Detecting Training Data of Large Language Models via Expectation Maximization

Gyuwan Kim[1*] **Yang Li**[2] **Evangelia Spiliopoulou**[2]
Jie Ma[2] **Miguel Ballesteros**[2] **William Yang Wang**[1†]
[1]University of California, Santa Barbara [2]AWS AI Lab
gyuwankim@ucsb.edu

Abstract

Membership inference attacks (MIAs) aim to determine whether a specific example was used to train a given language model. While prior work has explored prompt-based attacks such as ReCALL, these methods rely heavily on the assumption that using known non-members as prompts reliably suppresses the model's responses to non-member queries. We propose EM-MIA, a new membership inference approach that iteratively refines prefix effectiveness and membership scores using an expectation-maximization strategy without requiring labeled non-member examples. To support controlled evaluation, we introduce OLMoMIA, a benchmark that enables analysis of MIA robustness under systematically varied distributional overlap and difficulty. Experiments on WikiMIA and OLMoMIA show that EM-MIA outperforms existing baselines, particularly in settings with clear distributional separability. We highlight scenarios where EM-MIA succeeds in practical settings with partial distributional overlap, while failure cases expose fundamental limitations of current MIA methods under near-identical conditions.

1 Introduction

As large language models (LLMs) [1, 2] scale in capability, concerns have grown over the provenance and transparency of their training data [3, 4]. Uncertainty about data exposure raises legal and ethical risks, including privacy breaches [5, 6], copyright violations [7], and leakage of sensitive content [8].

Membership inference attacks (MIAs) provide a concrete lens on this problem by testing whether a given example was seen during training [9, 10]. They enable auditing for data contamination [11, 12, 13] and compliance with data usage policies [14, 15]. Yet, MIAs for LLMs remain difficult due to vast corpora and the subtle boundary between memorization and generalization in natural language [16]. Prompt-based MIA techniques such as ReCALL [17] assume that known non-members can serve as effective prompts for distinguishing members from non-members. However, we find that the effectiveness of such prompts is highly inconsistent and difficult to predict, motivating the need for a more adaptive approach that can account for variability in prompt effectiveness.

To address the limitations of approaches that rely on arbitrarily or randomly chosen prompts, we propose EM-MIA, a novel membership inference method that jointly refines prefix effectiveness and membership scores via expectation-maximization. Rather than relying on labeled non-members or assuming the quality of predefined prompts, EM-MIA iteratively estimates which prefixes are informative and which examples are likely members, enabling unsupervised bootstrapping of both prompt selection and membership scoring. This yields greater robustness across diverse settings, especially when prompt assumptions fail or non-member data is unavailable.

[*]Work done during an internship at AWS AI Labs
[†]Work done while at AWS AI Labs

NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling.

To support more controlled and reproducible evaluation, we also introduce OLMoMIA, a benchmark built from the pre-training corpus and checkpoints of the OLMo open-source LLM series [18]. Unlike existing benchmarks such as WikiMIA [19] and MIMIR [16], which provide limited control over the similarity between member and non-member examples, OLMoMIA allows researchers to systematically vary distributional overlap and assess how different methods perform across a range of difficulty levels. By partitioning the data based on semantic similarity and membership status with respect to the pre-training data, OLMoMIA supports fine-grained analysis of robustness, generalization, and failure modes in both easy and near-indistinguishable settings. Its design enables rigorous comparison of inference strategies under controlled conditions, and we will release both the benchmark and its generation pipeline to support scalable and reproducible MIA research.

Experiments show that EM-MIA outperforms existing MIA methods on WikiMIA across models of varying sizes and achieves robust results on OLMoMIA under systematically controlled difficulty conditions. In particular, EM-MIA demonstrates strong performance without access to labeled non-member data and maintains robustness to prompt variability, highlighting its practical value in realistic gray-box scenarios. At the same time, our results expose the inherent difficulty of membership inference when member and non-member distributions are nearly identical, which poses a significant challenge for all existing methods, including ours. These findings underscore the importance of evaluating MIA methods across a range of separability conditions and offer new insight into the limits and opportunities of prompt-based membership inference.

2 EM-MIA: Joint Estimation via EM

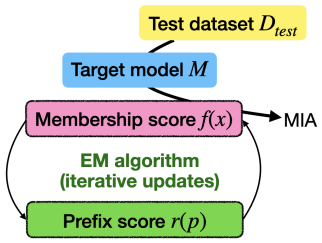
We consider membership inference in a gray-box setting, where the attacker has access to a language model $\mathcal{M}$ and can query $\mathcal{M}$ to obtain token-level probabilities or log-likelihoods. Given an input $x \in \mathcal{D}_{\text{test}}$, the goal is to predict a binary membership label indicating whether x was included in the pretraining corpus $\mathcal{D}_{\text{train}}$ of $\mathcal{M}$.

We provide the assumptions and limitations of ReCaLL in Appendix B, and present the motivation for EM-MIA through an analysis of its sensitivity to prefix choice in Appendix C. To address the practical setting where neither labeled non-members nor reliable prompt effectiveness can be assumed, we propose EM-MIA, a fully unsupervised method that jointly estimates prefix effectiveness and membership likelihood using an expectation-maximization (EM) procedure.

Let $f(x)$ denote the membership score for each test example $x \in \mathcal{D}_{\text{test}}$, and $r(p)$ denote the effectiveness score of a prefix p. The key insight is that membership scores and prefix scores can reinforce each other: better membership estimates allow more accurate estimation of prefix effectiveness, and more reliable prefixes lead to improved membership predictions. This mutual dependency motivates an iterative procedure in which each set of scores is refined based on the other.

Algorithm 1 EM-MIA

Input: Target LLM $\mathcal{M}$, Test dataset $\mathcal{D}_{\text{test}}$
Output: Membership scores $f(x)$ for $x \in \mathcal{D}_{\text{test}}$
1: Initialize $f(x)$ with an existing off-the-shelf MIA method
2: **repeat**
3: Update prefix scores $r(p) = S(\text{ReCaLL}_p, f, \mathcal{D}_{\text{test}})$ for $p \in \mathcal{D}_{\text{test}}$
4: Update membership scores $f(x) = -r(x)$ for $x \in \mathcal{D}_{\text{test}}$
5: **until** Convergence (no significant difference in f)



Algorithm 1 outlines the overall procedure of EM-MIA. We initialize membership scores using any existing off-the-shelf MIA method such as Loss [20] or Min-K%++ [21] (Line 1). We then alternate between two updates: (1) estimating prefix scores $r(p)$ based on current membership scores $f(x)$ (Line 3), and (2) updating $f(x)$ using the refined $r(p)$ (Line 4). This process continues until convergence (Line 5). Because EM-MIA is a general framework, initialization, score update rules (see Appendix D), stopping criteria, and datasets (see Appendix E) can be adapted to different applications. Discussion on computational costs can be found in Appendix F.

