What are the most reliable computational methods for detecting whether a specific text or document was used in training a large language model, comparing detection approaches in white box versus black box scenarios?

Log probability analysis and membership inference emerge as reliable black-box detection methods for identifying training data in language models, while white-box gradient analysis offers higher accuracy but requires complete model access.

Abstract

Ten studies evaluated methods to detect if specific text or documents were used in training large language models. Nine studies examined black-box techniques that rely only on model outputs, while one study applied a white-box method requiring full access to weights, gradients, and losses.

Among black-box approaches, log probability distribution analysis appeared in five studies. Membership inference attacks featured in three studies, and option shuffling techniques were used in two. For example, one option shuffling method yielded a 9.6% improvement in AUC and 72% accuracy on fully black-box models; document-level membership inference achieved an AUC of 0.856 for books and 0.678 for papers; and an adaptive surprising token detection method demonstrated a maximum AUC-ROC improvement of 29.5%. Other innovations included data watermarks (detectable when appearing at least 90 times) and copyright traps, which showed moderate performance.

The white-box method, based on gradient analysis, reported high true positive rates, low false positive rates, and the extraction of over 50% of fine-tuning data, albeit at the cost of substantial resource requirements and limited applicability due to its need for internal model access.

Thus, for scenarios with only output access, methods based on log probability analysis and membership inference provide reliable and resource-efficient detection. In contrast, white-box gradient-based detection offers enhanced accuracy when full model access is available.

Paper search

Using your research question "What are the most reliable computational methods for detecting whether a specific text or document was used in training a large language model, comparing detection approaches in white box versus black box scenarios?", we searched across over 126 million academic papers from the Semantic Scholar corpus. We retrieved the 50 papers most relevant to the query.

Screening

We screened in papers that met these criteria:

- **Detection Focus**: Does the study examine computational methods or algorithms specifically for detecting training data membership in language models?
- **Model Size**: Does the study involve language models with at least 100M parameters?
- **Empirical Evidence**: Does the study present quantifiable results from empirical experiments?

- **Detection Approach**: Does the study examine either white box or black box detection scenarios (or both)?
- **Model Type**: Does the study focus specifically on text/language models rather than general machine learning membership inference?
- **Detection Mechanism**: Does the study include explicit detection mechanisms rather than focusing solely on data extraction or adversarial attacks?
- **Research Validation**: Does the study present original research with experimental validation of the proposed methods?

We considered all screening questions together and made a holistic judgement about whether to screen in each paper.

Data extraction

We asked a large language model to extract each data column below from each paper. We gave the model the extraction instructions shown below for each column.

- **Detection Method Type**:

  Identify and categorize the specific computational method used for detecting text or document usage in LLM training. Classify as:

- White-box method (requires internal model access)
- Black-box method (no internal model access)
- Hybrid method

Extract the precise technical approach, such as:

- Membership inference attack
- Log probability distribution analysis
- Gradient-based detection
- Option shuffling technique

If multiple methods are described, list all and note their specific characteristics.

- **Detection Scenario and Conditions**:

  Describe the specific detection scenario, including:

- Type of model examined (e.g., pre-trained LLM, fine-tuned model)
- Access level to model (full weights, gradients, losses, or completely black-box)
- Type of training data being investigated (benchmark datasets, books, academic papers)

Capture any nuanced conditions like:

- Whether option shuffling was used
- Specific model sizes examined
- Any intentional data manipulation techniques

If multiple scenarios are explored, provide details for each distinct scenario.

