary attack throughout this work.

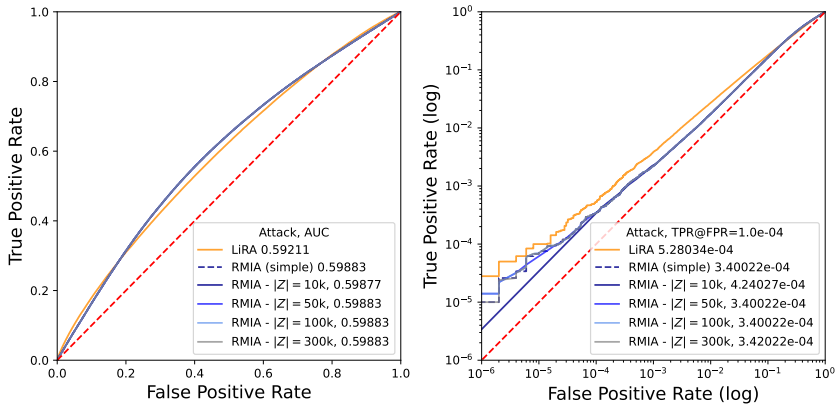



Figure 11: Performance of RMIA for increasing size of the population $\mathbb{Z}$ on a 10M parameter model trained for 1 epoch with a training set size of $2^{19}$.

### A.4 MIA performance in the offline setting

As stated in Section A.1, the literature distinguishes between an *online* and *offline* setting for reference-model based MIAs [3, 52]. In the online setting, the attacker has access to reference models trained on data including ($\Phi_{\text{IN}}$) and excluding ($\Phi_{\text{OUT}}$) the target example $\boldsymbol{x}$. In the offline setting, the attacker only has access to models not trained on $\boldsymbol{x}$, thus to $\Phi_{\text{OUT}}$. Throughout this work, we consider the strongest attacker and thus report all results in the online setting.

For completion, we here also instantiate MIAs in the offline setting in the same experimental setup as considered above. We adopt the offline versions for both LiRA and RMIA as originally proposed [3, 52]. For LiRA, without $\Phi_{\text{IN}}$ we are not able to approximate the probability $p_{\text{IN}}(f_\theta(\boldsymbol{x}))$, and thus just consider the one-sided hypothesis test as membership signal instead of the likelihood ratio:

$$\Lambda_{\text{LiRA,offline}}(\boldsymbol{x}) = 1 - p_{\text{OUT}}(f_\theta(\boldsymbol{x})).$$

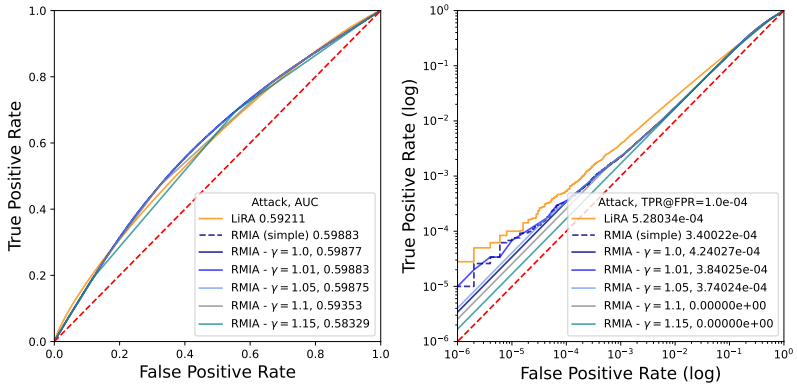
For RMIA, we now compute the denominator in $\alpha(\boldsymbol{x})$ by taking the expectation over the reference models that are available to the attacker, or:

$$\alpha_{\text{offline}}(\boldsymbol{x}) = \frac{f_\theta(\boldsymbol{x})}{\mathbb{E}_{\phi\in\Phi_{\text{OUT}}}[f_\phi(\boldsymbol{x})]}.$$

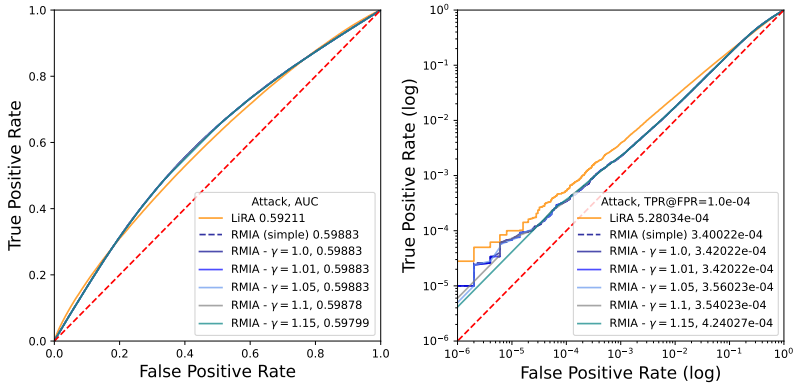
Note that Zarifzadeh et al. [52] proposes to further adjust the denominator by using a variable $a$ (their Appendix B.2.2) to better approximate the $\mathbb{E}_{\phi\in\Phi}[f_\phi(\boldsymbol{x})]$ while only using $\Phi_{\text{OUT}}$ in the offline setting. We here set $a = 1$ and just compute the empirical mean across all reference models in $\Phi_{\text{OUT}}$ to approximate the expectation in the denominator. We then compute $\alpha_{\text{offline}}(\boldsymbol{z})$ and use as membership inference signal:

$$\Lambda_{\text{RMIA,offline}}(\boldsymbol{x}) = \frac{1}{|\mathbb{Z}|}\sum_{\boldsymbol{z}\in\mathbb{Z}} \mathbb{1}\left[\frac{\alpha_{\text{offline}}(\boldsymbol{x})}{\alpha_{\text{offline}}(\boldsymbol{z})} \geq \gamma\right].$$

