CheckMIABench: Firm Foundations For Membership Inference Attacks on Language Models

Jeffrey G. Wang*[1], Jason Wang*[1], Marvin Li*[1], Seth Neel[2]
[1] Harvard University, [2] Harvard Business School
Correspondence: {jgwang21, sethneel.ai}@gmail.com

Abstract

Membership inference attacks (MIAs) are a canonical way to assess a machine learning model's privacy properties. Although several attempts have been made to evaluate MIAs on language models, the extant literature has suffered numerous difficulties in constructing clean evaluations to test new techniques. In particular, subtle distribution shifts between member and non-member sets can undermine the statistical validity of MIAs; recent work has underscored this by showing that "blind" methods with no access to the underlying model can perform far better than published methods on the same benchmarks. This paper constructs a benchmark for principled evaluation of MIAs against LLMs, by leveraging the insight that training data before and after a fixed point during training are drawn from the same distribution. Therefore, all open-source models with intermediate checkpoints and public training data can be converted into MIA testbeds. We apply our framework to a half-dozen published attacks on the Pythia and OLMo family of models, from 70M to 7B parameters. To facilitate further privacy research, we open-source a modular library for designing and implementing attacks in this setting: https://github.com/safr-ai-lab/pandora_llm.

1 Introduction

Large language models (LLMs) have become indispensable workhorses for knowledge-intensive tasks, from summarizing medical records and drafting clinical notes to screening legal contracts and flagging anomalous financial transactions. As these models are increasingly deployed in high-stakes arenas such as healthcare and finance, privacy has become a first-order requirement for responsible use. As such, recent security research has revisited a threat model long studied for classifiers and regressors: that of membership inference attacks (MIAs) (Liu et al., 2021; Shokri et al., 2017). In an MIA, an adversary with some level of access to the model attempts to decide whether a particular example was used in a model's training data ("member") or is merely drawn i.i.d. from the same distribution ("non-member"). Highly accurate MIAs against LLMs would be useful not only to demonstrate privacy leakage, but could also be adopted as practical probes for related phenomena, including quantifying memorization during training (Zhou et al., 2023) and empirically evaluating the success of machine unlearning techniques (Kurmanji et al., 2023; Pawelczyk et al., 2023; Hayes et al., 2024).

While there has been a recent surge of research on MIAs against LLMs (Duan et al., 2024; Li et al., 2023; Mattern et al., 2023), it has been notoriously difficult to implement a correct MIA evaluation benchmark for pre-trained LLMs (Das et al., 2025; Maini and Suri, 2024). In particular, Das et al. (2025) showed that many evaluations for MIAs commonly used in the literature are beaten by simple supervised learning methods trained on i.i.d splits of member/non-member data that have no access to the underlying model. This means the accuracy of these MIAs primarily emerges from the distributional differences between member and non-member examples rather than the privacy properties of underlying LLM. For instance, Shi et al. (2023) proposed using *temporal differences* (from training data cutoffs of models) to construct member and non-member sets with the WikiMIA benchmark, where Wikipedia articles written before Jan 1st, 2017 were treated as the member data and articles after Jan 1st, 2023 were treated as non-member data. While reasonable at first glance, as such cutoffs enforce the member/non-member distinction, these sets are separable for another reason: their contents are distinct due to changing writing patterns over time. Indeed, Das et al. (2025) trained a simple bag-of-words classifier that could distinguish between member and non-member data with-

*Equal contribution.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 364–370

out the underlying model with a True Positive Rate (TPR) of 94.7% at a False Positive Rate (FPR) at 5%, far outperforming the best attack proposed on the benchmark.

In this paper, we begin with an overview of the challenges researchers have faced in creating theoretically clean evaluations for membership inference. Then, we instantiate a cleaner dataset split for two common open LLM families, sanity checking them with "blind baselines" to verify the splits do not suffer from those same challenges. Finally, we benchmark a slate of common methods on these clean splits, finding more limited performance.

