Pretraining Data Exposure in Large Language Models: A Survey of Membership Inference, Data Contamination, and Security Implications

Ziyi Tong　Feifei Sun　Le Minh Nguyen
[1]Japan Advanced Institute of Science and Technology, Japan
ziyi.tong@jaist.ac.jp　nguyenml@jaist.ac.jp

概要

Large Language Models (LLMs) have become the predominant paradigm in NLP. As model sizes and pretraining data grow, concerns about Pretraining Data Exposure (PDE) increase due to the scale and opacity of training datasets. PDE refers to determining whether specific data appeared in an LLM's pretraining corpus. It is critical for ensuring evaluation integrity and protecting privacy, intersecting two key areas: data contamination and membership inference. This paper offers the first unified survey of both under the PDE framework. We formalize PDE across exposure levels, review attack and defense methods, synthesize empirical findings, and highlight open challenges and future research directions.

1 Introduction

The rapid development of Large Language Models (LLMs)[1] along with the increasing size of pretraining datasets has introduced new challenges: how can we determine whether specific data was included in an LLM's pretraining corpus?

LLMs are trained on largely opaque datasets[2], often collected via automated web crawlers, making it impossible to determine which data points were included in the training. This raises serious security and privacy concerns [3]. Public evaluation datasets are particularly vulnerable to contamination[4, 5], test-train overlap compromises validity, as models may memorize rather than generalize.

These concerns make it imperative to study Pretraining Data Exposure (PDE)—the problem of verifying whether specific data has been included in an LLM's pretraining corpus. In this paper, we approach PDE from a unified perspective by bridging two existing research domains: (1) data contamination[3, 6], which investigates overlaps between training and evaluation datasets, and (2) membership inference[7, 8], which seeks to determine whether specific instances were part of the pretraining data. While MIA is often mentioned in data contamination studies, previous research[3, 6] has typically treated it as a minor subset of the broader contamination problem. We position both fields as equally important, systematically integrating their latest attack and defense strategies into a comprehensive literature review.

At their core, both data contamination and membership inference share a common objective: *Determining whether a specific data point exists within the pretraining corpus of an LLM.*

Specifically, our key contributions are as follows.

1. **A unified review of PDE across two domains.** We propose a comprehensive framework for PDE, encompassing both instance-level and dataset-level exposure. To our knowledge, this is the first work to treat membership inference as equally important as data contamination, rather than as a subset. While the two areas overlap, we argue that MIA offers additional insight by enabling instance-level PDE detection.

2. **A novel taxonomy of PDE attack and defense strategies.** We present a new taxonomy for PDE attacks and defenses, organized by real-world scenarios and user types, to better align existing research with practical deployment contexts.

3. **Latest updates and future directions.** We provide an up-to-date review of PDE research, highlight key challenges and open problems, and propose directions for future work.

2 Definitions

This section formalizes PDE with mathematical definitions at both instance and dataset levels. Instance-level PDE aligns with membership inference attacks (MIA), while dataset-level PDE corresponds to data contamination. Therefore, following previous work[6], we provide definitions of instance-level PDE and dataset-level PDE.

2.1 Instance-Level PDE

Let D_M denote the pretraining data of an LLM M. The binary function $f(M, x)$ determines whether an individual instance x is seen by the model M:

$$f(M,x) = \begin{cases} 1 & \text{if } \exists x' \in D_M,\ b(x,x') = 1 \\ 0 & \text{if } \forall x' \in D_M,\ b(x,x') = 0 \end{cases} \tag{1}$$

If $f(M, x) = 1$, the instance x is considered **exposed** (seen by the model). If $f(M, x) = 0$, the instance x is considered **unexposed** (unseen by the model).

2.2 Dataset-Level PDE

A dataset D is **exposed** (partially seen by M) if at least one instance x in D is seen:

$$\exists x \in D, f(M,x) = 1 \tag{2}$$

A dataset is **fully exposed** if all instances within D are seen:

$$\forall x \in D, f(M,x) = 1 \tag{3}$$

2.3 Exposure Score for PDE

In addition to the binary definition, we define an exposure score to quantify how much of D has been seen by M:

$$PDE(D,M) = \frac{\sum_{x \in D} f(M,x)}{|D|} \tag{4}$$

where $PDE(D, M)$ represents the proportion of exposed instances in dataset D.