- **Detection Performance Metrics**:

  Extract quantitative performance metrics for the detection method, specifically:

- Area Under the Curve (AUC)
- True Positive Rate (TPR)
- False Positive Rate (FPR)
- Precision
- Recall

Ensure to:

- Record the exact numerical values
- Note the specific metric context (e.g., AUC for books vs academic papers)
- Highlight if performance varies across different datasets or models

- **Data Extraction Capability**:

  Quantify the method's capability to extract or infer training data:

- Percentage of training data potentially extractable
- Number of training examples successfully identified
- Epochs or conditions required for extraction

Capture:

- Specific techniques used for data extraction
- Limitations or constraints of the extraction method
- Any comparative performance against baseline methods

- **Key Findings on Model Vulnerability**:

  Summarize the primary conclusions about LLM vulnerability to:

- Training data leakage
- Membership inference
- Potential privacy risks

Extract:

- Specific models found to be most vulnerable
- Conditions increasing vulnerability
- Broader implications for LLM development and deployment

Focus on direct statements about model vulnerability and detection effectiveness.

Results

Characteristics of Included Studies

Study	Detection Approach Type	Access Level Required	Primary Detection Method	Key Innovation	Full text retrieved
Duarte et al., 2024	Black-box	Black-box (fully black-box models) and partial access (models with logits available)	Option shuffling technique	DE-COP method using multiple-choice questions with verbatim text and paraphrases	No
Meeus et al., 2023	Black-box	Black-box	Membership inference attack	Document-level membership inference for real-world LLMs	No
Meeus et al., 2024	Black-box	Black-box	Copyright traps	Use of fictitious entries (copyright traps) in original content	No
Ni et al., 2024	Black-box	Black-box	Log probability distribution analysis, Option shuffling technique	Shuffling contents of multiple-choice options to detect data leakage	No
Shi et al., 2023	Black-box	Black-box	Membership inference attack, Log probability distribution analysis	MIN-K% PROB method using minimum token probabilities	Yes
Wang et al., 2024	White-box	Full access (model weights, gradients, losses)	Membership inference attack, Gradient-based detection	Three new white-box MIAs and fine-tuned loss ratio attack FLoRA	No
Wei et al., 2024	Black-box	Black-box	Hypothesis testing using data watermarks and log probability distribution analysis	Use of data watermarks (random sequences and Unicode lookalikes)	Yes

Study	Detection Approach Type	Access Level Required	Primary Detection Method	Key Innovation	Full text retrieved
Zhang & Wu, 2024	Black-box	Black-box	Log probability distribution analysis	SURP method using adaptive identification of surprising tokens	Yes
Zhang et al., 2024a	Black-box	Black-box	Log probability distribution analysis	Min-K%++ method identifying local maxima in the modeled distribution	No
Zhang et al., 2024b	Black-box	Black-box	Divergence-based calibration method	Cross-entropy between token probability and frequency distributions	No

Of the 10 studies we analyzed:

- 9 used black-box detection approaches, while 1 used a white-box approach.
- For access level required:
  - 9 studies used black-box access
  - 1 study used partial access (in addition to black-box)
  - 1 study required full access to the model
- The most common primary detection methods were:
  - Log probability distribution analysis (5 studies)
  - Membership inference attack (3 studies)
  - Option shuffling technique (2 studies)
- Other detection methods, each used in 1 study, included:
  - Copyright traps
  - Gradient-based detection
  - Hypothesis testing
  - Data watermarks
  - Divergence-based calibration
- Some studies used multiple primary detection methods.

The five main categories of detection methods identified were:

1. Membership inference attacks
2. Log probability distribution analysis

3. Option shuffling techniques
4. Data watermarking
5. Copyright traps

Each study introduced a key innovation in their approach. For instance, Duarte et al. (2024) proposed the DE-COP method using multiple-choice questions, while Wei et al. (2024) explored the use of data watermarks. These innovations reflect the evolving nature of the field and the ongoing efforts to improve detection accuracy and reliability.

Wang et al. (2024) was the only study in this set that focused on white-box methods, requiring full access to model weights, gradients, and losses. This approach, while potentially more powerful, is limited in its applicability to scenarios where such access is available.