3 OLMoMIA: New MIA Benchmark

To enable controlled and reproducible evaluation of MIA methods under varying difficulty levels, we introduce OLMoMIA, a new benchmark constructed from the training data and checkpoints of the OLMo-7B model [18], which was pre-trained on the Dolma dataset [22]. Unlike existing benchmarks such as WikiMIA [19], which rely on time-based heuristics, or MIMIR [16], which draws member and non-member examples from randomly partitioned subsets of the same data distribution, OLMoMIA allows systematic control over the distributional overlap between members and non-members. This allows evaluation under more realistic and ambiguous conditions, where membership inference is inherently more difficult. Details on benchmark construction are described in Appendix G.

4 Experiments and Results

We evaluate EM-MIA (whose configurations are described in Appendix H) and compare it with baseline methods (described in Appendix I and Appendix J) on WikiMIA and OLMoMIA using AUC-ROC as a main evaluation metric.[3] We also report TPR@1%FPR results in Appendix O. WikiMIA [19] provides length-based splits of 32, 64, and 128, and we follow prior work [17, 21] in using Mamba 1.4B [23], Pythia 6.9B [24], GPT-NeoX 20B [25], LLaMA 13B/30B [26], and OPT 66B [27] as target models. For OLMoMIA, we use all six controlled difficulty settings of *Easy*, *Medium*, *Hard*, *Random*, *Mix-1*, and *Mix-2*, and evaluate using OLMo-7B checkpoints after 100k, 200k, 300k, and 400k training steps. Ablation study on different initializations and scoring functions can be found in Section N.

Method	**Mamba-1.4B**			**Pythia-6.9B**			**LLaMA-13B**			**NeoX-20B**			**LLaMA-30B**			**OPT-66B**			**Average**		
	32	64	128	32	64	128	32	64	128	32	64	128	32	64	128	32	64	128	32	64	128
Loss	61.0	58.2	63.3	63.8	60.8	65.1	67.5	63.6	67.7	69.1	66.6	70.8	69.4	66.1	70.3	65.7	62.3	65.5	66.1	62.9	67.1
Ref	60.3	59.7	59.7	63.2	62.3	63.0	64.0	62.5	64.1	68.2	67.8	68.9	65.1	64.8	66.8	63.9	62.9	62.7	64.1	63.3	64.2
Zlib	61.9	60.4	65.6	64.3	62.6	67.6	67.8	65.3	69.7	69.3	68.1	72.4	69.8	67.4	71.8	65.8	63.9	67.4	66.5	64.6	69.1
Min-K%	63.3	61.7	66.7	66.3	65.0	69.5	66.8	66.0	71.5	72.1	72.1	75.7	69.3	68.4	73.7	67.5	66.5	70.6	67.5	66.6	71.3
Min-K%++	66.4	67.2	67.7	70.2	71.8	69.8	84.4	84.3	83.8	75.1	76.4	75.5	84.3	84.2	82.8	69.7	69.8	71.1	75.0	75.6	75.1
Avg	70.2	68.3	65.6	69.3	68.2	66.7	77.2	77.3	74.6	71.4	72.0	68.7	79.8	81.0	79.6	64.6	65.6	60.0	72.1	72.1	69.2
AvgP	64.0	61.8	56.7	62.1	61.0	59.0	63.1	60.3	56.4	63.9	61.8	61.1	60.3	60.0	55.4	86.9	94.3	95.1	66.7	66.5	63.9
RandM	25.4	25.1	26.2	24.9	26.2	24.6	21.0	14.9	68.6	25.3	28.3	29.8	14.0	15.1	70.4	33.9	40.9	42.9	24.1	25.1	43.8
Rand	72.7	78.2	64.2	67.0	73.4	68.7	73.9	75.4	68.5	68.2	74.5	67.5	66.9	71.7	70.2	64.5	67.8	58.6	68.9	73.5	66.3
RandNM	90.7	90.6	88.4	87.3	90.0	88.9	92.1	93.4	68.8	85.9	89.9	86.3	90.6	92.1	71.8	78.7	77.6	67.8	87.5	88.9	78.7
TopPref	90.6	91.2	88.0	91.3	92.9	90.1	93.5	94.2	71.8	88.4	92.0	90.2	92.9	93.8	74.8	83.6	79.6	72.1	90.0	90.6	81.2
ReCaLL [17]	90.2	91.4	91.2	91.6	93.0	92.6	92.2	95.2	92.5	90.5	93.2	91.7	90.7	94.9	91.2	85.1	79.9	81.0	90.1	91.3	90.0
EM-MIA	97.1	97.6	96.8	97.5	97.5	96.4	98.1	98.8	97.0	96.1	97.6	96.3	98.5	98.8	98.5	99.0	99.0	96.7	97.7	98.2	96.9

Table 1: AUC-ROC results on WikiMIA.

WikiMIA Table 1 and Table 3 show results on WikiMIA, using AUC-ROC and TPR@1%FPR as evaluation metrics, respectively. EM-MIA achieves state-of-the-art performance across all models and length splits, significantly outperforming all baselines, including ReCaLL, even without access to labeled non-member examples and exceeding 96% AUC-ROC in all cases. For the largest model, OPT-66B, it reaches over 99% AUC-ROC for length 32 and 64, whereas ReCaLL falls below 86%.

All non-ReCaLL baselines remain below 76% AUC-ROC on average. The performance order among ReCaLL-based variants is consistent: *RandM* < *Avg*, *AvgP* < *Rand* < *RandNM* < *TopPref*. This pattern confirms that ReCaLL is highly sensitive to the choice of prefix. Particularly, the significant performance gap between *Rand* and *RandNM* highlights ReCaLL's reliance on the availability of given non-members. Importantly, *Rand*, which uses no test labels, performs worse than Min-K%++ on average, indicating that ReCaLL alone is insufficient under a fully unsupervised setting.

RandNM is similar to the original ReCaLL [17] in most cases except for the OPT-66B model and LLaMA models with sequence length 128, probably because $n = 12$ is not optimal for these cases.

[3] Our implementation and datasets to reproduce our experiments are available at https://github.com/gyuwankim/em-mia

TopPref consistently outperforms *RandNM*, demonstrating that prefix quality varies and that random prefix selection is suboptimal. This opens the door to prefix optimization [28, 29, 30], though finding high-quality prefixes without supervision remains challenging. Our method approximates prefix quality without labels and uses it to improve membership prediction.