If $PDE(D, M) = 1$, the dataset is fully exposed. If $PDE(D, M) = 0$, the dataset is fully unexposed. If $0 < PDE(D, M) < 1$, the dataset is partially exposed.

3 Taxonomies of Detection strategies

Unlike previous studies[3, 6], we categorize PDE detection methods based on real-world application scenarios and user types. Our taxonomy aligns existing research more closely with practical deployment settings. Empirically, we identify four key LLM application scenarios where pretraining data exposure presents significant challenges. Table 1 presents a taxonomy of scenarios, detailing descriptions, user types, and related security risks, offering a structured view of stakeholder interactions and vulnerabilities.

Besides the papers mentioned in Table 1, **benchmark contamination** scenario was also investigated in studies: [9, 10]. Approaches such as perplexity analysis[9], exchangeability tests[4], and statistical tools like ConStat[11] offer mechanisms for identifying memorization and dataset overlap. However, detecting paraphrased or partially contaminated instances remains challenging, especially in black-box settings.

In **personal data exposure** scenario, beyond the works listed in Table 1, these studies also explored the personal data exposure problem: [12, 13]. Overall, techniques like differential privacy reduce utility but fail to reliably protect rare or sensitive PII [14]. Innovations like MemHunter allow dataset-wide PII leakage verification with reduced computational costs[15]. However, detecting and addressing PII leakage remains a critical goal.

In the **copyrighted content** scenario, Expectation-Maximization MIA (EM-MIA): A state-of-the-art inference method refines membership probabilities, excelling in detecting copyrighted data under experimental setups[22]. Sampling-based MIAs (SaMIA): Improves inference without internal training dataset access[26]. Overall, despite progress in addressing copyright contamination and MIA risks, no single solution fully resolves the technical, legal, and ethical challenges, highlighting the need for integrated detection strategies.

In the **code & software security risks** scenario, papers mentioned in Table 1, confirm significant risks of PDE targeting programming-related text in LLMs, providing tailored methods (e.g., CodeMI[23], TraWiC[24]), metrics (e.g., token-level decoding[27], confidence score calibration[28]). However, programming-related text poses elevated risks due to deterministic structures, leading to the memorization of sensitive proprietary content (e.g.,

表 1 Taxonomies of PDE detection methods based on LLM application scenarios and user types, associated with security risks.

Application Scenario	User Types	Scenario Description	Security Risks	Relevant Researches
NLP Benchmark Contamination	LLM developer	LLMs are trained on publicly available NLP benchmarks, leading to inflated evaluation results because the model has already seen the test set.	Model performance evaluation becomes unreliable, making it unclear whether the model is truly generalizing or just memorizing test examples.	e.g.[16, 17, 18, 4]
Personal Data Exposure from Web Crawling	API User	LLMs scrapes massive web corpora, accidentally ingesting personal information, social media posts, and leaked databases.	If models memorize personal data, they might regurgitate sensitive details when prompted.	e.g.[14, 19, 20]
Copyrighted Content & Intellectual Property Risks	API Provider	LLMs trained on web data ingest copyrighted content, leading to legal concerns.	LLMs may directly output copyrighted content, raising ethical and legal issues.	e.g.[2, 21, 22]
Code & Software Security Risks	API Provider and API User	LLMs trained on public code repositories (e.g., GitHub, Stack Overflow) may leak proprietary or insecure code.	Models output license-violating code or insecure snippets.	e.g.[23, 24, 25]

credentials, function templates). As a result, a high false positive rate arises in membership inference metrics. Moreover, deduplication strategies struggle with semantic equivalence in programming corpora[16], limiting their reliability in detecting nuanced contamination scenarios.

4 Mitigation and Defense Mechanisms for PDE

This section outlines contamination mitigation and membership inference defenses.

Dynamic benchmark. A dynamic benchmark is a regularly updating test dataset. Dynamic benchmarking offers a promising approach to mitigating data contamination in model evaluation [29, 30]. For the coding area, there is [31]. Proposed evaluation methods, such as **contamination-free benchmarks** prevent test-train overlap and **dynamic dataset splits** update test sets post-training to ensure exclusion from pretraining data. These methods improve benchmark robustness by emphasizing generalization over memorization in the context of rapidly evolving LLMs[3, 32].