2 Related Work

MIAs refer to a class of methods that can determine if a given data point z was included in the training dataset of a particular model θ. They were initially motivated by privacy considerations; if an adversary can determine if z was used to train θ with accuracy higher than the base rate, then information theoretically θ *must* encode some information about z that the attack is able to leverage. The setup to evaluate such MIAs is fairly straightforward: given an initial dataset D, some subset of the data is set aside prior to training (or sampled from the distribution after training) as a validation set D_{nonmem}, and some dataset D_{mem} is used for training to produce a new model θ. Then the following process is repeated a set number of times:

1. Flip a fair coin. If heads, sample with replacement $z \sim D_{\text{mem}}$; otherwise, sample $z \sim D_{\text{nonmem}}$.
2. Given z and some access to θ, the MIA produces a score p for that data point.

Given a membership inference score, for any threshold τ, there is a corresponding membership inference attack that predicts $z \in D_{\text{mem}}$ if and only if $p < \tau$. Each τ corresponds to a point on the Receiver Operating Characteristic (ROC) curve for the binary classification. As is common in the literature, to evaluate MIAs in this paper, we report the area under the ROC curve (AUC) as well as the TPR at low FPR of each method. The latter metric is widely used in the literature because any attack which can extract with high confidence even a small fraction of the training data poses a serious privacy risk (Carlini et al., 2022).

One key assumption underlies this evaluation scheme: individual samples in the training and validation data are drawn i.i.d. from the same distribution $\mathcal{P}$. Clearly, if θ contains no information about D_{mem} then the maximum attack accuracy is 50%, since D_{mem} and D_{nonmem} have identical marginal distributions. Any additional information about whether $z \sim D_{\text{mem}}$ or $z \sim D_{\text{nonmem}}$ must come from the model θ. Indeed, if D_{mem} and D_{nonmem} differ in some way that is independent of whether θ is trained on it, then the maximum attack accuracy can be 100% even if the θ contains no information about D_{mem}. For instance, if one took a standard training and validation dataset for an LLM and prepended `The quick, brown fox jumps` to only the train dataset, an MIA without any access to θ could still achieve near-perfect accuracy by simply detecting if a given z begins with this phrase.

Existing Evaluations. Existing evaluations for MIAs fall in three broad categories. Several papers introduce benchmarks using the temporal cutoff approach of WikiMIA (Shi et al., 2023; Liu et al., 2024). Others utilize train/val splits of open models, most commonly using The Pile dataset with the Pythia family of models (Biderman et al., 2023; Gao et al., 2020). The MIMIR benchmark refines this approach, with an additional deduplication step of the non-member sets against training data[1] (Duan et al., 2024).

Challenges of Deduplication. The Pythia family of models includes those trained on a deduplicated version of The Pile's training set; however, since only the training data is deduplicated, it has a different distribution than the validation split of The Pile. One solution, used in MIMIR, deduplicates the validation set against the train set—but this approach is not perfectly sound either, as this induces a different marginal distribution on member vs. nonmember points. To see this, consider the setting where there are two very rare documents in the corpus. If both documents are split into the training corpus, only one will be included post deduplication; on the other hand, if they are both included in the validation split, they will both make it through pre-processing since they will only be deduplicated against the training set. Indeed, Meeus et al. (2025) find that a bag-of-words classifier is able to achieve extremely high AUCs on certain splits (up to 0.86), implying that deduplication causes distri-

[1] They create several member/non-member splits such that all non-member points with more than a p-proportion overlap in n-grams with any training data point are removed, for various settings of n and p.

bution shifts that violate the MIA assumptions. The correct approach then, would jointly deduplicate the entire corpus prior to randomly splitting it into train and validation sets, which can't be done ex-post. Joint deduplication would also mitigate the issue of "fuzzy membership" that also poses a barrier to rigorous MIA evaluations in LLMs: Duan et al. (2024) show there is substantial overlap between the training and validation sets for The Pile, which is the dataset used to train the Pythia and GPT-NeoX series of models, an oft-used benchmark for MIA papers.

It is important to note that these difficulties are in some sense unique to LLMs; in classical supervised learning, the evaluator will typically have access to the entire dataset and can choose the train/validation partition before training the model.