Method	Easy		Medium		Hard		Random		Mix-1		Mix-2	
	64	128	64	128	64	128	64	128	64	128	64	128
Loss	32.5	63.3	58.9	49.0	43.3	51.5	51.2	52.3	65.7	49.0	30.8	54.7
Ref	56.8	26.8	61.4	47.2	49.1	50.7	49.7	49.9	59.9	49.7	38.9	50.9
Zlib	24.0	51.8	44.8	50.7	40.5	51.1	52.3	50.5	63.2	47.2	31.5	54.3
Min-K%	32.4	50.0	54.0	51.9	43.0	51.2	51.7	51.0	60.8	50.4	34.9	51.7
Min-K%++	45.2	59.4	56.4	45.7	46.4	51.4	51.0	51.9	57.9	50.0	39.8	53.2
Avg	61.9	53.9	52.3	57.0	47.6	51.5	50.3	48.6	63.3	56.4	35.5	44.4
AvgP	79.2	39.9	53.9	61.7	50.2	51.4	49.0	50.1	55.7	63.0	42.7	41.8
RandM	32.3	22.7	39.2	30.3	45.8	50.5	48.1	48.2	49.7	48.0	29.1	28.7
Rand	63.7	46.3	56.0	59.4	48.9	52.1	49.7	49.1	60.6	68.0	38.0	38.6
RandNM	87.1	75.5	71.8	81.2	50.5	53.2	50.4	50.0	66.5	73.7	49.1	48.0
TopPref	88.9	88.5	79.7	64.4	55.7	54.5	52.3	52.7	79.9	80.2	55.3	62.1
EM-MIA	99.8	97.4	98.3	99.8	47.2	50.2	51.4	50.9	88.3	80.8	88.4	77.1

Table 2: AUC-ROC results on OLMoMIA.

OLMoMIA Table 2 and Table 4 show results on OLMoMIA, using AUC-ROC and TPR@1%FPR as evaluation metrics respectively. EM-MIA performs nearly perfectly on *Easy* and *Medium*, similar to its performance on WikiMIA. We did not observe consistent differences across checkpoints, despite the expectation that earlier training data would be harder to detect. Therefore, we report averages across four OLMo checkpoints. In contrast, it performs close to random guessing on *Hard* and *Random* similar to MIMIR, where member and non-member distributions heavily overlap and all methods are not sufficiently better than random guessing. On *Mix-1* and *Mix-2*, EM-MIA achieves reasonable scores, though not as high as in easier settings. In all but the hardest scenarios, EM-MIA significantly outperforms all baselines.

None of the baselines without ReCaLL-based approaches are successful in all settings, which implies that OLMoMIA is a challenging benchmark. The relative order between ReCaLL-based baselines is again consistent: *RandM* < *Avg*, *AvgP*, *Rand* < *RandNM* < *TopPref*, although none of the fully unsupervised variants are successful overall. Interestingly, *RandNM* works reasonably well on *Mix-1* but does not work well on *Mix-2*. This is likely because non-members from *Mix-1* are from the same cluster while non-members from *Mix-1* are randomly sampled from the entire distribution. *TopPref* again outperforms *RandNM*, reinforcing that not all non-members are equally effective as prompts.

5 Conclusion

We propose EM-MIA, a membership inference method for large language models that jointly estimates membership scores and prompt effectiveness through an expectation-maximization procedure. Unlike prior work that relies on labeled non-members or assumes prompt quality in advance, EM-MIA operates in a fully unsupervised gray-box setting, making it suitable for more realistic deployment scenarios. Our method outperforms ReCaLL, even without its strong assumptions, and achieves state-of-the-art results on WikiMIA. EM-MIA is modular and flexible, allowing different initialization strategies, scoring rules, and convergence criteria depending on the application context.

To support more rigorous and controlled evaluation, we introduce OLMoMIA, a new benchmark built from the OLMo pretraining pipeline that allows fine-grained control over distributional overlap between members and non-members. Through comprehensive experiments, we show that EM-MIA is robust across a wide range of difficulty settings, while also identifying scenarios where all existing methods struggle, particularly when member and non-member distributions are nearly identical. Our findings highlight the importance of evaluating MIA methods under diverse and ambiguous conditions, and suggest that future progress will require methods that adapt to both prompt variability and fine-grained data overlap.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[2] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[3] Peter Henderson, Xuechen Li, Dan Jurafsky, Tatsunori Hashimoto, Mark A Lemley, and Percy Liang. Foundation models and fair use. *Journal of Machine Learning Research*, 24(400):1–79, 2023.

[4] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.

[5] Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. Beyond memorization: Violating privacy via inference with large language models. *arXiv preprint arXiv:2310.07298*, 2023.

[6] Nikhil Kandpal, Krishna Pillutla, Alina Oprea, Peter Kairouz, Christopher A Choquette-Choo, and Zheng Xu. User inference attacks on large language models. *arXiv preprint arXiv:2310.09266*, 2023.

[7] Matthieu Meeus, Igor Shilov, Manuel Faysse, and Yves-Alexandre de Montjoye. Copyright traps for large language models. *arXiv preprint arXiv:2402.09363*, 2024.

[8] Kent K Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. Speak, memory: An archaeology of books known to chatgpt/gpt-4. *arXiv preprint arXiv:2305.00118*, 2023.

[9] 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.

[10] 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.

[11] Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. *arXiv preprint arXiv:2203.08242*, 2022.

[12] Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. *arXiv preprint arXiv:2310.18018*, 2023.

[13] Oscar Sainz, Iker García-Ferrero, Alon Jacovi, Jon Ander Campos, Yanai Elazar, Eneko Agirre, Yoav Goldberg, Wei-Lin Chen, Jenny Chim, Leshem Choshen, et al. Data contamination report from the 2024 conda shared task. *arXiv preprint arXiv:2407.21530*, 2024.

[14] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[15] California State Legislature. California consumer privacy act (ccpa). `https://oag.ca.gov/privacy/ccpa`, 2018.

[16] 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 *Conference on Language Modeling (COLM)*, 2024.

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

[18] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

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

[20] 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.

[21] Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Yang, and Hai Li. Min-k%++: Improved baseline for detecting pre-training data from large language models. *arXiv preprint arXiv:2404.02936*, 2024.

[22] Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024.

[23] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[24] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[25] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

[26] 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.

[27] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[28] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

[29] Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P Xing, and Zhiting Hu. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*, 2022.

[30] Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*, 2023.

[31] 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.

[32] Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR, 2022.

[33] Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 36, 2024.

[34] Igor Shilov, Matthieu Meeus, and Yves-Alexandre de Montjoye. Mosaic memory: Fuzzy duplication in copyright traps for large language models. *arXiv preprint arXiv:2405.15523*, 2024.

[35] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*, 2023.

[36] Hamid Mozaffari and Virendra J Marathe. Semantic membership inference attack against large language models. *arXiv preprint arXiv:2406.10218*, 2024.

[37] 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.

[38] 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.

[39] 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 *33rd USENIX Security Symposium (USENIX Security 24)*, pages 2369–2385, 2024.

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

[41] Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. Inherent challenges of post-hoc membership inference for large language models. *arXiv preprint arXiv:2406.17975*, 2024.

[42] Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. Llm dataset inference: Did you train on my dataset? *arXiv preprint arXiv:2406.06443*, 2024.

[43] Cédric Eichler, Nathan Champeil, Nicolas Anciaux, Alexandra Bensamoun, Heber Hwang Arcolezi, and José Maria De Fuentes. Nob-mias: Non-biased membership inference attacks assessment on large language models with ex-post dataset construction. *arXiv preprint arXiv:2408.05968*, 2024.

[44] Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv preprint arXiv:2211.04325*, 2022.

