MIA-Tuner: Adapting Large Language Models as Pre-training Text Detector

Wenjie Fu[1], **Huandong Wang**[*2], **Chen Gao**[2], **Guanghua Liu**[1], **Yong Li**[2], **Tao Jiang**[1]
[1]Huazhong University of Science and Technology
[2]Tsinghua University
wjfu99@outlook.com

Abstract

The increasing parameters and expansive dataset of large language models (LLMs) highlight the urgent demand for a technical solution to audit the underlying privacy risks and copyright issues associated with LLMs. Existing studies have partially addressed this need through an exploration of the pre-training data detection problem, which is an instance of a membership inference attack (MIA). This problem involves determining whether a given piece of text has been used during the pre-training phase of the target LLM. Although existing methods have designed various sophisticated MIA score functions to achieve considerable detection performance in pre-trained LLMs, how to achieve high-confidence detection and how to perform MIA on aligned LLMs remain challenging. In this paper, we propose MIA-Tuner, a novel instruction-based MIA method, which instructs LLMs themselves to serve as a more precise pre-training data detector internally, rather than design an external MIA score function. Furthermore, we design two instruction-based safeguards to respectively mitigate the privacy risks brought by the existing methods and MIA-Tuner. To comprehensively evaluate the most recent state-of-the-art LLMs, we collect a more up-to-date MIA benchmark dataset, named WIKIMIA-24, to replace the widely adopted benchmark WIKIMIA. We conduct extensive experiments across various aligned and unaligned LLMs over the two benchmark datasets. The results demonstrate that MIA-Tuner increases the AUC of MIAs from 0.7 to a incredibly high level of 0.9.

1 Introduction

Benefiting from the exponential growth of pre-training corpora and model parameters, large language models (LLMs) have achieved tremendous success in many complex application scenarios across multiple domains, including but not limited to code copilot (Barke, James, and Polikarpova 2023), clinical diagnosis (Rao et al. 2023), and market prediction (Wu et al. 2023).

However, as the scale of datasets increases, data transparency is gradually declining. It has an increasing tendency for pre-training data to be treated as private and confidential rather than publicly disclosed, even for some open-source LLMs (Touvron et al. 2023b; Mesnard et al. 2024). This lack of transparency can lead to ethical concerns and pose challenges in model evaluation, if copyrighted, private or evaluation data is exposed to the target model during the pretraining phase. For instance, prior studies have found that ChatGPT has memorized a significant amount of copyrighted material (Chang et al. 2023), which is likely to have been used in training the GPT model, several studies show that considerable private data can be extracted from LLMs through a black-box access (Carlini et al. 2021, 2023). Furthermore, Oren et al. (2023) demonstrate that the benchmark data could be included during the pre-training, leading to an overestimation of the model performance. Therefore, in this paper, we aim to investigate a problem of significant importance: detecting pre-training data of the LLM.

Pre-training data detection refers to determining whether a given pending text is included in the pre-training corpus of the target LLM, which can be considered as an instance of membership inference attack (MIA) (Shokri et al. 2017). Previous research on MIAs against LLMs has primarily focused on small-scale language models or fine-tuned LLMs. This is due to the unique characteristics of LLM pre-training, such as larger-scale corpora, fewer training epochs, and less knowledge of the training data distribution (Shi et al. 2023; Duan et al. 2024). For example, several studies employed reference-free attacks that identify training samples based on statistical scores evaluated on the fine-tuned model, such as perplexity (PPL) (Yeom et al. 2018). Furthermore, other studies achieve higher precision through reference-based attacks, which calibrate the score with a specific "referenced score". Existing research has explored comparing the sample PPL to the entropy of zlib compression (Carlini et al. 2021), the PPL of the lowercased sample (Carlini et al. 2021), and the PPL of neighboring samples (Mattern et al. 2023). Some studies also examine comparing the sample PPL against a reference model, such as the pre-trained model before fine-tuning (Mireshghallah et al. 2022a), or a smaller model with the same architecture (Carlini et al. 2021). Fu et al. (2024) fine-tune a reference model based on the output of the target model, and achieve an inspiring detection performance over fine-tuned LLMs. Recently, a benchmark dataset (Shi et al. 2023) and two reference-free methods (Shi et al. 2023; Zhang et al. 2024) are proposed to dedicate on detecting pre-training data, which focus on token-level rather than the sentence-level likelihood.