Figure 13 compares the MIA performance between the online and offline setting, for LiRA, RMIA (simple) which does not use the reference population $\mathbb{Z}$ and RMIA with $\gamma = 1$ and $|\mathbb{Z}| = 300,000$. We again consider the 10M parameter model trained for 1 epoch with a training set size of $2^{19}$ using 128 reference models for the online setting and 64 in the offline setting (on average per target example). We find that, in this configuration and with this number of reference models, offline RMIA outperforms offline LiRA, in terms of both ROC AUC and TPR at low FPR. This suggests that RMIA's offline signal more accurately captures membership information compared to the one-sided hypothesis test used in offline LiRA. In the online setting, in contrast, LiRA and RMIA achieve similar ROC AUC values, with LiRA performing better than RMIA in the low-FPR regime.

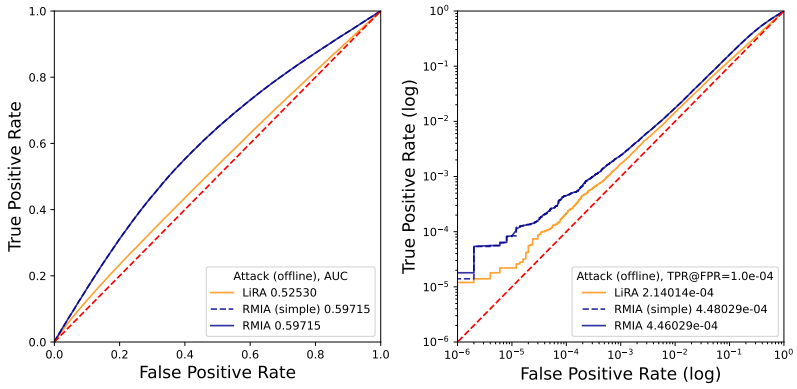



(a) $|\mathbb{Z}| = 10,000$

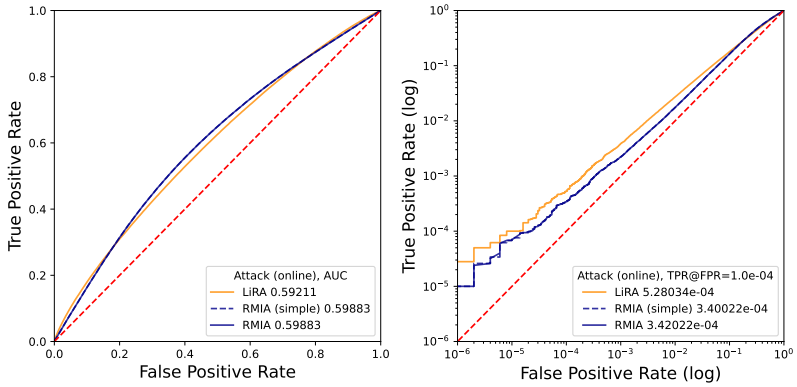



(b) $|\mathbb{Z}| = 300,000$

Figure 12: Performance of RMIA for increasing value of $\gamma$ on a 10M parameter model trained for 1 epoch with a training set size of $2^{19}$.




(a) Offline

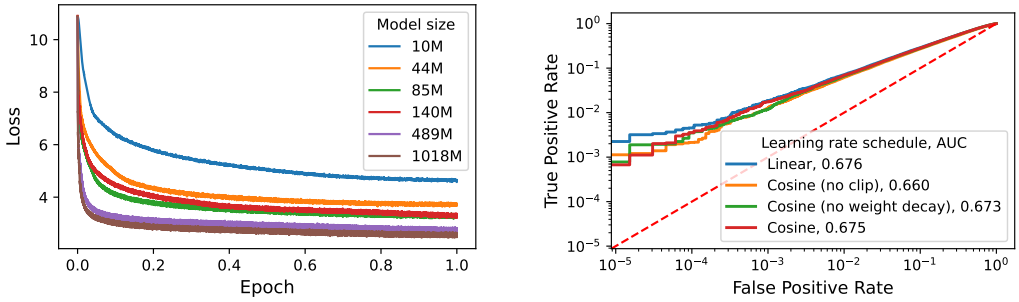



(b) Online

Figure 13: MIA performance in the offline and online setting, on a 10M parameter model trained for 1 epoch with a training set size of $2^{19}$, considering 128 reference models in the online setting and only the corresponding models $\Phi_{\text{OUT}}$ in the offline setting (on average 64 per sample).

## B More experiments on Chinchilla-optimal models

We provide additional details on our experiments involving LiRA attacks on Chinchilla-optimal [18] models of different sizes in Section 3.2: 10M, 44M, 85M, 140M, 489M, and 1018M parameters. We provide concrete training hyperparameters in Section E.




(a) Validation loss over time (1 epoch)

(b) LiRA ROC for different learning rate schedules

Figure 14: **Investigating training dynamics hyperparameters.** As a sanity check, we plot the loss (**a**) throughout the single training epoch that we run for our experiments involving Chinchilla-optimal trained models of various sizes. We also test (**b**) the effect of different learning rate schedules on LiRA's attack success for 140M models using 128 reference models.