Concurrent Work. Kim et al. (2025) also suggest a checkpoint-based approach to evaluate MIAs on OLMo, but they neither verify that their member/non-member splits are clean via blind baselines nor release a checkpoint-based MIA dataset for The Pile, as we do in this paper. While our checkpoint-based benchmark demonstrates that existing MIAs fail to achieve statistically-significant results, Hayes et al. (2026) more recently demonstrated that scaling up strong white-box MIAs can succeed for LLMs even in the pre-training setting. This method requires the pre-training of many reference models and is thus extremely computationally expensive, so we are unable to evaluate it on our benchmark.

3 Our MIA Evaluation Pipeline

In this section, we will describe a clean method to derive member and non-member splits. This method only relies on access to the model checkpoints in the middle of training and the training data order, which is available for many models, like the Pythia family or OLMo (Groeneveld et al., 2024; Kim et al., 2025). We will also describe the procedure to generate member and non-member data using different checkpoints during the training, checking it against blind MIAs as in (Das et al., 2025; Maini and Suri, 2024).

Member and Non-member Data Generation. Recall that the LLM data collection and training procedure typically involves forming documents taken from links scraped from the web, which are then tokenized and packed into training examples of a fixed size. These packed examples are deduplicated against each other and shuffled to form the *training order* of the model (Figure 1). This shuffling of the training order will be the key to our MIA evaluation setup. In our setup, we select a model, e.g., Checkpoint 300 in Figure 1, using the data before the checkpoint as the member data (herein {B2,B4,B1}) and the data after the checkpoint (herein {B3}) as the non-member data. We then evaluate the model at that checkpoint. Because the data before and after are drawn from the same distribution, the member and non-member data have the same marginal distributions.

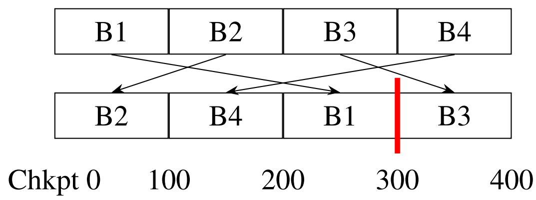

Figure 1: The shuffling of packed samples to construct the training data of Pythia and our checkpoint-based method to evaluate MIAs. In this scenario, we would sample from chunks {B2,B4,B1} for member data and chunk {B3} for the non-member data to evaluate MIAs for the model at Checkpoint 300.

Generating and Learning from Features. Using member and non-member data generated in the previous steps, we can evaluate different MIAs. We release our codebase as a modular Python library, with documentation and tests, that allows one to implement and benchmark new MIAs in this setting. We also implement (and in Section 4, benchmark) many MIAs from extant literature, including: simple loss thresholding (Yeom et al., 2018), Min-K (Shi et al., 2023), Min-K%++ (Zhang et al., 2025), Zlib entropy (Carlini et al., 2020), ReCaLL (Xie et al., 2024), and MoPe (Li et al., 2023).

Blind MIAs. Finally, our pipeline implements many "blind" MIAs as a baseline check that member and non-member data don't have distributional differences easily checkable with a simple supervised learning method. For a given set of splits, we train classifiers on statistical representations of the text, including Bag of Words, TF-IDF, Word2Vec representations (Mikolov et al., 2013), and BERT embeddings (Devlin et al., 2019). Full details on the training details of these classifiers are available in the Appendix.

4 Results

Setup. We instantiate our evaluation on the Pythia family of models, which were trained for 143,000 optimizer steps on a deduplicated version of The Pile, representing approximately 1.5 epochs. We evaluate the model at step 97,000 (over 95% of the way through a full epoch), uniformly sampling data points before and after this step.[2]

We also benchmark our method on the OLMo family of models trained on the Dolma dataset (Groeneveld et al., 2024; Soldaini et al., 2024). Due to computational constraints, we only benchmark OLMo results on the 7B parameter model. We use the checkpoint at step 400,000, which represents 88% of the way through the first epoch of training. See full results in Appendix C.