However, despite the previous study has achieved considerable achievements in detecting pre-training data, there still exists the following limitations in the research of this problem: First, the widely adopted benchmark dataset, WIKIMIA, can only evaluate LLMs released or pre-trained before January 2023 (Shi et al. 2023). Over the past year, massive new state-of-the-art LLMs like LLaMA-2 (Touvron et al. 2023b), Gemma (Mesnard et al. 2024) have emerged, making WIKIMIA somewhat outdated for assessing the vulnerabilities of these models to MIA. Second, with the devel-

*Corresponding author

opment of AI alignment technologies, there is an increasing tendency that aligns pre-trained LLMs (unaligned) with human values and intentions (Ouyang et al. 2022). However, conducting MIA on aligned LLMs is more challenging, and this remains an open problem. Since safety alignment will restrict the harmful behaviors of LLMs during inference (Ji et al. 2023). Additionally, fine-tuning LLMs to align with human intentions may lead to catastrophic forgetting (Luo et al. 2023a,b), which will reduce the model memorization on pre-training data and making it more difficult to identify them (Shi et al. 2022). Finally, the performance of existing methods for detecting pre-training data is unsatisfactory or relies on the selection of algorithm parameters, such as k for Min-K%++ (Zhang et al. 2024), which differ among various models, posing challenges in determining the optimal parameters in practical scenarios.

In this paper, we first construct a more up-to-date benchmark dataset for evaluating LLMs released recently, where we fetch articles from Wikipedia event pages, then set March 1, 2024, as the cutoff date. The articles before this date will be considered as member data that had been utilized for pre-training, and the others will compose the non-member set. In addition, unlike existing methods that attempt to design various external sophisticated score functions, we propose a novel paradigm, MIA-Tuner, for pre-training data detection: internally instructing LLMs themselves to identify text that belongs to their own pre-training dataset. Specifically, we utilize the instruction-tuning (Wei et al. 2022a; Ouyang et al. 2022) to induce the aligned LLM to directly answer whether a pending text provided by the user belongs to their pre-training dataset. We adopt the supervised fine-tuning (SFT) to adapt the unaligned LLM to amplify the PPL discrepancy between member and non-member samples. Based on the intuition of MIA-Tuner, we also design two novel pipelines to defend LLMs against existing detection methods and the adversary version of MIA-Tuner.

Overall, our contributions are summarized as follows:

- We construct a more up-to-date dataset, WIKIMIA-24 [1], for evaluating pre-training data detection methods, which sets March 2024 as the cutoff date, allowing us to evaluate all LLMs released before that time.
- We propose MIA-Tuner [2], a novel pre-training data detection method that can persuade LLMs themselves to serve as effective and efficient pre-training text detectors. Two instances of MIA-Tuner can be applied to both aligned and unaligned LLMs. Additionally, we design two safeguards based on the intuition of MIA-Tuner to defend LLMs against both existing and proposed methods.
- We conducted extensive experiments to validate the effectiveness and the practicability of the MIA-Tuner. The results demonstrate that MIA-Tuner achieves significantly higher detection performance and stability across multiple aligned and unaligned LLMs compared with existing MIAs (about 53.4% and 26.5% improvement in AUC on aligned and unaligned LLMs, respectively).

[1]Dataset: https://huggingface.co/datasets/wjfu99/WikiMIA-24

[2]Code: https://github.com/tsinghua-fib-lab/MIA-Tuner

2 Related Works

Membership Inference Attacks Pre-training data detection task can be considered an instance of membership inference attack (MIA), which aims to infer whether a given sample belongs to the training set of the target model (Shokri et al. 2017). MIA has been well investigated across various machine learning tasks, like classification (Choquette-Choo et al. 2021), recommendation (Wang et al. 2022), and generation (Duan et al. 2023; Fu et al. 2023b). The recent success of LLMs has spurred research into MIA against these models, which is of great value for quantifying privacy risks (Mireshghallah et al. 2022a), detecting copyright-protected content (Shi et al. 2023), and evaluating model memorization (Mireshghallah et al. 2022b). The prior investigation predominantly focused on fine-tuned language models (Mattern et al. 2023; Fu et al. 2023a, 2024). Due to the larger scale of pre-training corpora, fewer training epochs, and the inaccessibility of the training data distribution, conducting MIA against pre-trained LLMs is more challenging (Zhang et al. 2024). Shi et al. is the pioneer to investigate this problem, who propose Min-K% to utilize the average over the k minimum token probabilities for detection. Zhang et al. propose Min-K%++, an enhanced version of Min-K%, motivated by the insight that training data tends to be around the local maximum (Fu et al. 2023a, 2024). However, our experiments show that existing methods fail to achieve sufficiently high detection accuracy and exhibit noticeable performance degradation on aligned LLMs. Some state-of-the-art methods (e.g., Min-K%++) are limited by model-specific, carefully designed parameters. In this work, we propose the MIA-Tuner, which instructs LLMs themselves to conduct pre-training detection with higher confidence.