In Figure 14a, we show the decrease in validation loss over a single epoch for these models. The $x$-axis represents the fraction of the training epoch completed (from 0.0 to 1.0), and the $y$-axis shows the corresponding loss. As expected, all models exhibit a characteristic decrease in loss as training progresses. Larger models (namely, 489M and 1018M) demonstrate faster convergence to lower loss values, reflecting their increased capacity to fit the training data. They also maintain a lower loss throughout the epoch compared to smaller models (10M–140M).

**Investigating the role of learning rate schedule.** In the Chinchilla-optimal setting, we also investigate the role of hyperparameters on MIA performance. In Figure 14b, we present ROC curves that compare the MIA vulnerability (with LiRA) of 140M-parameter models (trained on approximately 7M records, with 128 reference models), where we vary the learning rate schedule: Linear (AUC 0.676), Cosine (no global norm clipping, AUC 0.660), Cosine (no weight decay, AUC 0.673), and standard Cosine (AUC 0.675). As with all of our ROC plots, the TPR is plotted against the FPR on a $\log-\log$ scale. The AUC values for each curve are relatively close. This indicates that, while there are some minor differences in attack performance, the choice of learning rate schedule among those tested does not lead to drastically different MIA outcomes.

## C Additional experiments on LiRA limitations

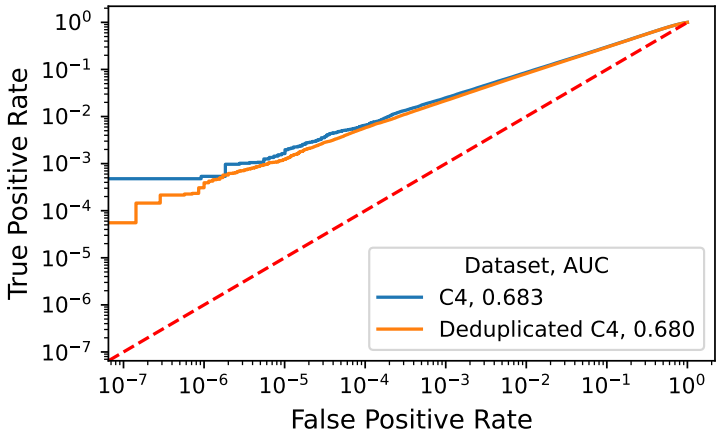



Figure 15: **The role of duplicates on MIA vulnerability.** We observe no significant differences (particularly as FPR increases) between models trained on C4 and de-duplicated C4.

**Investigating the role of duplicate training examples.** Given the relationship between MIA and memorization, and that prior work observes an important relationship between memorization and training-data duplication [21], we test the relationship between MIA vulnerability and the presence of duplicates. In Figure 15, we test the Chinchilla-optimally trained 140M model on C4 and a de-duplicated version of C4. We de-duplicate C4 according to methodology described in Lee et al. [21], where we remove sequences that share a common prefix of at least some threshold length. This reduced the C4 dataset size from 364,613,570 to 350,475,345 examples. We observe that the presence of duplicates has a negligible impact on AUC: it is 0.683 for C4, and 0.680 for de-duplicated C4. In other words, at least in terms of average attack success, the presence of duplicates does not seem to have a significant impact. However, further work is needed to assess attack success changes with more stringent de-duplication, since our de-duplication procedure only remove 10M examples from the dataset.

**Varying training epochs and dataset size.** In Figure 16, we reduce the training set size from 7M (Figure 16a) $2^{19} \approx 500K$ (Figure 16b) on the 140M model and train for 10 (Figure 16a) and 20 epochs (Figure 16b).

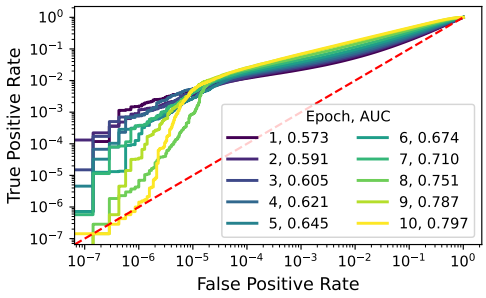
Figure 16 consists of two subplots, (a) and (b), both showing ROC curves that illustrate how MIA vulnerability changes with an increasing number of training epochs. The goal of these experiments are to investigate if MIA becomes better with more training epochs, and if so, how attack performance improves over epochs as a function of training dataset size.

- **Subplot (a)**: This plot shows MIA performance for a model (indicated as 140M trained on approximately 7 million examples across 10 epochs. The AUC increases with more epochs, starting from 0.573 at 1 epoch and reaching 0.797 at 10 epochs.
- **Subplot (b)**: This plot shows a more dramatic increase in MIA vulnerability for a 140M model trained on $2^{19}$ (approximately 524,000) points over 1 to 20 epochs. The AUC starts at 0.604 for 1 epoch, rapidly increases to 0.864 by 2 epochs, 0.944 by 3 epochs, and approaches perfect MIA (AUC close to 1.000) after 13 epochs.