Blind MIAs. First, we validate that blind MIAs without any access to the model, like supervised learning techniques on a split of member and non-member data points, get no signal in our setting (Table 1). In all settings, we train three different classifiers (logistic regression, random forest, and a neural network) on 4,000 points from member and non-member classes using the tokens or text alone; then we evaluate our classifier on 1,000 points from each class. We report the best test AUC among these three methods in Tables 1 and 3, providing standard errors from 1,000 bootstraps. Full details on the supervised classifiers are in Appendix A.

Table 1: Sanity check: supervised model-free ("blind") MIAs get no signal on our cleaned split of The Pile.

Blind MIA	AUC	AUC SE	$\mathbf{TPR}_{1\%}$
BoW (Tokens)	0.505	0.0128	0.01
TFIDF (Tokens)	0.503	0.0127	0.006
W2V (Text)	0.490	0.0131	0.015
BERT (Text)	0.497	0.0127	0.012

Existing MIAs. After validating our dataset, we benchmark the MIAs listed in Section 3, which we implement in our library. We find that current MIAs have limited success on our dataset drawn from the pre-training data (see Table 2). Note that this does not exclude the possibility that on certain subsets of the training data (in particular, the GitHub documents in The Pile), MIAs can still be performant, which was found in previous works (Duan et al., 2024). We note that while one attack, MoPe, achieves performance that is 2.5x above the random baseline as measured by $\mathbf{TPR}_{1\%}$ on the 70m model, this performance diminishes on larger models. None of the attacks proposed achieve AUCs that are statistically significantly above the blind baseline of 0.5 across any model size from 70m to 2.8b.

[2] Note that the use of intermediate checkpoints is an additional access assumption, but most literature already assumes access to a 1-epoch intermediate checkpoint to ensure that each datapoint is seen a uniform number of times.

Table 2: We benchmark several MIAs against the Pythia family of models on our cleaned split of The Pile.

Model	MIA	AUC	AUC SE	$\mathbf{TPR}_{1\%}$
70m	LOSS	0.513	0.0130	0.006
70m	Min-K	0.500	0.0130	0.012
70m	Min-K++	0.496	0.0128	0.011
70m	Zlib	0.499	0.0129	0.016
70m	MoPe	0.487	0.0126	0.025
70m	ReCaLL	0.490	0.0130	0.010
160m	LOSS	0.513	0.0130	0.007
160m	Min-K	0.500	0.0127	0.011
160m	Min-K++	0.499	0.0126	0.014
160m	Zlib	0.499	0.0127	0.016
160m	MoPe	0.495	0.0132	0.015
160m	ReCaLL	0.490	0.0124	0.006
410m	LOSS	0.512	0.0128	0.007
410m	Min-K	0.498	0.0132	0.006
410m	Min-K++	0.496	0.0126	0.014
410m	Zlib	0.499	0.0127	0.016
410m	MoPe	0.504	0.0130	0.014
410m	ReCaLL	0.493	0.0131	0.008
1b	LOSS	0.513	0.0122	0.007
1b	Min-K	0.502	0.0125	0.007
1b	Min-K++	0.498	0.0126	0.013
1b	Zlib	0.500	0.0129	0.016
1b	MoPe	0.495	0.0126	0.015
1b	ReCaLL	0.487	0.0123	0.008
2.8b	LOSS	0.506	0.0131	0.007
2.8b	Min-K	0.489	0.0124	0.006
2.8b	Min-K++	0.489	0.0126	0.014
2.8b	Zlib	0.499	0.0129	0.016
2.8b	MoPe	0.493	0.0131	0.006
2.8b	ReCaLL	0.498	0.0131	0.010

5 Conclusion

In this paper, we identify subtle distributional pitfalls of previous MIA evaluations for LLMs and propose a principled framework to avoid them. Our pipeline supports several popular open model families and we benchmark many existing attacks on it. Finally, this work provides a clear call-to-action for future model releases: the underlying dataset should be processed and deduplicated in tandem, then split into training and validation sets to ensure matching training and validation distributions.