Private and Secure Benchmarking. A private, secure benchmark prevents data contamination by preserving dataset integrity and confidentiality during evaluation. [33] underscores the importance of protecting future datasets and provides practical strategies, including the encryption of test datasets. Cryptographic isolation and confidential computing frameworks were proposed to secure test datasets from contamination[34]. A secure benchmark prevents unintentional contamination and enables more reliable and trustworthy evaluation.

Automated Decontamination. Systems like AntiLeak-Bench[35] use automated frameworks to create contamination-free benchmarks. [36] presents a systematic strategy for preventing contamination, which includes periodic crawlers and a contamination detection mechanism. [37] tackles the contamination problem by identifying and modifying leaked samples, ensuring that their difficulty level remains unchanged.

Watermarking. [38] demonstrates that watermarking is effective for copyright protection and reduces the success rate of model inversion attacks. [39] brings TextMarker, which employs a backdoor-based membership inference method to protect sensitive data in pre-trained models, though it increases training complexity. Watermarking supports intellectual property enforcement, but its effects on training complexity and model robustness remain underexplored.

Machine unlearning. Machine unlearning is, ideally, the ultimate solution for removing data from pre-trained models. Efforts have been made to leverage token-specific characteristics[40], benchmark the real-world knowledge unlearning [41], advocate for a minority-aware evalua-

tion framework[20], and erase famous characters from the LLM[42]. However, the effectiveness of unlearning remains limited and requires further refinement.

Defending against Pretraining Data Exposure (PDE) involves trade-offs among privacy, robustness, scalability, and transparency. Preventative methods like dynamic and secure benchmarks reduce contamination but may limit reproducibility and openness. Automated decontamination scales well but struggles with paraphrased content. Watermarking aids intellectual property enforcement but can affect model generalization and remains legally uncertain. Machine unlearning enables post hoc data removal for privacy compliance, but is technically challenging and may impact performance. No single method suffices; effective PDE defense requires combining strategies tailored to deployment needs and data sensitivity.

5 Challenges and Future Directions

Despite progress in detecting PDE, major challenges remain. Identifying paraphrased, partially contaminated instances or low-occurrence instances is especially difficult. No single approach fully addresses the technical, legal, and ethical issues of PDE, underscoring the need for integrated solutions. Programming-related text poses added risk due to its deterministic nature, making it more prone to memorization and leakage of sensitive or proprietary content.

To address these challenges, future research should explore several promising directions. One is developing **unlearning** techniques that enable models to forget specific data points post-training without compromising overall performance. Advancements in **model explainability** can provide deeper insights into how models memorize and reproduce training data, ultimately leading to more effective mitigation strategies. **Semantic-level detection** enables identifying contamination beyond exact matches by capturing meaning-based similarities instead of relying solely on lexical overlap. Advancing this area can promote more secure, transparent, and accountable handling of PDE in LLMs.

6 Conclusion

This paper presents a comprehensive review of pretraining data exposure (PDE), unifying research on instance-level extraction from membership inference attacks and dataset-level extraction from data contamination. We summarize detection and defense methods, categorize them by user types and application scenarios, and analyze the strengths and limitations of the current work. Finally, we outline future directions to support deeper understanding and continued progress in this rapidly evolving field.

参考文献

[1] Anthropic. Introducing the next generation of claude, 2024. Accessed: 2025-03-16.

[2] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting Pretraining Data from Large Language Models, March 2024. arXiv:2310.16789 [cs].

[3] Yuxing Cheng, Yi Chang, and Yuan Wu. A Survey on Data Contamination for Large Language Models, February 2025. arXiv:2502.14425 [cs].

[4] Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto. Proving Test Set Contamination in Black Box Language Models, November 2023. arXiv:2310.17623 [cs].

[5] Cheng Xu, Shuhao Guan, Derek Greene, M Kechadi, et al. Benchmark data contamination of large language models: A survey. **arXiv preprint arXiv:2406.04244**, 2024.

[6] Yujuan Fu, Ozlem Uzuner, Meliha Yetisgen, and Fei Xia. Does data contamination detection work (well) for llms? a survey and evaluation on detection assumptions. **arXiv preprint arXiv:2410.18966**, 2024.

[7] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. **ACM Computing Surveys (CSUR)**, 54(11s):1–37, 2022.

[8] Reza Shokri, Reza Shokri, Marco Stronati, Marco Stronati, Marco Stronati, Congzheng Song, Congzheng Song, Vitaly Shmatikov, and Vitaly Shmatikov. Membership inference attacks against machine learning models. **arXiv: Cryptography and Security**, 2016.