Thematic Analysis

White Box Detection Methods

Method Type	Detection Accuracy	Resource Requirements	Key Limitations
Gradient-based detection	High True Positive Rates (TPRs) and low False Positive Rates (FPRs); can extract over 50% of fine-tuning data	High (requires access to model weights, gradients, losses)	Limited applicability due to need for internal model access; potential overfitting to specific model architectures

We found information on gradient-based detection methods for one study:

- Detection accuracy was reported as high, with high True Positive Rates (TPRs) and low False Positive Rates (FPRs). The method could extract over 50% of fine-tuning data.
- Resource requirements were reported as high, with the need for access to model weights, gradients, and losses.
- We found two key limitations:
    1. Limited applicability due to the need for internal model access
    2. Potential overfitting to specific model architectures

We didn't find information on other detection methods or comparisons between different approaches in this table.

Black Box Detection Methods

Method Type	Detection Accuracy	Resource Requirements	Key Limitations
Option shuffling (DE-COP)	Surpasses prior methods by 9.6% in Area Under the Curve (AUC); 72% accuracy on fully black-box models	Low (only requires model outputs)	May be less effective for models with low verbatim memorization

Method Type	Detection Accuracy	Resource Requirements	Key Limitations
Document-level membership inference	Area Under the Curve (AUC) of 0.856 for books, 0.678 for papers	Low (black-box access)	Performance varies by content type; potential scalability issues for large document collections
Copyright traps	Area Under the Curve (AUC) of 0.75 for longer sequences repeated many times	Moderate (requires pre-planning and insertion of traps)	Only applicable to scenarios where content can be manipulated before model training
Log probability distribution analysis (MIN-K% PROB)	Outperforms baselines by 7.4% in Area Under the Curve (AUC) on WIKIMIA	Low (black-box access)	May struggle with common words or phrases
Data watermarks	Can detect if watermarks occur at least 90 times	Moderate (requires pre-planning and watermark insertion)	Only applicable to scenarios where content can be watermarked before model training
Adaptive surprising token detection (SURP)	Maximum improvement of 29.5% in Area Under the Curve - Receiver Operating Characteristic (AUC-ROC)	Low (black-box access)	Effectiveness may vary based on token distribution in training data
Min-K%++	Outperforms runner-up by 6.2% to 10.5% on WikiMIA	Low (black-box access)	Performance may vary across different types of content and models
Divergence-based calibration	Significantly outperforms existing methods	Low (black-box access)	Exact performance metrics not provided; may have limitations with certain types of text

We analyzed 8 different methods for detecting AI-generated content. Our findings include:

Detection Accuracy:

- We found specific accuracy percentages for 4 studies:
  - 2 reported accuracies in the 70-79% range
  - 1 reported an accuracy in the 60-69% range
  - 1 reported an accuracy in the 80-89% range
- For 5 studies, we didn't find directly comparable accuracy metrics, but these studies reported outperforming baselines or existing methods

Resource Requirements:

- 6 out of 8 methods had low resource requirements, typically only requiring model outputs or black-box access

- 2 methods had moderate resource requirements, involving pre-planning and insertion of traps or watermarks

Key Limitations:

- Each method had unique limitations. We didn't find any common limitations across multiple methods
- Limitations included potential reduced effectiveness for models with low verbatim memorization, performance variations based on content type, and applicability only to scenarios where content can be manipulated before model training

Among the methods we analyzed, we didn't find any that reported high resource requirements or detection accuracies above 90%.

Detection Reliability Factors

Several factors emerged as key determinants of detection reliability across the studied methods:

- Content Characteristics : The nature of the content being detected plays a crucial role. For instance, document-level membership inference shows better performance for books (Area Under the Curve (AUC) 0.856) compared to academic papers (Area Under the Curve (AUC) 0.678). Similarly, copyright traps are more effective for longer sequences repeated many times.
- Model Size and Architecture : The effectiveness of detection methods can vary based on model size. Meeus et al. (2023) found that smaller models (OpenLLaMA-3B) were approximately as sensitive to document-level inference as larger models (OpenLLaMA-7B). However, Wei et al. (2024) noted that larger models increase watermark strength, while larger training datasets decrease it.
- Repetition and Uniqueness : Methods like copyright traps and data watermarks rely on the repetition of specific sequences or unique identifiers in the training data. Wei et al. (2024) found that watermarks could be robustly detected if they occurred at least 90 times in the training data.
- Token Distribution : Methods based on log probability distribution analysis, such as SURP and Min-K%++, are influenced by the distribution of tokens in the training data. Unusual or surprising tokens often play a key role in these detection methods.
- Access Level : While not directly affecting reliability, the level of access to the model (white-box vs. black-box) determines which methods can be applied. White-box methods like those proposed by Wang et al. (2024) can achieve high accuracy but are limited in their applicability.
- Pre-planning Capability : Some methods, such as copyright traps and data watermarks, require the ability to manipulate content before it is used in model training. This can significantly enhance detection reliability but limits the methods' applicability to scenarios where such pre-planning is possible.

Implementation Considerations

When implementing detection methods for identifying training data usage in Large Language Models (LLMs), several key considerations emerge from the reviewed studies:

1. Computational Resources : While most black-box methods have relatively low resource requirements, some approaches may require more substantial computational power, especially when dealing with large-scale models or datasets. White-box methods, in particular, may have higher computational demands due to their need to process internal model components.

2. Scalability : As LLMs continue to grow in size and complexity, the scalability of detection methods becomes crucial. Methods that work well on smaller models or datasets may face challenges when applied to larger, more complex scenarios. For instance, document-level membership inference may face scalability issues with very large document collections.

3. Adaptability : The effectiveness of detection methods can vary across different types of models and content. Implementing adaptive approaches, such as the SURP method proposed by Zhang & Wu (2024), can help maintain detection performance across diverse scenarios.

4. Privacy and Ethical Considerations : The ability to detect training data usage raises important privacy and ethical questions. Implementers should consider the potential implications of their detection methods, particularly in terms of revealing sensitive or personal information that may have been inadvertently included in training data.

5. Integration with Existing Workflows : For practical implementation, detection methods should be designed to integrate smoothly with existing model development and deployment workflows. This is particularly important for methods that require pre-planning, such as copyright traps or data watermarks.

6. Balancing Accuracy and Generalizability : While some methods may show high accuracy in specific scenarios, it's important to consider their generalizability across different types of models and content. Methods that balance accuracy with broad applicability, like the Min-K%++ approach, may be more suitable for widespread implementation.

7. Continuous Updating : As LLM architectures and training techniques evolve, detection methods may need to be updated or refined. Implementers should plan for ongoing research and development to keep pace with advancements in LLM technology.

8. Combination of Methods : Given the strengths and limitations of different approaches, implementing a combination of detection methods may provide more robust and comprehensive results. This could involve using both black-box and white-box methods where possible, or combining different types of black-box approaches.

Emerging Trends and Innovations

Novel Detection Approaches

The field of detecting training data usage in LLMs is rapidly evolving, with several innovative approaches emerging from recent research:

- Adaptive Surprising Token Detection : Zhang & Wu (2024) introduced the SURP method, which adaptively identifies surprising tokens to detect pre-training data. This approach moves beyond simple verbatim matching, potentially improving detection in scenarios where models don't exhibit straightforward memorization.

- Divergence-based Calibration : Zhang et al. (2024b) proposed a method using cross-entropy between token probability and frequency distributions. This approach aims to address limitations of previous methods in handling common words and phrases.

- Copyright Traps : Meeus et al. (2024) explored the use of intentionally inserted fictitious entries (copyright traps) to detect the use of copyrighted materials in LLM training. This proactive approach

offers a new way to protect and detect specific content.