6 Limitations

While our setting is general and can be instantiated across many open model families, there are several limitations and areas for future work. First, our framework requires open training data, model checkpoints, and the exact permutation of training examples throughout training. While many open model families provide this information, others, e.g., Llama and Qwen, do not (Touvron et al., 2023; DeepSeek-AI et al., 2025; Yang et al., 2025). Despite this important limitation to any checkpoint-based approach, we argue that this methodology enables us to investigate MIA efficacy much more rigorously than existing work, and hope that it encourages model providers to release this auxiliary information in the future. Second, due to computational and data constraints, we only evaluated MIAs on models pretrained almost exclusively on English web data. Finally, in our experimental setup, we do not benchmark any models larger than 7 billion parameters, again due to compute constraints. We are excited by future work verifying the extent of our negative results across scales, data subsets, languages, and training paradigms.

References

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *Preprint*, arXiv:2304.01373.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. *Preprint*, arXiv:2112.03570.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Xiaodong Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting training data from large language models. *Preprint*, arXiv:2012.07805.

Debeshee Das, Jie Zhang, and Florian Tramèr. 2025. Blind baselines beat membership inference attacks for foundation models. *Preprint*, arXiv:2406.16201.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. DeepSeek-V3 technical report. *Preprint*, arXiv:2412.19437.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

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

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. OLMo: Accelerating the science of language models. *Preprint*, arXiv:2402.00838.

Jamie Hayes, Ilia Shumailov, Christopher A. Choquette-Choo, Matthew Jagielski, George Kaissis, Milad Nasr, Sahra Ghalebikesabi, Meenatchi Sundaram Mutu Selva Annamalai, Niloofar Mireshghallah, Igor Shilov, Matthieu Meeus, Yves-Alexandre de Montjoye, Katherine Lee, Franziska Boenisch, Adam Dziedzic, and A. Feder Cooper. 2026. Exploring the limits of strong membership inference attacks on large language models. *Preprint*, arXiv:2505.18773.

Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. 2024. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. *Preprint*, arXiv:2403.01218.

Gyuwan Kim, Yang Li, Evangelia Spiliopoulou, Jie Ma, Miguel Ballesteros, and William Yang Wang. 2025. Detecting training data of large language models via expectation maximization. *Preprint*, arXiv:2410.07582.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. 2023. Towards unbounded machine unlearning. *Preprint*, arXiv:2302.09880.

Marvin Li, Jason Wang, Jeffrey Wang, and Seth Neel. 2023. MoPe: Model perturbation based privacy attacks on language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13647–13660, Singapore. Association for Computational Linguistics.

Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. 2021. When machine learning meets privacy: A survey and outlook. *ACM Comput. Surv.*, 54(2).

Zhenhua Liu, Tong Zhu, Chuanyuan Tan, Haonan Lu, Bing Liu, and Wenliang Chen. 2024. Probing language models for pre-training data detection. *Preprint*, arXiv:2406.01333.

Pratyush Maini and Anshuman Suri. 2024. Reassessing EMNLP 2024's best paper: Does divergence-based calibration for membership inference attacks hold up? Accessed May 1, 2025.

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

Matthieu Meeus, Igor Shilov, Shubham Jain, Manuel Faysse, Marek Rei, and Yves-Alexandre de Montjoye. 2025. SoK: Membership inference attacks on LLMs are rushing nowhere (and how to fix it). *Preprint*, arXiv:2406.17975.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

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

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *Preprint*, arXiv:2310.16789.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. *Preprint*, arXiv:1610.05820.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. *Preprint*, arXiv:2402.00159.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Zhenqiang Gong, and Bhuwan Dhingra. 2024. ReCaLL: Membership inference via relative conditional log-likelihoods. *Preprint*, arXiv:2406.15968.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. *Preprint*, arXiv:1709.01604.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2025. Min-k%++: Improved baseline for detecting pre-training data from large language models. *Preprint*, arXiv:2404.02936.

Zhenhong Zhou, Jiuyang Xiang, Chaomeng Chen, and Sen Su. 2023. Quantifying and analyzing entity-level memorization in large language models. *Preprint*, arXiv:2308.15727.

A Attack Details.