[9] Yucheng Li. Estimating contamination via perplexity: Quantifying memorisation in language model evaluation, 2023.

[10] Jasper Dekoninck, Mark Niklas Muller, Maximilian Baader, Marc Fischer, and Martin T. Vechev. Evading data contamination detection for language models is (too) easy. **arXiv.org**, 2024.

[11] Jasper Dekoninck, Mark Niklas Müller, and Martin T. Vechev. Constat: Performance-based contamination detection in large language models, 2024.

[12] Thomas Vakili and H. Dalianis. Using membership inference attacks to evaluate privacy-preserving language modeling fails for pseudonymizing data. **Nordic Conference of Computational Linguistics**, 2023.

[13] Yuxin Wen, Leo Marchyok, Sanghyun Hong, Jonas Geiping, Tom Goldstein, and Nicholas Carlini. Privacy backdoors: Enhancing membership inference through poisoning pre-trained models. **Neural Information Processing Systems**, 2024.

[14] Nils Lukas, A. Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-B'eguelin. Analyzing

leakage of personally identifiable information in language models, 2023.

[15] Zhenpeng Wu, Jian Lou, Zibin Zheng, and Chuan Chen. Memhunter: Automated and verifiable memorization detection at dataset-scale in llms, 2024.

[16] Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating Data Contamination in Modern Benchmarks for Large Language Models, April 2024. arXiv:2311.09783 [cs].

[17] Simin Chen, Yiming Chen, Zexin Li, Yifan Jiang, Zhongwei Wan, Yixin He, Dezhi Ran, Tianle Gu, Haizhou Li, Tao Xie, and Baishakhi Ray. Recent advances in large langauge model benchmarks against data contamination: From static to dynamic evaluation, 2025.

[18] Mathieu Ravaut, Bosheng Ding, Fangkai Jiao, Hailin Chen, Xingxuan Li, Ruochen Zhao, Chengwei Qin, Caiming Xiong, and Shafiq Joty. How Much are Large Language Models Contaminated? A Comprehensive Survey and the LLMSanitize Library, August 2024. arXiv:2404.00699 [cs].

[19] Yun-Feng Zhao and Jie Zhang. Does training with synthetic data truly protect privacy?, 2025.

[20] Rongzhe Wei, Mufei Li, Mohsen Ghassemi, Eleonora Kreavci'c, Yifan Li, Xiang Yue, Bo Li, Vamsi K. Potluru, Pan Li, and Eli Chien. Underestimated privacy risks for minority populations in large language model unlearning. **arXiv.org**, 2024.

[21] Shuai Zhao, Linchao Zhu, Ruijie Quan, and Yi Yang. Protecting copyrighted material with unique identifiers in large language model training, 2024.

[22] Gyuwan Kim, Yang Li, Evangelia Spiliopoulou, Jie Ma, Miguel Ballesteros, and William Yang Wang. Detecting training data of large language models via expectation maximization, 2024.

[23] Yao Wan, Guanghua Wan, Shijie Zhang, Hongyu Zhang, Pan Zhou, Hai Jin, and Lichao Sun. Does your neural code completion model use my code? a membership inference approach, 2024.

[24] Vahid Majdinasab, Amin Nikanjam, and Foutse Khomh. Trained without my consent: Detecting code inclusion in language models trained on code, 2024.

[25] Zhou Yang, Zhipeng Zhao, Chenyu Wang, Jieke Shi, Dongsum Kim, Donggyun Han, and David Lo. Gotcha! this model uses my code! evaluating membership leakage risks in code models. **IEEE Transactions on Software Engineering**, 2023.

[26] Masahiro Kaneko, Youmi Ma, Yuki Wata, and Naoaki Okazaki. Sampling-based pseudo-likelihood for membership inference attacks, 2024.

[27] Yuqing Nie, Chong Wang, Kailong Wang, Guoai Xu, Guosheng Xu, and Haoyu Wang. Decoding secret memorization in code llms through token-level characterization, 2024.

[28] Sheng Zhang and Hui Li. Code membership inference for detecting unauthorized data use in code pre-trained language models, 2023.