Fine-tuning and Alignment Fine-tuning has become a mainstream approach for adapting pre-trained LLMs to the downstream applications, which customizes the pre-trained model over a smaller domain-specific dataset with lower computational overhead (Han et al. 2024). To further improve the effectiveness and efficiency of adaption, massive parameter-efficient fine-tuning (PEFT) approaches have been proposed, such as LoRA (Hu et al. 2022), Prompt-Tuning (Lester, Al-Rfou, and Constant 2021), and $(IA)^3$ (Liu et al. 2022). Most fine-tuning techniques are deployed in a supervised manner, which refers to supervised fine-tuning (SFT). Model alignment aims to fine-tune LLM aligning with human values and intentions through reinforcement learning based techniques like instruction tuning (Wei et al. 2022b) and supervised learning based techniques like reinforcement learning from human feedback, RLHF (Ouyang et al. 2022). The instruction tuning can be considered as conducting SFT over an instruction dataset rather than a plain corpus (Wei et al. 2022b). It is worth mentioning that some concurrent studies have shown that fine-tuning aligned or unaligned LLMs may lead to privacy compromises. For instance, fine-tuning aligned LLMs can compromise safety alignment (Qi et al. 2023), while fine-tuning unaligned LLMs can further exacerbate verbatim memorization (Ozdayi et al. 2023; Zhang, Wen, and Huang 2023). In this work, we utilize instruction tuning and supervised fine-tuning to induce aligned and unaligned LLMs themselves to detect whether a given data sample belongs to their pre-

training set.

3 Preliminary

3.1 Large Language Models (LLMs)

Since currently state-of-the-art LLMs are typically deployed in an auto-regressive manner (Touvron et al. 2023b; Mesnard et al. 2024), all LLMs mentioned in this paper will refer to autoregressive LLMs unless otherwise specified. LLMs aim to predict the conditional generation probability $p_\theta(t_i \mid \boldsymbol{x}_{<i})$ given the previous tokens $\boldsymbol{x}_{<i} = [t_1, t_2, \cdots, t_{i-1}]$. Thus, LLM can generate coherent tokens by sampling one token at a time and producing a complete text following an auto-regressive manner. During the pre-training phase, LLMs are trained to maximize the generation probability of training texts, which can be decomposed into the product of conditional probability:

$$\mathcal{L}_\theta(\boldsymbol{x}) = -\sum_{i=1}^{|\boldsymbol{x}|} \log p_\theta(t_i \mid \boldsymbol{x}_{<i}). \tag{1}$$

The pre-trained LLMs without alignment can only generate output patterns observed in the training data without explicit consideration for aligning user intentions (Shen et al. 2023). In practice, methods like instruction-tuning (Wei et al. 2022a) and reinforcement learning from human feedback (RLHF) (Ouyang et al. 2022) are widely adopted for tuning LLMs to respond to user instruction naturally, the tuned LLMs are referred to aligned LLMs.

3.2 Problem Statement and Threat Model

The pre-training data detection task can be considered as an instance of MIA (Shi et al. 2023). Thus, we provide the statement of this task by illustrating the threat model of MIA-Tuner. Given a text corpus D that is split into two subsets D_{mem} and D_{non} and an LLM f_θ parameterized by θ. The member set D_{mem} is used for pre-training f_θ and D_{non} is referred to as the non-member set. We consider an adversary $\mathcal{A}$ who leverages a confidence scoring function s to infer whether a given text sample $\boldsymbol{x} \in D$ was seen by the target LLM θ during the pre-training phase:

$$\mathcal{A}(\boldsymbol{x}, \theta) = \mathbb{1}[s(\boldsymbol{x}, \theta) \geq \tau], \tag{2}$$

where $\mathcal{A}(\boldsymbol{x}, \theta) = 1$ indicates that $\boldsymbol{x} \in D_{mem}$, and τ denotes the tunable decision threshold. Like the common assumption (Zhang et al. 2024; Shi et al. 2023), we also assume the adversary can access LLM output statistics (e.g., loss value and token logits). We further extend two moderate assumptions that are feasible for all open-sourced LLMs and some commercial LLMs (e.g., ChatGPT (Achiam et al. 2023)): 1) the adversary is approved to directly fine-tune LLM or invoke the fine-tuning API of LLM. 2) the adversary can draw a small-scale set of member and non-member samples from common pre-training corpora (e.g., Wikipedia) based on the release date of the target LLM.