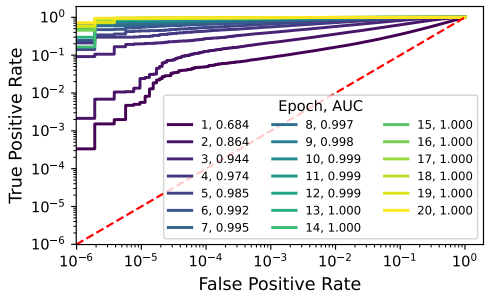
## D Discussion and other experimental results

**Does memorisation imply strong membership inference attacks?**

While memorisation is a key factor that can make a model susceptible to membership inference attacks, it does not automatically guarantee that strong MIAs will always be successful. Memorisation refers to a model learning specific details about its training data, rather than just general patterns.




(a) 140M model with training set size of $\approx 7M$ examples for 10 epochs.




(b) 140M model with training set size of $2^{19} \approx 500K$ examples for 20 epochs.

Figure 16: ROC curves demonstrate that MIA success significantly increases as models are trained for more epochs. (a) A 140M model shows AUC rising from 0.573 (1 epoch) to 0.797 (10 epochs). (b) Another 140M model trained on a smaller dataset shows a rapid escalation in AUC, from 0.604 (1 epoch) to near-perfect inference (AUC $= 1$) by 13-20 epochs, highlighting that overfitting from prolonged training severely heightens privacy risks.

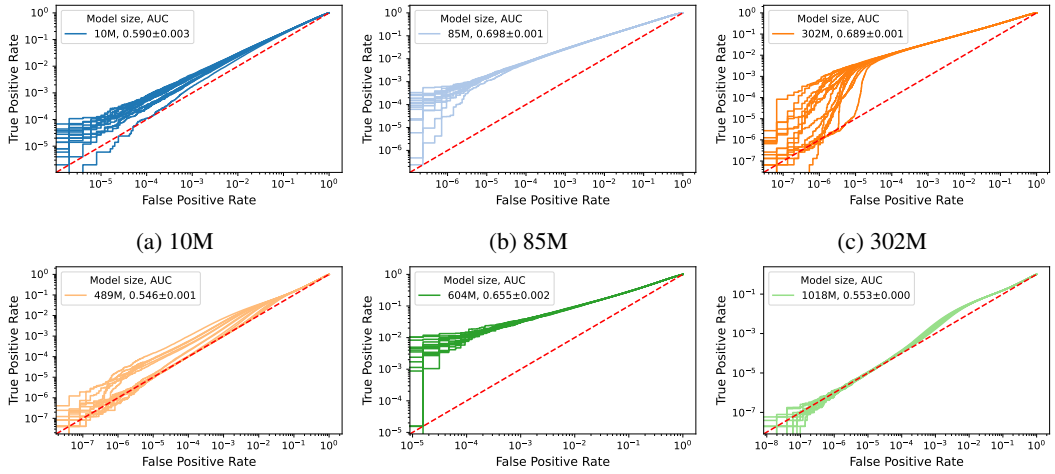
When a model heavily memorises training samples, it often exhibits distinct behaviours for these samples, which MIA attackers, in principle, can exploit. Indeed, studies have shown that the risk of membership inference is often highest for those samples that are highly memorised. However, the practical success and strength of a MIA can also depend on other factors, such as the model architecture, the type of data, the specifics of the attack method, and whether the memorisation leads to clearly distinguishable outputs or behaviours for member versus non-member data. Some models might memorise data in ways that are not easily exploitable by current MIA techniques, or the signals of memorisation might be subtle for well-generalising models, making strong attacks more challenging despite the presence of memorisation.

**How do we calibrate the FPR in practice?**

We acknowledge that calibrating the False Positive Rate (FPR) in real-world scenarios is a challenging and unresolved issue. The key difficulty lies in getting the necessary cooperation or data from different parties (e.g., model developers, data owners) to establish a reliable baseline for what constitutes a "false positive" in a practical setting [53].