[45] Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[46] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[47] Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 1961.

[48] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[49] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*, 2024.

[50] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022.

[51] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[52] Jonathan Tow. Stablelm alpha v2 models, 2023.

[53] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[54] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[55] 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.

[56] Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. Membership inference attack susceptibility of clinical language models. *arXiv preprint arXiv:2104.08305*, 2021.

[57] Saeed Mahloujifar, Huseyin A Inan, Melissa Chase, Esha Ghosh, and Marcello Hasegawa. Membership inference on word embedding and beyond. *arXiv preprint arXiv:2106.11384*, 2021.

[58] Virat Shejwalkar, Huseyin A Inan, Amir Houmansadr, and Robert Sim. Membership inference attacks against nlp classification models. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[59] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. *arXiv preprint arXiv:2203.03929*, 2022.

[60] Shangqing Tu, Kejian Zhu, Yushi Bai, Zijun Yao, Lei Hou, and Juanzi Li. Dice: Detecting in-distribution contamination in llm's fine-tuning phase for math reasoning. *arXiv preprint arXiv:2406.04197*, 2024.

[61] Qizhang Feng, Siva Rajesh Kasa, Hyokun Yun, Choon Hui Teo, and Sravan Babu Bodapati. Exposing privacy gaps: Membership inference attack on preference data for llm alignment. *arXiv preprint arXiv:2407.06443*, 2024.

[62] Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, and Ge Li. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. *arXiv preprint arXiv:2402.15938*, 2024.

[63] Baohang Zhou, Zezhong Wang, Lingzhi Wang, Hongru Wang, Ying Zhang, Kehui Song, Xuhui Sui, and Kam-Fai Wong. Dpdllm: A black-box framework for detecting pre-training data from large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 644–653, 2024.

[64] Masahiro Kaneko, Youmi Ma, Yuki Wata, and Naoaki Okazaki. Sampling-based pseudo-likelihood for membership inference attacks. *arXiv preprint arXiv:2404.11262*, 2024.

A Related Work

Membership Inference on LLMs. Membership inference on LLMs presents unique challenges. First, LLMs are trained on massive corpora, and individual examples are typically seen only once or a few times [31], leaving minimal memorization footprint. Second, defining membership is inherently ambiguous in natural language, in that texts often repeat or partially overlap even after rigorous decontamination [32, 33], and paraphrased or semantically similar content can blur membership boundaries [34, 35, 36]. Traditional MIA methods often rely on training shadow models using labeled data from a similar distribution [9], but this is impractical in LLM settings due to limited access to comparable data and training specifications.

In contrast, MIA methods for LLMs typically use the model's loss (e.g., negative log-likelihood) as a membership score, under the assumption that models tend to memorize or overfit their training data [20, 10]. Building on this idea, several techniques calibrate membership scores based on input difficulty [37], using reference models [10], compression-based heuristics [38], or nearest neighbors in embedding space [35]. Other methods focus on low-likelihood tokens [19] or compute calibrated token-level ratios [21].

ReCALL [17] proposes a different strategy by using known non-member examples as prompts to condition the model's response. It assumes that such prompts suppress memorization signals, enabling members to stand out by their elevated likelihood under the same prompt. However, this assumption is brittle, as prompt effectiveness varies significantly across examples, and a fixed prompt often fails to generalize across models or domains. We address this limitation by proposing a fully unsupervised method that jointly estimates prompt effectiveness and membership likelihood, without relying on labeled non-members or fixed prompting strategies.

Evaluation Benchmarks. Robust evaluation of MIA methods for LLMs remains challenging because existing benchmarks rarely provide both reliable membership labels and controllable distributional settings. Most benchmarks fall into one of two categories. Some, such as WikiMIA [19, 39], determine membership based on document timestamps and model release dates. This approach risks conflating membership inference with distribution shift detection [40, 41, 42]. Others, such as MIMIR [16], use random splits to ensure that member and non-member distributions are nearly identical. In such cases, no existing method performs significantly better than random guessing.

These limitations make it difficult to understand how well a method generalizes across different data conditions. Pre-training corpora are typically drawn from diverse sources, while inference-time inputs may come from entirely different domains. Effective evaluation therefore requires testing under a range of membership separability conditions. However, constructing such benchmarks is practically difficult, especially given the lack of true non-member data and the challenge of controlling test distributions. There is a clear need for evaluation setups that reflect varied, realistic scenarios while maintaining access to reliable ground-truth labels [41, 43].

B Assumptions and Limitations of ReCaLL

ReCaLL [17] is a prompt-based membership inference method that computes the ratio between the conditional and unconditional log-likelihoods of a target example x under $\mathcal{M}$. Given a prefix p, the ReCaLL score is defined as $\mathrm{ReCaLL}_p(x; \mathcal{M}) = \mathrm{LL}(x \mid p; \mathcal{M})/\mathrm{LL}(x; \mathcal{M})$, where LL denotes the average log-likelihood over tokens, and $p = p_1 \oplus \cdots \oplus p_n$ is a concatenation of non-member examples p_i. The intuition is that conditioning on non-members tends to reduce the likelihood of members more than that of non-members, making the ratio indicative for membership prediction.

ReCaLL demonstrates strong empirical performance, achieving over 90% AUC-ROC on WikiMIA [19] and outperforming prior methods such as Min-K%++ [21]. However, this performance depends on strong assumptions and lacks theoretical justification. In its original implementation, ReCaLL constructs prefixes by randomly selecting non-members from the test set, assuming that (1) ground-truth non-members are available at inference time, and (2) all non-members are equally effective as prompts.

In practice, such assumptions rarely hold so labeled non-members are often unavailable, especially when the training and test data distributions substantially overlap [44, 45]. Even synthetic prefixes generated using GPT-4, as explored in [17], rely on seed non-members drawn from the test distribution.

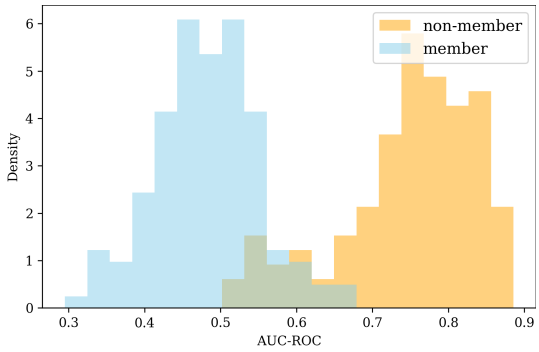


Figure 1: Distribution of prefix scores (measured by AUC-ROC in the oracle setting) for members and non-members on WikiMIA [19] (length 128) using Pythia-6.9B [24].

This reliance on known non-members gives ReCaLL an unfair advantage over methods that operate without access to test labels.

Ablation studies in [17] further show that ReCaLL's performance degrades when the prefix and test inputs differ in distribution, and that different random samples yield significant variance in accuracy. These findings suggest that non-members vary widely in their effectiveness as prompts, and that ReCaLL does not generalize reliably across domains or distribution shifts. These limitations motivate the need for a more flexible and fully unsupervised approach that does not depend on labeled non-members or assume prompt effectiveness in advance.