Supervised Learning for Classifiers. We use random forest, logistic regression, and a neural network as supervised classifiers for our model-free blind baselines (traditional word embedding features). For random forest, we use 100 trees with no limit on maximum depth and the Gini impurity splitting criterion. For logistic regression, we use a default lbfgs solver with 1,000 iteration maximum and L2-regularization. For neural network, we train for 10 epochs on a batch size of 128 using the Adam optimizer with a learning rate of 0.001, with a ReLU architecture of 4 layers going from input dimension to 250, to 100, to 10, then to 1 dimensions.

Evaluation Configuration. We evaluate all attacks using 1,000 train and validation points. For all supervised methods used in the Pythia blind baselines, we train on 4,000 points; for all supervised methods used in OLMo baselines, we train on 8,000 points.

Other Evaluation Details. MoPe uses 10 perturbations with $\sigma = 0.005$, as recommended in the paper (Li et al., 2023). For Min-K and Min-K++, we use $k = 0.1$. For ReCaLL, we use 100-token long prefixes as the extra conditioning.

B Dataset Details

Pythia Data Construction. We evaluate a Pythia checkpoint at 97,000 steps into training. To construct the dataset for evaluation, we randomly sample data points from steps 0 to 97,000, and then from 97,000 to 98,500 (approximately the end of Epoch 1). We run all attacks on Pythia models trained using *deduplicated* training data.

OLMo Data Construction. OLMo 7B was trained on 1.25 epochs from the 2T token training corpus Dolma for a total of 556,000 training steps. The remaining 0.25 epochs after the first epoch are taken from another shuffling of the training corpus. Because our checkpoint-based method is only valid if the model did not see the data after the checkpoint, we restrict our attention to the model state through the first epoch, after 452,000 training steps. We then choose to evaluate MIAs for the model at checkpoint 400,000. We choose the member data by randomly sampling from data that the model saw between checkpoints 0 and 400,000, and the non-member data by randomly sampling from data the model saw between checkpoints 401,000 and 452,000. Because the entire Dolma dataset already undergoes several different kinds of deduplicating before being used to train OLMo (see Section 5.4 of (Soldaini et al., 2024) for details), this guarantees that member and non-member data have the same marginal distributions.

C OLMo Results

Blind MIAs. As with the Pythia models in Section 4 of the paper, we run blind supervised baselines for the OLMo model as well. These results are given in Table 3.

Table 3: Sanity check: supervised model-free (“blind") MIAs get no signal on our cleaned split of Dolma.

Blind MIA	AUC	AUC SE	$\text{TPR}_{1\%}$
BoW (Tokens)	0.491	0.0131	0.01
TFIDF (Tokens)	0.491	0.0128	0.008
W2V (Text)	0.494	0.0129	0.012
BERT (Text)	0.496	0.0126	0.009

Other Attacks. We benchmark various MIAs from previous works, as in Section 4, this time against OLMo 7B. See Table 4 for full results.

Table 4: We benchmark several published MIAs against OLMo 7B on our cleaned split of Dolma.

MIA	AUC	AUC SE	$\text{TPR}_{1\%}$
LOSS	0.505	0.0125	0.01
Min-K	0.504	0.0129	0.013
Min-K++	0.490	0.0124	0.009
Zlib	0.499	0.0128	0.019
ReCaLL	0.521	0.0130	0.011

D Additional Details

Compute Estimates. To run the experiments, we used a compute node with an NVIDIA A100 80GB GPU. All experiments in this paper can be run on a single one of these GPUs. All results for pretrained MIAs are on model sizes 70M, 160M, 410M, 1B, and 2.8B for Pythia, and 7B for OLMo. To run blind baselines, we create features using 4,000 randomly sampled member/non-member points, train a classifier, and evaluate the classifier on 1,000 distinct member/non-member points. Most of these steps are runnable on a consumer laptop. As noted previously, in the pretrained setting, we evaluate all MIAs on 1,000 points from member and non-member splits, which requires only running inference on models. In MoPe, we run inference on ten times as many points (because we use ten perturbed models). In total these attacks took around 3 A100 GPU-days.

AI Assistants. While all work was done and checked by the authors, language models were used in the process to refine ideas, write small snippets of code, and tune writing for clarity.