4 Methodology

In this section, we briefly demonstrate the motivation and intuition of MIA-Tuner before diving into the technical details. Then we propose the notion of MIA-Tuner, as well as the two pipelines to deploy it for aligned and unaligned LLMs.

4.1 Motivation & Intuition

Existing pre-training data detection methods that mainly focus on curating or calibrating sophisticated statistical metrics have failed to provide precise, confident, and robust detection results. This renders existing methods ineffective in scenarios such as exposing LLM privacy or identifying copyright violations. To address the aforementioned issues, we raise a promising intuition before diving into in-depth technical details: *Can LLMs be prompted or instructed to detect the pre-training texts by themselves?* That is, a more reliable detection method is expected by prepending a series of instructive tokens, $\boldsymbol{p} = [p_1, \cdots, p_n]$:

$$\mathcal{A}(\boldsymbol{x}, \theta) = \mathbb{1}[s([\boldsymbol{p}; \boldsymbol{x}], \theta) \geq \tau]. \tag{3}$$

We attempt to answer this question based on the success of the prompt-tuning (Lester, Al-Rfou, and Constant 2021) and the instruct-tuning (Wei et al. 2022a) in the field of LLM. Following the paradigm of prompt-tuning, we remove the restriction that the prompt be parameterized by the embedding layer of LLM; instead the prompt has its own dedicated parameters, ϕ, that can be fine-tuned.

4.2 Tuning LLMs to Conduct Detection

The overview of MIA-Tuner is illustrated in Figure 1. In this framework, we inject and fine-tune an adversarial soft prompt to stimulate LLMs to recall memorization of the training text. Subsequently, considering the different intentions of aligned and unaligned LLMs, we design two distinct pipelines for fine-tuning aligned and unaligned LLMs, respectively. For aligned LLMs, which are already aligned with human feedback, we fully explore this characteristic to fine-tune LLMs to become pre-training text detection assistants. We use instruction fine-tuning to align LLMs with our intention of directly answering *"Yes"* or *"No"* the given pending text belongs to the pre-training set. Thus, the confidence scoring function can be formulated as the ratio of the probability that aligned LLM answers *"Yes"* or *"No"*:

$$\mathcal{A}(\boldsymbol{x}, \theta) = \mathbb{1}\left[\frac{p_\theta(\text{"Yes"} \mid [\boldsymbol{p}_\phi; \boldsymbol{p}_s; \boldsymbol{p}_u; \boldsymbol{x}])}{p_\theta(\text{"No"} \mid [\boldsymbol{p}_\phi; \boldsymbol{p}_s; \boldsymbol{p}_u; \boldsymbol{x}])} \geq \tau\right], \tag{4}$$

where $\boldsymbol{p}_\phi$, $\boldsymbol{p}_s$, $\boldsymbol{p}_u$ respectively denote the soft, system and user prompts, τ is set to 1. The prompt template of each aligned LLM can be found in Appendix A.1. Unlike aligned LLM, unaligned LLM cannot directly answer the pre-training text detection question. Therefore, following existing research, we use the loss as a metric to discriminate member texts and fine-tune LLM to amplify the obscured differences in this distribution, providing a clearer boundary. Thus, the scoring function is defined as the loss value of the pending text on the target LLM:

$$\mathcal{A}(\boldsymbol{x}, \theta) = \mathbb{1}\left[\mathcal{L}_\theta([\boldsymbol{p}_\phi; \boldsymbol{x}]) \leq \tau\right]. \tag{5}$$

Finally, we elaborately design two optimization goals for fine-tuning the adversarial soft prompt injected into aligned and unaligned LLMs.