C Observation on Sensitivity to Prefix Choice

We empirically examine how ReCaLL's performance varies with the choice of prefix, particularly when labeled non-members are unavailable. To this end, we define a *prefix score* $r(p)$ as the effectiveness of a prefix p in distinguishing members from non-members when used in ReCaLL.

In an oracle setting with access to ground-truth membership labels, we compute $r(p)$ as the AUC-ROC of $\mathrm{ReCaLL}_p(x)$ over a test set $\mathcal{D}_{\text{test}}$, using each $x \in \mathcal{D}_{\text{test}}$ as a standalone prefix. This allows us to empirically measure the effectiveness of each test example when used as a prefix.

Figure 1 shows that non-member prefixes generally lead to strong ReCaLL performance, with AUC-ROC often exceeding 0.7. In contrast, member prefixes perform poorly, with scores clustering near 0.5 (i.e., random guessing). Additional comparisons using alternative metrics for prefix scoring are included in Appendix L. These results highlight two limitations of current ReCaLL-based methods: (1) Even among non-members, prefix effectiveness varies widely; (2) In realistic scenarios, ground-truth labels needed to evaluate or filter prefixes are unavailable.

These findings underscore the need for an approach that can identify effective prefixes and infer membership without access to labels. We address this challenge in the following section by proposing a fully unsupervised method that jointly estimates membership likelihood and prefix effectiveness through iterative refinement.

D Score Update Rules

Updating Prefix Scores. AUC-ROC is an effective function S for evaluating a prefix p in the oracle setting given ground truth labels. Since ground-truth labels are not available, we generate pseudo-labels using a threshold τ over current membership scores $f(x)$ and use them to calculate prefix scores: $\text{AUC-ROC}(\{(\mathrm{ReCaLL}_p(x), \mathbf{1}_{f(x)>\tau}) \mid x \in \mathcal{D}_{\text{test}}\})$. We typically set τ to the median of $f(x)$, assuming a balanced dataset. Alternatively, instead of relying on hard thresholds, we can measure rank alignment between $\mathrm{ReCaLL}_p(x)$ and $f(x)$ using the average absolute rank difference or rank correlation coefficients such as Kendall's tau [46] or Spearman's rho [47].

Updating Membership Scores. A negative prefix score $-r(x)$ is a simple yet effective membership score. Alternatively, one could construct a prefix $p = p_1 \oplus \cdots \oplus p_n$ using top-k examples ranked by $r(x)$, and compute $f(x) = \mathrm{ReCaLL}_p(x)$ using this prefix. The ordering of p_i within p is also a design choice. Placing stronger prefixes closer to x may amplify their influence due to LLMs' attention bias toward recent tokens.

E Using External Data

We may extend the test dataset $\mathcal{D}_{\mathrm{test}}$ by utilizing external data to provide additional signals. Suppose we have a dataset of known members ($\mathcal{D}_{\mathrm{m}}$), a dataset of known non-members ($\mathcal{D}_{\mathrm{nm}}$), and a dataset of instances without any membership information ($\mathcal{D}_{\mathrm{unk}}$). For example, $\mathcal{D}_{\mathrm{m}}$ could be old Wikipedia documents, sharing the common assumption that LLMs are usually trained with Wikipedia. As discussed above, we target the case of $\mathcal{D}_{\mathrm{nm}} = \phi$, or at least $\mathcal{D}_{\mathrm{nm}} \cap \mathcal{D}_{\mathrm{test}} = \phi$. However, we can construct it with completely unnatural texts (e.g., "*b9qx84;5zln"). $\mathcal{D}_{\mathrm{unk}}$ is desirably drawn from the same distribution of $\mathcal{D}_{\mathrm{test}}$ but could be from any corpus when we do not know the test dataset distribution. Finally, we can incorporate all available data for better prediction of membership scores and prefix scores: $\mathcal{D}_{\mathrm{test}} \leftarrow \mathcal{D}_{\mathrm{test}} \cup \mathcal{D}_m \cup \mathcal{D}_{\mathrm{nm}} \cup \mathcal{D}_{\mathrm{unk}}$.

F Computational Costs

MIAs for LLMs only do inference without any additional training, so they are usually not too expensive. Therefore, MIA accuracy is typically prioritized over computational costs as long as it is reasonably feasible. Nevertheless, maintaining MIAs' computational costs within a reasonable range is important. Computations on all our experiments with the used datasets (WikiMIA and OLMoMIA) were manageable even in an academic setting. We compare computational complexity between EM-MIA and other baselines (mainly, ReCaLL) and describe how computational costs of EM-MIA can be further reduced below.

EM-MIA is a general framework in that the update rules for prefix scores and membership scores can be designed differently (as described in §2), and they determine the trade-off between MIA accuracy and computational costs. For the design choice described in Algorithm 1 that was used in our experiments, EM-MIA requires a pairwise computation $LL_p(x)$ for all pairs (x, p) once, where $x, p \in \mathcal{D}_{\mathrm{test}}$. These values are reused to calculate the prefix scores in each iteration without recomputation. The iterative process does not require additional LLM inferences. The time complexity of EM-MIA is $O(D^2L^2)$, where $D = |\mathcal{D}_{\mathrm{test}}|$ and L is an average token length of each data on $\mathcal{D}_{\mathrm{test}}$, by assuming LLM inference cost is quadratic to the input sequence length due to the Transformer architecture. In this case, EM-MIA does not have other tuning hyperparameters, while Min-K% and Min-K%++ have K and or ReCaLL has n. This is more reasonable since validation data to tune them is not given.

Of course, the baselines other than ReCaLL (Loss, Ref, Zlib, Min-K%, and Min-K%++) only compute a log-likelihood of each target text without computing a conditional log-likelihood with a prefix, so they are the most efficient: $O(DL^2)$ time complexity. Since ReCaLL uses a long prefix consisting of n non-member data points, its time complexity is $O(D(nL)^2) = O(n^2DL^2)$. According to the ReCaLL paper, they sweep n from 1 to 12 to find the best n, which means $O((1^2 + 2^2 + \cdots + n^2)DL^2) = O(n^3DL^2)$. Also, in some cases (Figure 3 and Table 7 in their paper), they used $n = 28$ to achieve a better result. In theory, it may seem EM-MIA does not scale well with respect to D. Nevertheless, the amount of computation and time for EM-MIA with $D \sim 1000$ is not significantly larger than ReCaLL, considering the n factor.

Moreover, ReCaLL requires $O(n^2)$ times larger memory than others including EM-MIA, so it may not be feasible for hardware with a small memory. In this sense, EM-MIA is more parallelizable, and we make EM-MIA faster with batching. Lastly, there is room to improve the time complexity of our method. We have not explored this yet, but for example, we may compute ReCaLL scores on a subset of the test dataset to calculate prefix scores as an approximation of our algorithm. We left improving the efficiency of EM-MIA as future work.