**Full results for Figure 2b and Figure 4b**

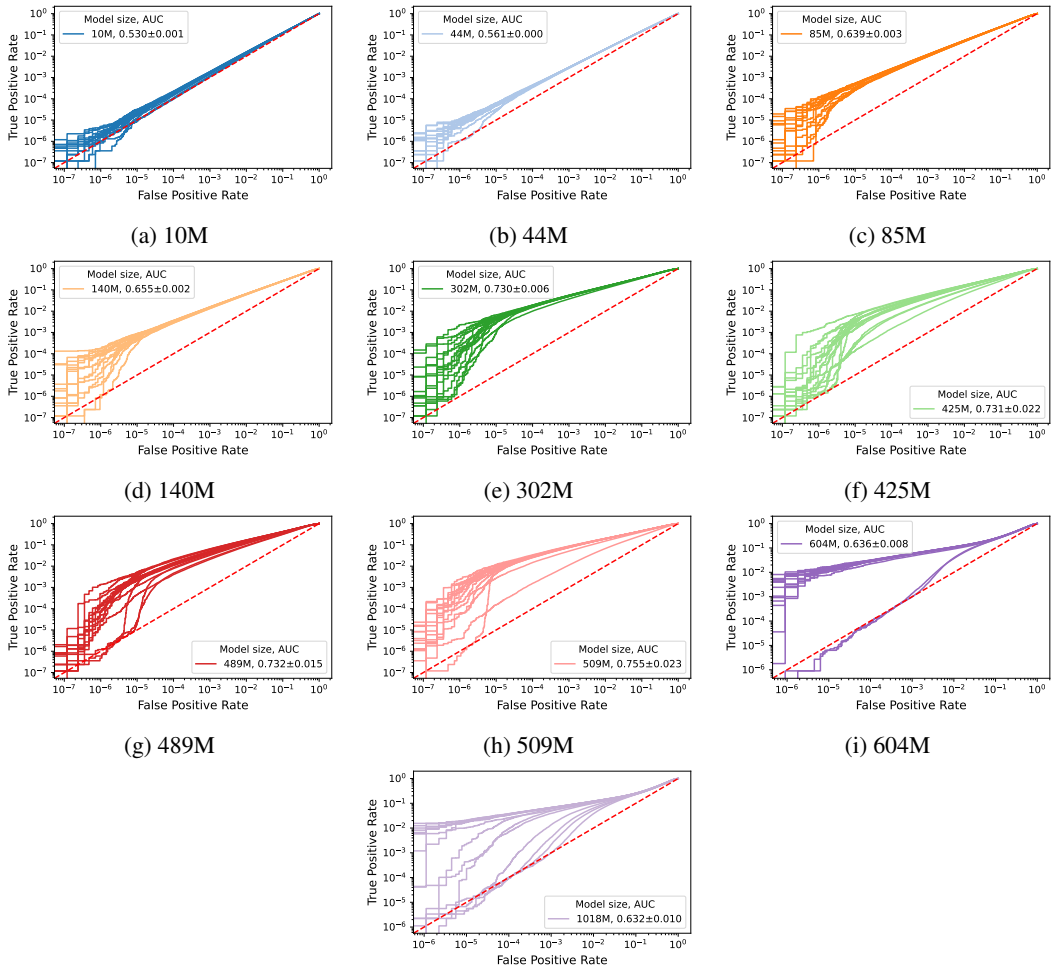
In Figure 17 and Figure 18 we give individual ROC curves for experimental results summarized in Figure 2b and Figure 4b, respectively. For each subplot, each line indicates a different target model that we use to perform the attack on. As discussed previously, some larger models appear to have more variance in their ROC curves over different experimental runs. In Figure 18i, we see that although AUC is similar over different target models, there is catastrophic failure against one model at small FPRs.




(a) 10M (b) 85M (c) 302M

(d) 489M (e) 604M (f) 1018M

Figure 17: Accompanying AUC-ROC curves for Figure 2b over different model sizes. For each subplot, each line indicates a different target model that we use to perform the attack on.




(a) 10M (b) 44M (c) 85M

(d) 140M (e) 302M (f) 425M

(g) 489M (h) 509M (i) 604M

(j) 1018M

Figure 18: Accompanying AUC-ROC curves for Figure 4b over different model sizes. For each subplot, each line indicates a different target model that we use to perform the attack on.

# E Experiment details

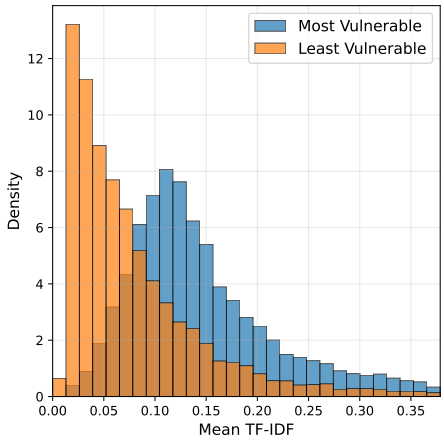
In Table 1, we give experimental hyperparameters and details. Unless otherwise stated, we used the AdamW optimizer [24] with a cosine scheduler. The initial learning rate is set to $10^{-7}$ and increases linearly over 750 warm up steps to a peak learning rate of $3 \cdot 10^{-4}$, after which it decreases according to the cosine schedule to a final value of $3 \cdot 10^{-5}$. We use 128 reference models, and a single target model to measure MIA vulnerability over $2\times$ the training set size (the training set is subsampled from a dataset twice its size). That is, for each reference and target model, the training set is subsampled from the same, larger dataset. This means each example in this larger dataset falls into the training set of $\approx 64$ reference models. The batch size is fixed to 128 and sequence length to 1024, if an example has fewer tokens we pad to 1024. The weight decay is set to 0.1, and a global clipping norm is set to 1.0. Note that we can approximately convert the training set size to total number of training tokens by multiplying the training set size by 400, as this the approximate average number of tokens within a C4 sample. This means, e.g., in Figure 2 the 1018M model was trained on 20.4B tokens.