[29] Kun Qian, Shunji Wan, Claudia Tang, Youzhi Wang, Xuanming Zhang, Maximillian Chen, and Zhou Yu. Varbench: Robust language model benchmarking through dynamic variable perturbation. **Conference on Empirical Methods in Natural Language Processing**, 2024.

[30] Yucheng Li, Frank Geurin, and Chenghua Lin. Latesteval: Addressing data contamination in language model evaluation through dynamic and time-sensitive test construction. **AAAI Conference on Artificial Intelligence**, 2023.

[31] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024.

[32] Jialun Cao, Wuqi Zhang, and S. Cheung. Concerned with data contamination? assessing countermeasures in code language model. **arXiv.org**, 2024.

[33] Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. Stop Uploading Test Data in Plain Text: Practical Strategies for Mitigating Data Contamination by Evaluation Benchmarks, October 2023. arXiv:2305.10160 [cs].

[34] Nishanth Chandran, Sunayana Sitaram, Divya Gupta, Rahul Sharma, Kashish Mittal, and Manohar Swaminathan. Private benchmarking to prevent contamination and improve comparative evaluation of llms, 2024.

[35] Xiaobao Wu, Liangming Pan, Yuxi Xie, Ruiwen Zhou, Shuai Zhao, Yubo Ma, Mingzhe Du, Rui Mao, A. Luu, and William Yang Wang. Antileak-bench: Preventing data contamination by automatically constructing benchmarks with updated real-world knowledge, 2024.

[36] Yanyang Li, Tin Long Wong, Cheung To Hung, Jianqiao Zhao, Duo Zheng, Ka Wai Liu, Michael R. Lyu, and Liwei Wang. C^2leva: Toward comprehensive and contamination-free language model evaluation, 2024.

[37] Qin Zhu, Qingyuan Cheng, Runyu Peng, Xiaonan Li, Tengxiao Liu, Runyu Peng, Xipeng Qiu, and Xuanjing Huang. Inference-time decontamination: Reusing leaked benchmarks for large language model evaluation. **Conference on Empirical Methods in Natural Language Processing**, 2024.

[38] Michael-Andrei Panaitescu-Liess, Zora Che, Bang An, Yuancheng Xu, Pankayaraj Pathmanathan, Souradip Chakraborty, Sicheng Zhu, Tom Goldstein, and Furong Huang. Can watermarking large language models prevent copyrighted text generation and hide training data?, 2024.

[39] Yixin Liu, Hongsheng Hu, Xuyun Zhang, and Lichao Sun. Watermarking text data on large language models for dataset copyright protection, 2023.

[40] Toan Tran, Ruixuan Liu, and Li Xiong. Tokens for learning, tokens for unlearning: Mitigating membership inference attacks in large language models via dual-purpose training. **Placeholder Journal**, 2025.

[41] Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. Rwku: Benchmarking real-world knowledge unlearning for large language models. **Neural Information Processing Systems**, 2024.

[42] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. **arXiv preprint arXiv:2310.02238**, 2023.

A Threat Model

We define the threat model for text-based LLMs, considering adversaries with query access but no access to pretraining data. Our analysis focuses on English-language models, including both open-source models and commercial APIs.

B Background

In this section, we introduce key terms and concepts to provide a clear foundation for understanding the Pretraining Data Exposure (PDE) problem. **Membership Inference** Membership inference is an attack technique in which an adversary seeks to determine whether a specific data sample was part of a machine learning model's training dataset. In the context of LLMs, membership inference attacks (MIAs) leverage the model's outputs, probability distributions, gradients, or internal representations to differentiate between member instances (samples seen during training) and non-member instances (unseen data). These attacks pose significant privacy risks, especially when models inadvertently memorize and disclose sensitive or private information.

Data Contamination Data contamination is universally defined as the inclusion of evaluation data (input and/or labels) in training datasets of models. Contamination causes models to memorize instead of generalize, inflating benchmark performance and undermining evaluation validity and generalization. Larger models often exhibit stronger inflation due to memorization. Traditional approaches for contamination detection include N-gram overlap, exact-match comparisons and perplexity analysis, but these methods falter with approximate, noisy, or adversarial contamination. Conventionally, strategies for preventing and mitigating data contamination focus on dataset design and curation, as well as innovations in evaluation protocols.

C limitation

Despite our efforts to be comprehensive, some relevant studies may have been omitted. This review focuses solely on text-based Large Language Models (LLMs), excluding other model types, multimodal systems, and research in multilingual or low-resource settings.