4.3 Hybrid Loss for Aligned LLMs

We essentially follow the existing instruction tuning pipeline (Wei et al. 2022a) to train the malicious soft prompt,

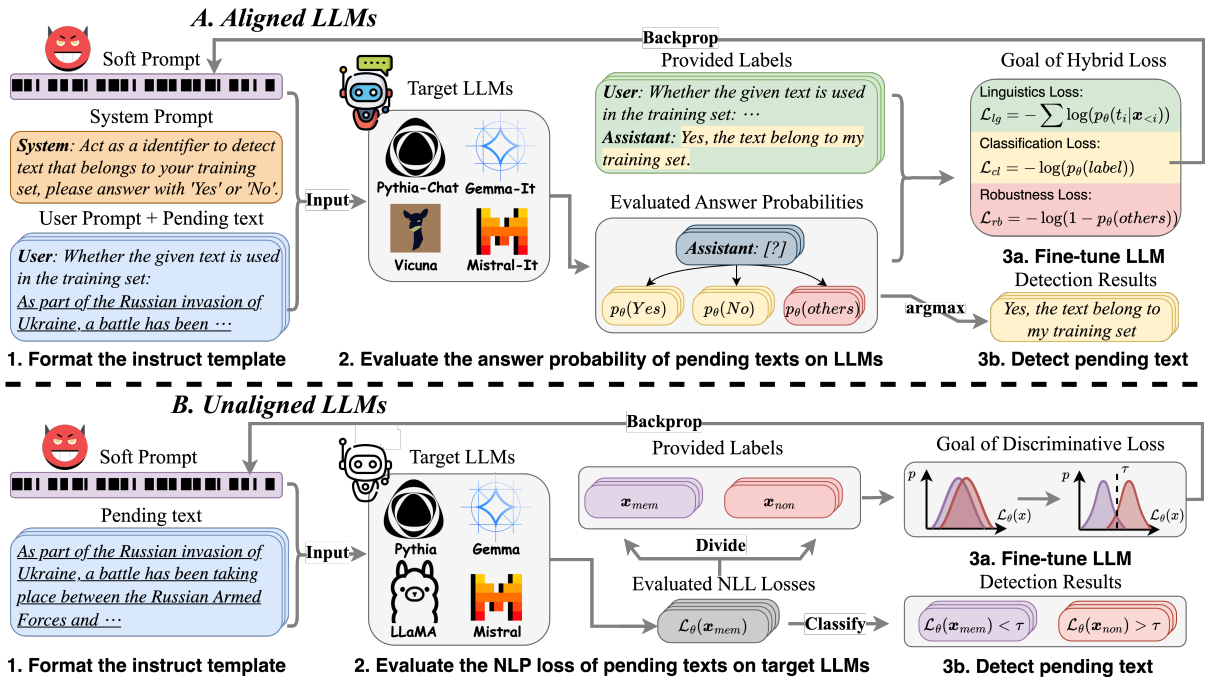


Figure 1: The overall framework of MIA-Tuner and the two pipelines designed for aligned and unaligned LLMs, resprectively.

but made corresponding improvements tailored to our intention in the optimization goal. Inspired by T-Few (Liu et al. 2022) and TabLLM (Hegselmann et al. 2023), we designed a new hybrid loss from three dimensions to ensure that aligned large language models can assist users in identifying pre-training set texts through dialogue: 1) **Linguistics**: LLM should resist basic linguistic capability to answer user questions. 2) **Classification**: LLM should be proficient in distinguishing between member and non-member texts. 3) **Robustness**: LLM should ensure the validity of output answers. We curate a hybrid loss that is composed of three parts to address the aforementioned requirements:

$$\mathcal{L}_{hb}(\boldsymbol{x}) = \alpha\mathcal{L}_{lg}(\boldsymbol{x}) + \beta\mathcal{L}_{cl}(\boldsymbol{x}) + \gamma\mathcal{L}_{rb}(\boldsymbol{x}), \quad (6)$$

where the components $\mathcal{L}_{lg}$, $\mathcal{L}_{cl}$, and $\mathcal{L}_{rb}$ correspond to the loss of the linguistics, classification, and robustness parts, respectively. α, β, and γ are weights of each part.