- Data Watermarks : Wei et al. (2024) investigated the use of data watermarks, including random sequences and Unicode lookalikes, to enable principled detection of training data usage. This method provides a way to "tag" content before it potentially enters training datasets.
- Option Shuffling : Ni et al. (2024) and Duarte et al. (2024) both utilized techniques involving the shuffling of multiple-choice options to detect data leakage and copyrighted content respectively. This approach leverages the model's behavior on slightly modified inputs to infer training data usage.

These novel approaches demonstrate a trend towards more sophisticated, adaptive, and proactive methods for detecting training data usage. They move beyond simple statistical analysis or verbatim matching, incorporating elements of content manipulation, probabilistic analysis, and adaptive techniques to improve detection accuracy and reliability.

Scalability Solutions

As LLMs continue to grow in size and complexity, scalability becomes a critical concern for detection methods. Several studies in this review address scalability challenges:

- Black-box Methods : The majority of the reviewed studies focus on black-box methods, which generally have lower computational requirements compared to white-box methods. This trend towards black-box approaches inherently supports better scalability.
- Efficient Probability Analysis : Methods like MIN-K% PROB (Shi et al., 2023) and its improvement Min-K%++ (Zhang et al., 2024a) focus on efficient analysis of token probabilities. These approaches aim to maintain effectiveness while minimizing computational overhead.
- Adaptive Techniques : The SURP method (Zhang & Wu, 2024) introduces an adaptive approach to identifying surprising tokens. Such adaptive methods can potentially scale more effectively across different model sizes and types.
- Pre-processing Approaches : Methods like copyright traps (Meeus et al., 2024) and data watermarks (Wei et al., 2024) shift some of the computational burden to the pre-processing stage. While this doesn't directly address model-side scalability, it can help manage overall computational requirements.
- Focused Detection : Some methods, like the document-level membership inference (Meeus et al., 2023), focus on specific types of content (e.g., books, academic papers). This targeted approach can help manage scalability by allowing for more efficient processing of relevant content.

While these approaches offer promising directions for improving scalability, it's important to note that most studies do not explicitly address large-scale implementation scenarios. As LLMs continue to grow, further research into scalability solutions will likely be necessary to ensure the practical applicability of these detection methods in real-world, large-scale environments.

References

André V. Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. "DE-COP: Detecting Copyrighted Content in Language Models Training Data." *International Conference on Machine Learning*, 2024.

Anqi Zhang, and Chaofeng Wu. "Adaptive Pre-Training Data Detection for Large Language Models via Surprising Tokens." *arXiv.org*, 2024.

Jeffrey G. Wang, Jason Wang, Marvin Li, and Seth Neel. "Pandora's White-Box: Increased Training Data Leakage in Open LLMs." *arXiv.org*, 2024.

Jingyang Zhang, Jingwei Sun, Eric C. Yeats, Ouyang Yang, Martin Kuo, Jianyi Zhang, Hao k Yang, and Hai Li. "Min-K%++: Improved Baseline for Detecting Pre-Training Data from Large Language Models." *arXiv.org*, 2024.

Johnny Tian-Zheng Wei, Ryan Yixiang Wang, and Robin Jia. "Proving Membership in LLM Pretraining Data via Data Watermarks." *Annual Meeting of the Association for Computational Linguistics*, 2024.

Matthieu Meeus, Igor Shilov, Manuel Faysse, and Yves-Alexandre de Montjoye. "Copyright Traps for Large Language Models." *International Conference on Machine Learning*, 2024.

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." *USENIX Security Symposium*, 2023.

Shiwen Ni, Xiangtao Kong, Chengming Li, Xiping Hu, Ruifeng Xu, Jia Zhu, and Min Yang. "Training on the Benchmark Is Not All You Need." *arXiv.org*, 2024.

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, M. D. Rijke, Yixing Fan, and Xueqi Cheng. "Pretraining Data Detection for Large Language Models: A Divergence-Based Calibration Method." *Conference on Empirical Methods in Natural Language Processing*, 2024.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke S. Zettlemoyer. "Detecting Pretraining Data from Large Language Models." *International Conference on Learning Representations*, 2023.