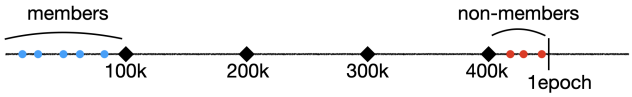


Figure 2: The basic setup of OLMoMIA benchmark. The horizontal line indicates a training step. For any intermediate checkpoint at a specific step, we can consider training data before and after that step as members and non-members, respectively.

G Details on OLMoMIA Benchmark Construction

Membership Label Assignment. Figure 2 illustrates the benchmark setup. OLMo provides intermediate model checkpoints and a detailed index mapping training steps to data examples, offering a rare opportunity to precisely define membership. We use four OLMo-7B checkpoints saved at 100k, 200k, 300k, and 400k training steps, where one full epoch consists of just over 450k steps. We define member examples as those seen before step 100k and non-members as those introduced between steps 400k and 500k. This setup reflects a practical incremental training scenario. Some ambiguity in membership may remain despite deduplication, as discussed in Section A.

Dataset Sampling with Varying Difficulty We construct six dataset variants to simulate different levels of distributional overlap. The basic *Random* setting samples member and non-member examples uniformly from their respective intervals. This is analogous to MIMIR [16], which is known to be more challenging than WikiMIA due to minimal distributional differences between members and non-members [48].

To introduce controlled variation in difficulty, we first embed the candidate examples using NV-Embed-v2 [49], the top-performing model on the MTEB leaderboard [50] as of August 2024. We then perform K-means clustering [51] separately on member and non-member embeddings with $K = 50$. To ensure diversity within clusters, we apply greedy deduplication by removing examples that are too similar (cosine distance below 0.6) to other points in the same cluster.

Based on these clusters, we define three difficulty-controlled variants: *Easy* selects the most dissimilar member and non-member clusters and samples examples furthest from the opposing group; *Hard* selects the most similar clusters and samples examples closest to the opposing group; *Medium* selects clusters with median inter-cluster distance and samples randomly from each.

We additionally define two hybrid settings: *Mix-1* combines members from *Random* and non-members from *Hard*, simulating tightly clustered test-time distributions; *Mix-2* does the reverse, combining members from *Hard* and non-members from *Random*. Together, these configurations span a broad range of separability conditions, providing a robust testbed for evaluating MIA methods. Formal definitions of each construction step are included in Appendix M.

Dataset Specifications. Each difficulty variant includes two subsets with maximum sequence lengths of 64 and 128 tokens. Each subset contains 500 members and 500 non-members, for a total of 1,000 examples per dataset.

H EM-MIA Configurations for Experiments

As described in Section 2, EM-MIA is a general framework where each component can be tuned for improvement, but we use the following options as defaults based on results from preliminary experiments. Overall, Min-K%++ performs best among baselines without ReCaLL-based approaches, so we use it as a default choice for initialization. Alternatively, we may use ReCaLL-based methods that do not rely on any labels like *Avg*, *AvgP*, or *Rand*. For the update rule for prefix scores, we use AUC-ROC as a default scoring function S. For the update rule for membership scores, we use negative prefix scores as new membership scores. For the stopping criterion, we repeat ten iterations and stop without thresholding by the score difference since we observed that membership scores and prefix scores converge quickly after a few iterations. We also observed that EM-MIA is not sensitive to the choice of the initialization method and the scoring function S and converges to similar results.

I Baselines

We compare EM-MIA against the following baselines: Loss [20], Ref [10], Zlib [38], Min-K% [19], and Min-K%++ [21]. We use Pythia-70m for WikiMIA and StableLM-Base-Alpha-3B-v2 model [52] for OLMoMIA as the reference model of the Ref method, following [19] and [16]. We use $K = 20$ for Min-K% and Min-K%++. Among the commonly used baselines, we omit Neighbor [35] because it is not the best in most cases though it requires LLM inference multiple times for neighborhood texts, so it is much more expensive.

J ReCaLL-based Baselines

We include several variants of ReCaLL that differ in how the prefix $p = p_1 \oplus \cdots \oplus p_n$ is constructed: *Rand*, *RandM*, *RandNM*, and *TopPref*. *Rand* randomly selects any data from $\mathcal{D}_{\text{test}}$. *RandM* randomly selects member data from $\mathcal{D}_{\text{test}}$. *RandNM* randomly selects non-member data from $\mathcal{D}_{\text{test}}$. *TopPref* selects data from $\mathcal{D}_{\text{test}}$ with the highest prefix scores calculated with ground truth labels the same as §C.

Among these, only *Rand*is fully unsupervised; the others either partially or fully rely on labels in the test dataset, making them unsuitable for realistic scenarios. For all methods using a random selection (*Rand*, *RandM*, and, *RandNM*), we execute five times with different random seeds and report the average. We fix $n = 12$ since it provides a reasonable performance while not too expensive. We report the results from the original ReCaLL paper but explain why this is not a fair comparison in Appendix K.

We also evaluate two unsupervised averaging variants. *Avg* and *AvgP* average ReCaLL scores over all data points in $\mathcal{D}_{\text{test}}$: $\textit{Avg}(x) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{p \in \mathcal{D}_{\text{test}}} \text{ReCaLL}_p(x)$ and $\textit{AvgP}(p) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{x \in \mathcal{D}_{\text{test}}} \text{ReCaLL}_p(x)$. The intuition is averaging will smooth out ReCaLL scores with a non-discriminative prefix while keeping ReCaLL scores with a discriminative prefix without exactly knowing discriminative prefixes.

K Comparison with ReCaLL

As explained in §B, the original ReCaLL [17] uses labeled data from the test dataset, which is unfair to compare with the above baselines and ours. More precisely, p_i in the prefix $p = p_1 \oplus p_2 \oplus \cdots \oplus p_n$ are known non-members from the test set $\mathcal{D}_{\text{test}}$, and they are excluded from the test dataset for evaluation, i.e., $\mathcal{D}_{\text{test}}{}' = \mathcal{D}_{\text{test}} \setminus \{p_1, p_2, \cdots, p_n\}$. However, we measure the performance of ReCaLL with different prefix selection methods to understand how ReCaLL is sensitive to the prefix choice and use it as a reference instead of a direct fair comparison.

Since changing the test dataset every time for different prefixes does not make sense and makes the comparison even more complicated, we keep them in the test dataset. A language model tends to repeat, so $\text{LL}(p_i|p; \mathcal{M}) \simeq 0$. Because $\text{LL}(p_i|p; \mathcal{M}) \ll 0$, $\text{ReCaLL}_p(p_i; \mathcal{M}) \simeq 0$. It is likely to $\text{ReCaLL}_p(p_i; \mathcal{M}) \ll \text{ReCaLL}_p(x; \mathcal{M})$ for $x \in \mathcal{D}_{\text{test}} \setminus \{p_1, p_2, \cdots, p_n\}$, meaning that ReCaLL will classify p_i as a non-member. The effect would be marginal if $|\mathcal{D}_{\text{test}}| \gg n$. Otherwise, we should consider this when we read numbers in the result table.