Table 1: Experimental details

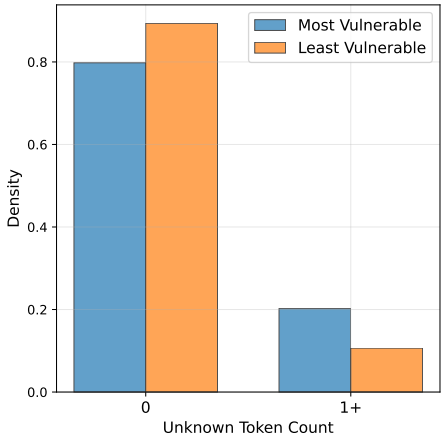
| Experiment | Training set size | Model size | Other information (which diverges from default experimental settings) |
|---|---|---|---|
| Figure 8a | 7M | 140M | Max 256 reference models |
| Figure 8b | 7M | 140M | Max 64 reference models, 10K Z population |
| Figure 2 | 500K<br>2.2M<br>4.25M<br>7M<br>15.1M<br>24.4M<br>30.2M<br>50.9M | 10M<br>44M<br>85M<br>140M<br>302M<br>489M<br>604M<br>1018M | |
| Figure 3a | 2.2M<br>1.1M | 44M<br>44M | |
| Figure 3b | 7M | 140M | 10 epochs |
| Figure 4a | 50K<br>100K<br>500K<br>1M<br>5M<br>10M | 140M<br>140M<br>140M<br>140M<br>140M<br>140M | 80 warm up steps |
| Figure 4b | $2^{23}$ | 10M<br>44M<br>85M<br>140M<br>302M<br>425M<br>489M<br>509M<br>604M<br>1018M | |
| Figure 3 | $2^{23}$ | 140M | |
| Figure 6 | 7M | 140M | |
| Figure 9 | 7M | 140M | 256 reference models |
| Figure 10 | 500K | 10M | 10K Z population size |
| Figure 11 | 500K | 10M | 10K-300K Z population size |
| Figure 12 | 500K | 10M | 10K-300K Z population size |
| Figure 7 | 7M | 140M | |
| Figure 14b | 50K | 140M | Cosine, Cosine with 0 weight decay, Cosine with no clipping, Linear. We use 50 warm up steps. |
| Figure 15 | 7M | 140M | |
| Figure 16a | 7M | 140M | 10 epochs |
| Figure 16b | $2^{19}$ | 140M | 20 epochs |
| Figure 17 | - | - | Identical to Figure 2 where we use 16 different target models |
| Figure 18 | - | - | Identical to Figure 4b where we use 16 different target models |
| Figure 19 | - | - | Identical to Figure 5b |
| Figure 20 | $2^{23}$ | - | 10M-302M model sizes |
| Figure 22 | - | - | Identical to Figure 3 |

# F More per-example MIA results

Figure 5b indicates that it is often the case that vulnerable sequences tend to be longer. Beyond sequence length, we observe that samples more vulnerable to MIA tend to have higher mean TF-IDF scores (Figure 19a), suggesting that texts with distinctive, uncommon terms leave stronger signals for membership inference. We compute these TF-IDF scores without normalization, collecting document frequency statistics over a random subsample of the original dataset, then taking the mean across all tokens in each sample. Similarly, examples containing unknown tokens (<unk>) appear more vulnerable to MIA (Figure 19b).




(a) Mean TF-IDF scores across vulnerability categories




(b) Unknown token (<unk>) counts across vulnerability categories

Figure 19: **Text property distributions by MIA vulnerability.** The most vulnerable examples tend to have higher TF-IDF scores compared to least vulnerable examples (**a**), and more likely to contain at least one unknown token (**b**).

## G Evolution of losses over different model sizes

In Figure 20, we plot the evolution of losses over different model sizes for three examples in the C4 dataset. Each of these models are trained for 1 epoch on $2^{23} \approx 8.3$M samples. This is a sanity check that the losses decreases (on the same sample) as the model size increases. It is also interesting to note that the distance between member and non-member distributions doesn't significantly shift as the model size grows.

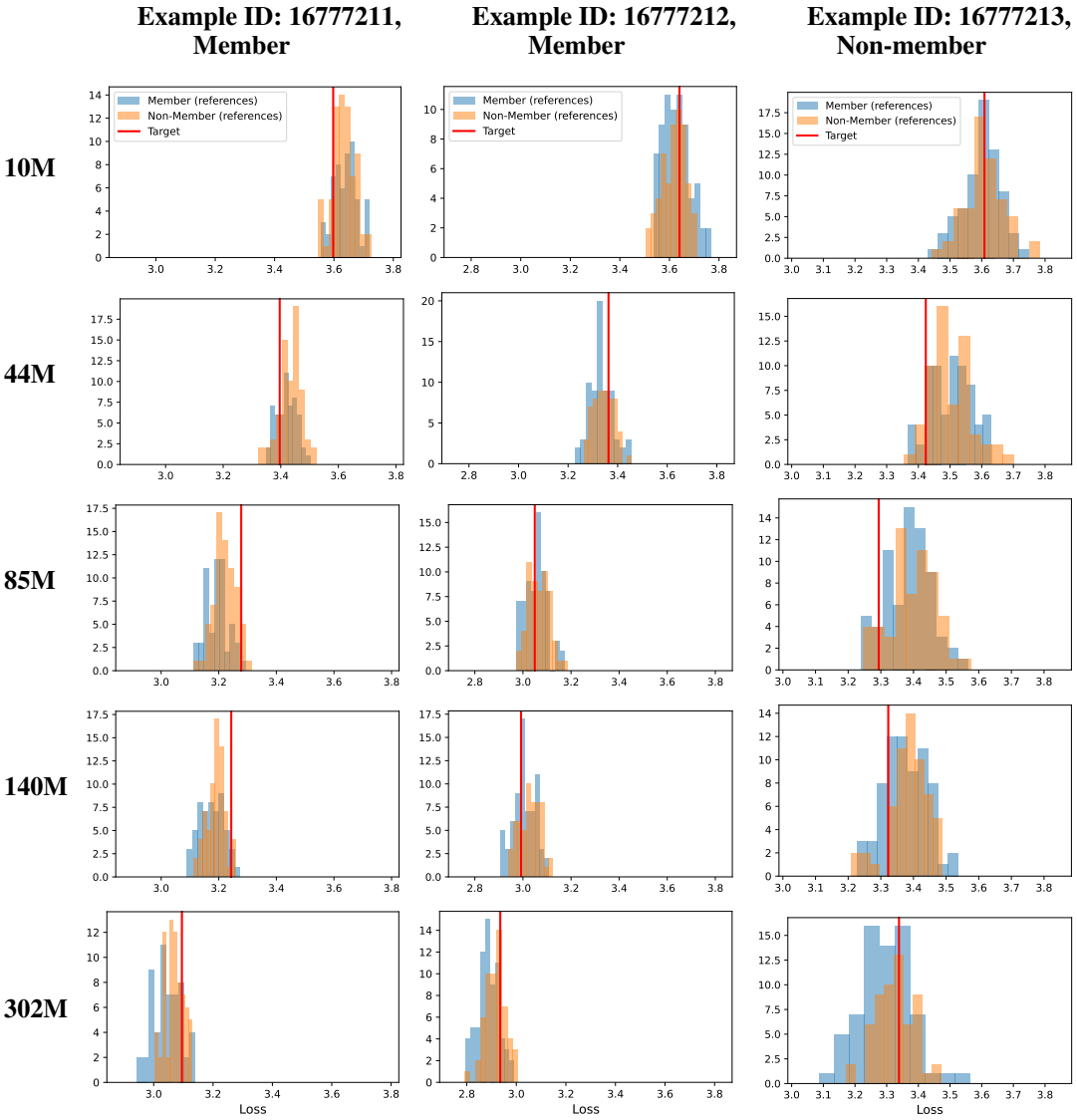
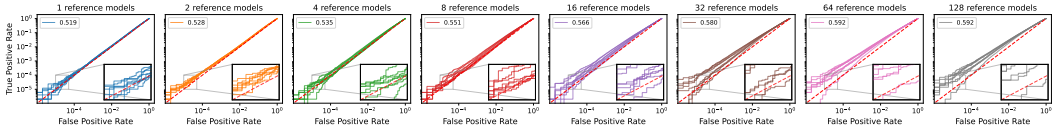