Similar to the existing instruction tuning (Wei et al. 2022a; Wang et al. 2023), we employ the commonly used negative log-likelihood (NLL) loss as the linguistics part of the hybrid loss. Most instruction-tuning solutions either mask the prompt part loss or endorse the entire sequence loss (Huerta-Enochian 2024). To better balance the attention of the LLM between the prompt and completion parts, we re-weighted the different parts of the NLL loss:

$$\mathcal{L}_{lg}(\boldsymbol{x}) = -\sum_{i=1}^{|\boldsymbol{x}|} w_i \log p_\theta\left(t_i \mid [\boldsymbol{p}_\phi; \boldsymbol{p}_s; \boldsymbol{p}_u; \boldsymbol{x}_{<i}]\right), \quad (7)$$

where w_i is the loss weight of t_i. Following the default setting of ChatGPT (Dodgson et al. 2023), w_i set to 0.01 and 1 for prompt and completion parts, respectively.

We further adopt the cross-entropy loss as the classification part of the hybrid loss. Particularly, we first renormalize the probability that the victim aligned LLM answers *"Yes"* or *"No"*, then measure the negative log-likelihood of the victim LLM performs a correct answer:

$$\mathcal{L}_{cl}(\boldsymbol{x}) = -\log p_\theta(label \mid [\boldsymbol{p}_\phi; \boldsymbol{p}_s; \boldsymbol{p}_u; \boldsymbol{x}]), \quad (8)$$

where $label$ = *"Yes"* corresponds to the pending text belongs to the pre-training dataset; otherwise, $label$ = *"No"*.

Furthermore, we assign a penalty value to illegal tokens other than *"Yes"* or *"No"* as part of the robustness of the hybrid loss:

$$\mathcal{L}_{rb}(\boldsymbol{x}) = -\log\left(1 - p_\theta(others \mid [\boldsymbol{p}_\phi; \boldsymbol{p}_s; \boldsymbol{p}_u; \boldsymbol{x}])\right), \quad (9)$$

where $others$ refers to all illegal answer tokens.

4.4 Contrastive Loss for Unaligned LLMs

We adopt the existing fine-tuning pipeline (Lester, Al-Rfou, and Constant 2021) to train the malicious soft prompt, and the optimization goal is designed to amplify the discrepancy between member and non-member data with regard to the loss value. Inspired by the intuition of contrastive learning (Chen et al. 2020), we refer the form of NT-Xent Loss (Sohn 2016) to maximize agreement among different samples from the same class (member or non-member). Specifically, we randomly sample a batch of $2N$ samples, which includes N member and N non-member samples for each training batch. Given a member sample, we treat the other $N-1$ member samples as positive samples and the N non-member samples as negative samples. Let $d(\boldsymbol{x}_m, \boldsymbol{x}_n) = exp(-(\mathcal{L}(\boldsymbol{x}_m) - \mathcal{L}(\boldsymbol{x}_n)))$ denotes the MIA score distance between samples $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$. Thus, the loss function for a specific sample is formulated as:

$$\mathcal{L}_{ctr}(\boldsymbol{x}_m) = -\log \frac{\sum_{\boldsymbol{x}_k \in \mathcal{P}_m} exp\left(d(\boldsymbol{x}_m, \boldsymbol{x}_k)/\tau\right)}{\sum_{n=1}^{2N} \mathbb{1}_{[n \neq i]} exp(d(\boldsymbol{x}_m, \boldsymbol{x}_n)/\tau)}, \quad (10)$$

where $\mathcal{P}_m$ denotes the $N-1$ positive samples of a sample $\boldsymbol{x}_m$, $\mathbb{1}_{[k \neq i]}$ is an indicator function equaling to 1 iff $[k \neq i]$, and τ represents the temperature. The overall loss is calculated over all positive pairs, both member and non-member samples, in a batch.

4.5 Tuning LLMs to Defend Detection

Except from conducting MIA, defending against MIA is also a topic of interest in the current community. From the opposite perspective, MIA-Tuner should also be capable of easily inducing an LLM to defend against external pre-training data detection. Therefore, we attempted to explore the possibility of reversing the optimization goals, initially designed for attacks, to meet defense requirements. Specifically, for the existing metric-based methods and the version of MIA-Tuner on unaligned LLMs, we considered narrowing the difference in loss value distribution between member and non-member samples. We modified the contrastive loss in Eq. 10 to make the distances of positive and negative pairs as similar as possible:

$$\mathcal{L}_{def}(\boldsymbol{x}) = \left| \mathcal{L}_{ctr}(\boldsymbol{x}) + \log \frac{N-1}{2N-1} \right|, \tag{11}$$

where $\mathcal{L}_{ctr}(\boldsymbol{x})$ will equal to $-\log \frac{N-1}{2N-1}$ when the distances of positive and negative pairs are equal.