The original ReCaLL [17] is similar to *RandNM*, except they report the best score after trying all different n values, which is again unfair. The number of shots n is an important hyper-parameter determining performance. A larger n generally leads to a better MIA performance but increases computational cost with a longer p.

L Metrics for Prefix Scores

Figure 3 shows ROC curves when negative prefix scores, computed using different metrics, are used directly as membership scores. We compare prefix scoring metrics including AUC-ROC, Accuracy, and TPR@k%FPR for $k \in \{0.1, 1, 5, 10, 20\}$. Among them, using AUC-ROC to compute prefix scores yields the best result, achieving 98.6% AUC-ROC for membership inference.

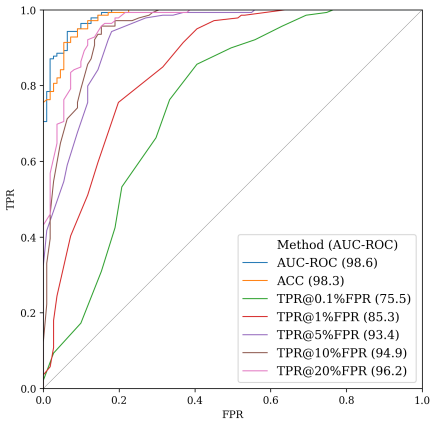


Figure 3: ROC curves for MIA using the negative prefix score as the membership score, evaluated with different metrics for prefix scores in the oracle setting on WikiMIA [19] (length 128) using Pythia-6.9B [24].

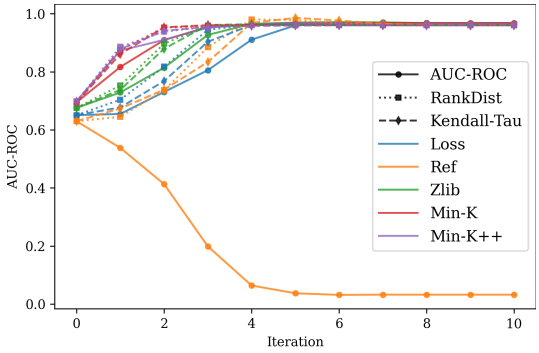


Figure 4: Performance of EM-MIA for each iteration with varying baselines for initialization and scoring functions S on WikiMIA [19] (length 128) using Pythia-6.9B [24].

M Formulation of OLMoMIA Settings

After the filtering of removing close points, let member clusters as C_i^m for $i \in [1, K]$ and non-member clusters as C_j^{nm} for $j \in [1, K]$. These clusters satisfy $d(x, y) > 0.6$ for all $x, y \in C_i^m$ and $d(x, y) > 0.6$ for all $x, y \in C_j^{nm}$. The following equations formalize how we construct different settings of OLMoMIA:

- *Random*: $\mathcal{D}_{\text{random}} = \mathcal{D}^{\text{m}}_{\text{random}} \cup \mathcal{D}^{\text{nm}}_{\text{random}}$
- *Easy*: $\mathcal{D}_{\text{easy}} = \mathcal{D}^{\text{m}}_{\text{easy}} \cup \mathcal{D}^{\text{nm}}_{\text{easy}}$, where $i_{easy}, j_{easy} = \arg\max_{(i,j)} \mathbb{E}_{x \in C_i, y \in C_j} d(x, y)$, $\mathcal{D}^{\text{m}}_{\text{easy}} = \arg\text{topk}_x \, \mathbb{E}_{y \in C^{nm}_{j_{easy}}} d(x, y)$, and $\mathcal{D}^{\text{nm}}_{\text{easy}} = \arg\text{topk}_y \, \mathbb{E}_{x \in C^{m}_{i_{easy}}} d(x, y)$
- *Hard*: $\mathcal{D}_{\text{hard}} = \mathcal{D}^{\text{m}}_{\text{hard}} \cup \mathcal{D}^{\text{nm}}_{\text{hard}}$, where $i_{hard}, j_{hard} = \arg\min_{(i,j)} \mathbb{E}_{x \in C_i, y \in C_j} d(x, y)$, $\mathcal{D}^{\text{m}}_{\text{hard}} = \arg\text{topk}_x \, -\mathbb{E}_{y \in C^{nm}_{j_{hard}}} d(x, y)$, and $\mathcal{D}^{\text{nm}}_{\text{hard}} = \arg\text{topk}_y \, -\mathbb{E}_{x \in C^{m}_{i_{hard}}} d(x, y)$
- *Medium*: $\mathcal{D}_{\text{medium}} = \mathcal{D}^{\text{m}}_{\text{medium}} \cup \mathcal{D}^{\text{nm}}_{\text{medium}}$, where $i_{medium}, j_{medium} = \text{median}_{(i,j)} \, \mathbb{E}_{x \in C_i, y \in C_j} d(x, y)$, $\mathcal{D}^{\text{m}}_{\text{medium}} \subset C^m_{i_{medium}}$, and $\mathcal{D}^{\text{nm}}_{\text{medium}} \subset C^{nm}_{j_{medium}}$
- *Mix-1*: $\mathcal{D}_{\text{mix}-1} = \mathcal{D}^{\text{m}}_{\text{random}} \cup \mathcal{D}^{\text{nm}}_{\text{hard}}$
- *Mix-2*: $\mathcal{D}_{\text{mix}-2} = \mathcal{D}^{\text{m}}_{\text{hard}} \cup \mathcal{D}^{\text{nm}}_{\text{random}}$

N Ablation Study on Initializations and Scoring Functions

Figure 4 displays the ablation study of EM-MIA with different combinations of the initialization (Loss, Ref, Zlib, Min-K%, and Min-K%++) and the scoring function S (AUC-ROC, RankDist, and Kendall-Tau) using the WikiMIA dataset with a length of 128 and Pythia-6.9B model. Each

curve indicates the change of AUC-ROC calculated from the estimates of membership scores at each iteration during the expectation-maximization algorithm. In most combinations, the algorithm converges to a similar accuracy after 4-5 iterations. In this figure, there is only one case in which AUC-ROC decreases quickly and reaches a value close to 0. It is difficult to know when this happens, but it predicts members and non-members oppositely, meaning that using negative membership scores gives a good AUC-ROC.