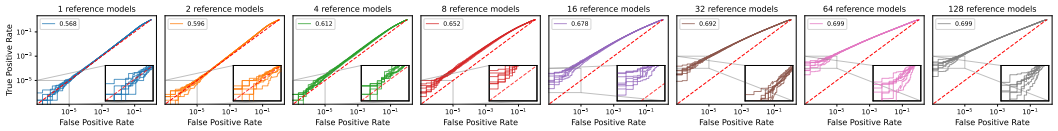

Figure 20: For three different samples (referenced by their ID in the C4 dataset, and if they were a member or non-member of training), we plot the loss of the reference distributions and the loss of the sample of the target model (as a vertical red line). We plot this over different model sizes ($y$-axis).

# H Comparing MIA over different number of reference models for all Chinchilla-optimally trained model sizes

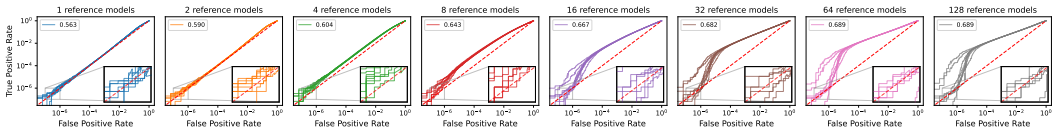
In Figure 21 we replicate Figure 17, where we vary the number of reference models used in LiRA. *Each column represents LiRA which uses a different number of total reference models to perform the attack*. Unsurprisingly, as more reference models are used the attack becomes better. This mirrors our findings in Figure 8a. The key point of these figures is to show the general pattern of where the ROC curve is relative to the reference line $y = x$, as well as the fact that there is variance (in the insets) across runs. These are (as a result) not to be taken as detailed results that should be closely examined. (This is why they are not very large.)