O TPR@1%FPR Results

Method	**Mamba-1.4B**			**Pythia-6.9B**			**LLaMA-13B**			**NeoX-20B**			**LLaMA-30B**			**OPT-66B**			**Average**		
	32	64	128	32	64	128	32	64	128	32	64	128	32	64	128	32	64	128	32	64	128
Loss	4.7	2.1	1.4	6.2	2.8	3.6	4.7	4.2	7.9	10.3	3.5	4.3	4.1	5.3	7.2	6.5	3.5	3.6	6.1	3.6	4.7
Ref	0.5	0.7	0.7	1.6	1.1	1.4	2.3	3.9	2.9	3.1	2.5	1.4	1.3	2.5	3.6	1.8	1.8	0.7	1.8	2.1	1.8
Zlib	4.1	4.9	7.2	4.9	6.0	11.5	5.7	8.1	12.9	9.3	6.3	5.0	4.9	9.5	10.1	5.7	7.0	11.5	5.8	7.0	9.7
Min-K%	7.0	4.2	5.8	8.8	3.9	7.2	5.2	6.0	15.1	10.6	3.9	7.2	4.7	7.0	5.8	9.0	7.7	8.6	7.5	5.5	8.3
Min-K%++	4.1	7.0	1.4	5.9	10.6	10.1	10.3	12.0	25.2	6.2	9.5	1.4	8.3	6.7	9.4	3.6	12.0	13.7	6.4	9.6	10.2
Avg	3.9	0.4	5.0	8.0	1.1	7.9	3.1	7.0	6.5	6.2	2.1	8.6	2.8	6.7	8.6	2.6	2.1	4.3	4.4	3.2	6.8
AvgP	0.5	0.4	0.7	1.8	0.4	0.0	0.0	0.7	0.0	1.3	0.7	0.0	0.0	0.0	2.9	2.1	12.3	24.5	0.9	2.4	4.7
RandM	0.8	0.1	0.6	0.9	0.0	1.9	0.2	0.4	7.6	0.5	0.3	1.6	0.4	0.6	8.1	0.7	0.1	0.9	0.6	0.2	3.4
Rand	3.7	3.9	2.4	2.3	3.2	7.6	1.6	2.7	7.3	4.4	5.0	4.7	1.6	3.2	7.9	2.1	3.2	3.2	2.6	3.5	5.5
RandNM	19.2	8.3	15.4	12.6	10.5	18.7	18.5	17.2	7.5	12.9	11.6	12.5	13.8	18.7	8.1	5.0	5.0	6.6	13.7	11.9	11.5
TopPref	12.7	4.2	25.2	16.0	1.4	29.5	14.2	9.2	7.9	13.4	13.7	20.9	27.1	29.9	8.6	3.9	5.6	9.4	14.6	10.7	16.9
ReCaLL [17]	11.2	11.0	4.0	28.5	20.7	33.3	13.3	30.1	26.3	25.3	6.9	30.3	18.4	18.3	1.0	8.3	5.3	6.1	17.5	15.4	16.9
EM-MIA	54.0	47.9	51.8	50.4	56.0	47.5	66.4	75.7	58.3	51.4	64.1	59.0	61.5	66.2	71.9	83.5	73.2	39.6	61.2	63.8	54.7

Table 3: TPR@1%FPR results on WikiMIA benchmark. The second block (grey) is ReCaLL-based baselines. *RandM*, *RandNM*, ReCaLL, and *TopPref* use labels in the test dataset, so comparing them with others is unfair. We report their scores for reference. We borrow the original ReCaLL results from [17] which is also unfair to be compared with ours and other baselines.

Method	**Easy**		**Medium**		**Hard**		**Random**		**Mix-1**		**Mix-2**	
	64	128	64	128	64	128	64	128	64	128	64	128
Loss	2.8	12.8	7.2	1.4	0.1	1.2	1.3	0.7	7.2	1.7	0.0	0.7
Ref	6.2	4.0	4.9	0.6	1.0	0.9	1.2	1.2	8.4	0.5	0.2	1.6
Zlib	2.0	9.8	6.7	1.1	0.2	1.6	0.9	0.7	6.4	1.7	0.0	0.7
Min-K%	1.3	6.5	5.8	1.4	0.1	1.3	1.1	0.7	6.1	2.0	0.0	0.7
Min-K%++	1.4	8.0	5.0	0.7	0.4	1.0	1.0	0.4	5.0	0.9	0.0	0.5
Avg	4.1	11.5	4.0	1.7	0.2	2.2	1.2	0.6	6.1	2.2	0.0	0.9
AvgP	11.7	0.1	2.6	7.2	0.7	1.6	0.7	1.4	4.8	12.1	0.1	0.0
RandM	3.0	4.9	2.4	1.1	0.4	2.2	0.9	0.8	7.6	1.3	0.0	0.4
Rand	4.3	7.8	3.7	1.7	0.4	2.7	1.0	0.8	10.6	3.0	0.0	0.7
RandNM	16.9	14.2	5.2	1.8	0.3	1.9	1.0	0.8	9.2	2.9	0.0	1.1
TopPref	22.0	16.6	6.3	1.9	0.4	2.2	1.1	1.4	8.1	5.1	0.0	0.5
EM-MIA	95.0	52.1	79.8	96.7	1.8	1.0	1.1	1.4	12.2	3.8	14.8	4.3

Table 4: TPR@1%FPR results on OLMoMIA benchmark. The second block (grey) is ReCaLL-based baselines. *RandM*, *RandNM*, ReCaLL, and *TopPref* use labels in the test dataset, so comparing them with others is unfair. We report their scores for reference.

TPR@low FPR is a useful MIA evaluation metric [10] in addition to AUC-ROC, especially when developing a new MIA and comparing it with other MIAs. Due to the space limitation in the main text, we put TPR@low FPR here: Table 3 for WikiMIA and Table 4 for OLMoMIA.

P Limitations

Our paper focuses on detecting LLMs' pre-training data with the gray-box access, where computing the probability of a text from output logits is possible. However, many proprietary LLMs are usually further fine-tuned [53, 54], and they only provide generation outputs, which is the black-box setting. We left the extension of our approach to MIAs for fine-tuned LLMs [55, 56, 57, 58, 59, 60, 61] or LLMs with black-box access [62, 63, 64] as future work.

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 abstract and introduction state the core contributions of EM-MIA, the setting and assumptions, and the scope of evaluation, and these match the algorithms and experiments reported in the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: We include a Limitations section.

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: The paper focuses on an algorithmic framework and empirical evaluation without formal theorems.

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: We document datasets, preprocessing, model versions, metrics, and all hyperparameters needed to reproduce the main tables. Step-by-step procedures and ablation settings are also provided in detail.

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: We will release a public repository upon acceptance. Instructions include the exact commands, environment, and data access notes.

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 specify evaluation setups and model configurations.

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: [No]

   Justification: [NA]

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: [No]

Justification: We provide complexity analysis.

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 followed the NeurIPS Code of Ethics, used publicly available or licensed datasets, and adhered to model provider terms for API use. Anonymity is preserved for submission materials.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Our work is beneficial to the trustworthy evaluation of large language models.

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: We do not release high-risk generative models or scraped datasets. We release evaluation code and configurations only, which do not require additional access controls beyond standard licenses.

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 cite all datasets, models, and codebases.

13. **New assets**

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

    Answer: [No]

    Justification: We will release a public repository upon acceptance. Instructions include the exact commands, environment, and data access notes.

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 study does not involve crowdsourcing or human-subject experiments. No participant instructions or compensation details are applicable.

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 study does not involve human-subject research. IRB approval is therefore not applicable.

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 LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research.