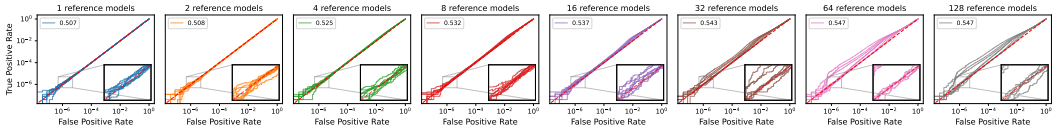

(a) 10M



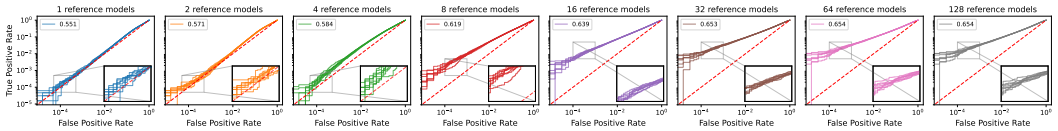

(b) 85M



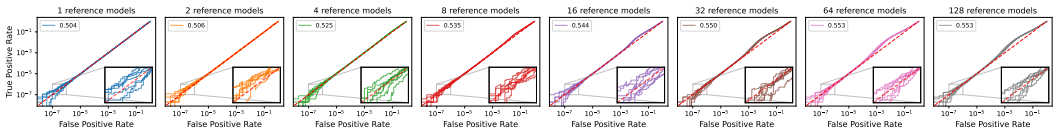

(c) 302M




(d) 489M




(e) 604M

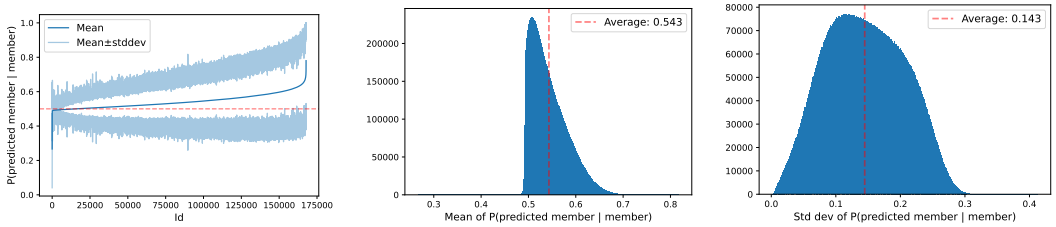



(f) 1018M

Figure 21: Accompanying AUC-ROC curves for Figure 2b over different model sizes. For each subplot, each line indicates a different target model that we use to perform the attack on. *Each column represent LiRA which uses a different number of total reference models to perform the attack*. Unsurprisingly, as more reference models are used the attack becomes better. This mirrors our findings in Figure 8a. Each subplot also records the average AUC of the attack.

# I Showing variance in sample predictions

As noted in Section 5.1, we observe significant degrees of instability in some membership predictions: there is considerable variance in the underlying sample true-positive probabilities. At any particular training step, the true-positive probabilities over a batch of samples can vary by more than 15%. In this appendix, we provide some additional figures that dig into this instability. We plot the mean and standard deviation of the per-sample true-positive probabilities, $P(\text{predicted as member}|\text{member})$ for $2^{24} = 16,777,216$ samples. We compute variance across 64 target models; this experiment trained 128 models on different random splits of the $2^{24}$ samples. We loop over each model, selecting it as the target model and the remainder as reference models used for LiRA. Since each sample had a probability of 0.5 for inclusion in the training set, for each sample, we have on average 64 target models where the sample was in training.




(a) Per-sample true positive probability for $2^{24}$ samples (mean and standard deviation computed over 64 target models). Ordered from smallest to largest.

(b) Histogram of average per-sample true positive probabilities from Figure 22a.

(c) Histogram of standard deviation of per-sample true positive probabilities from Figure 22a.

Figure 22: **Different views of instability in per-example true positive probabilities.** We compute the mean and standard deviation for each sample's true positive probability (i.e., $P(\text{predicted as member}|\text{member})$) for $2^{24}$ samples across 64 target models. (a) shows the mean and variance of these true positive probabilities, where we sort the results by the mean of each example's true positive probability. (b) and (c) together show a different view of the same data; the former shows a histogram of the mean true positive probabilities for these examples, and the latter shows the histogram of the standard deviation.

In Figure 22, we provide three plots that give different views of the same data. Figure 22a plots the true positive probability for each example. We sort examples by the mean value of their true-positive probability (i.e., the mean of $P(\text{predicted as member}|\text{member})$ over 64 target models), and we also show the variance over the 64 target models.

Together, Figures 22b and 22c provide an alternate view of Figure 22a. Figure 22b plots the histogram of the mean $P(\text{predicted as member}|\text{member})$ for the $2^{24}$ samples, each across the 64 target models. The average across these mean true positive probabilities for each example is 0.543. However, note that this is a skewed distribution; there are many examples that have their mean $P(\text{predicted as member}|\text{member}) > 0.6$. Figure 22c shows a related histogram: the standard deviations for the mean per-example true positive probabilities shown in Figures 22b. On average, the standard deviation for an example's true positive probability is close to 15%, as we note in the figure; it is 0.143. Note that there is a large amount of mass on either side of this average. Importantly, there are many examples for which the standard deviation of the true positive probability computed over 64 target models exceeds 0.2.

Overall, variance is significant. The individual example true positive probabilities for each target are, when considered together, highly unstable. This variance can help explain why attack ROC AUC is perhaps lower than one might have hoped; there is considerable variance in the underlying example predictions. Altogether, this provides additional nuance concerning the extent of (alternatively, the limits of) attack robustness.

## NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The results we provide in Sections 3, 4, and the Appendix provide an accurate and nuanced treatment of the main claims introduced in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In the introduction, we note the cost of our work, which is a limitation for those that wish to reproduce our experiments. We document the challenges we observe with MIA attack variability.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [NA]

Justification: While the strong attacks that we investigate in this paper are theoretically grounded, our main contributions are empirical. We do not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: While the cost of training thousands of LLMs ranging 10M to 1B parameters is substantial, those with the resources to do so would be able to faithfully reproduce our main results. We thoroughly document the tools we use and our experimental configurations throughout the paper, as well as in one centralized place in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: As discussed in the prior answer, while we do not link to our code repository, we provide ample details on the open model architectures [23] and datasets [38] that we used to conduct the experiments in this paper.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide extensive details on our general setup in Section 3, on more specific experimental configurations in the following sections, and more detailed information in the Appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We discuss variability and instability across different attacks for 140M models. See Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide these details in a centralized location in the Appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We read the NeurIPS Code of Ethics and confirm that the research conducted in this paper conforms in every respect.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss and motivate our work with respect to the broader impact it has regarding developing scientific knowledge about improving the privacy of LLMs.

Guidelines:

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We use open-source model architectures [23] and datasets [38], which we credit in several places in the main paper, Appendix, and this checklist.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification: The apper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.