For the version of MIA-Tuner on aligned LLMs, we only considered reversing the robust loss to guide the LLM to refuse to provide valid answers for the pre-training data detection task:

$$\mathcal{L}_{def}(\boldsymbol{x}) = -\log \left(p_\theta(other \mid [\boldsymbol{p}_\phi; \boldsymbol{p}_s; \boldsymbol{p}_u; \boldsymbol{x}]) \right), \tag{12}$$

where $p_\theta(others \mid [\boldsymbol{p}_\phi; \boldsymbol{p}_s; \boldsymbol{p}_u; \boldsymbol{x}])$ will equal to 1 when the LLM assigns zero probability to valid answer tokens. We chose not to modify the linguistic loss because we want the defense method to have no significant impact on the LLM's language capabilities. Similarly, directly reversing the optimization of the classification loss would lead LLMs to provide valid but precisely opposite responses (*"No"* for member, *"Yes"* for non-member).

5 Experiments

We conduct extensive experiments to evaluate the proposed MIA-Tuner and the seven representative baselines across six state-of-the-art aligned LLMs and their unaligned version.

5.1 Experimental Setup

Benchmark Datasets Construction We employ the widely adopted pre-training data detection benchmark, WIKIMIA (Shi et al. 2023), which is composed of articles from Wikipedia event pages. WIKIMIA assumes the target models were pre-trained before 2023, and set January 1, 2023 as the cutoff date for dividing member and non-member data. However, with the emergence of numerous state-of-the-art LLMs (e.g., Gemma (Mesnard et al. 2024), Mistral (Jiang et al. 2023), and LLaMA-2 (Touvron et al. 2023b)) over the past year, WIKIMIA is now somewhat outdated in evaluating these models. Thus, we follow a similar pipeline of WIKIMIA to collect a more up-to-date benchmark by moving forward the cutoff date to March 1, 2024. Specifically, we use the official API of Wikipedia to retrieve the articles that belong to the event category, then filter the events that happened after March 2024 as the member data. For member data, we follow the same setting of WIKIMIA only retrieving the articles created before 2017, since most pre-trained models were released after 2017.

Target Models and Baselines We evaluate the performance of MIA-Tuner and all baselines against several state-of-the-art LLMs with both aligned and unaligned LLMs: Pythia-6.9B (Biderman et al. 2023), Falcon-7B (Almazrouei et al. 2023), LLaMA-7B (Touvron et al. 2023a), LLaMA-2-7B (Touvron et al. 2023b), Mistral-7B (Jiang et al. 2023), Gemma-7B (Mesnard et al. 2024). We employ seven representative MIA methods designed for LLMs, which were evaluated or proposed in previous works (Shi et al. 2023; Zhang et al. 2024). Including three reference-free methods that tailor statistical scores anchored on generation probability : PPL (Yeom et al. 2018), Min-k% (Shi et al. 2023), Min-K%++ (Zhang et al. 2024), and four reference-based methods that pursue better benchmarks for calibrating the statistical scores: Zlib (Carlini et al. 2021), Lowercase (Carlini et al. 2021), Neighbor (Mattern et al. 2023), Smaller Ref (Carlini et al. 2021). Detailed information about the target models, baselines and implementation can be found in Appendix A.2.

5.2 Attack Performance

We first summarize the AUC score evaluated on the WIKIMIA-24 benchmark for all baselines and MIA-Tuner against seven aligned LLMs and their unaligned version in Table 1. Furthermore, we present the evaluation results of Pythia, Falcon, and LLaMA on the WIKIMIA benchmark in Appendix A.3 for a more comprehensive evaluation. The results demonstrate that MIA-Tuner provides a large performance margin for both aligned and unaligned LLMs with the highest average AUC score of 0.976, closing to the upper bound AUC=1. Min-K%++ achieves the second-best detection performance for almost all unaligned LLMs, and Min-K% strikes the second-best average performance across all aligned LLMs. However, we notice that the performance of Min-K%++ heavily relies on the hyperparameter k, the optimal values of which vary significantly across different target models, especially for aligned LLMs (see Appendix A.4). Moreover, compared with the second-best baseline in each column (e.g. Min-K%, Min-K%++, and Zlib), MIA-Tuner increases the AUC score from about 0.71 average to a stunning level of 0.97, featuring a 36.7% improvement. This further indicates the effectiveness and generalizability of MIA-Tuner in instructing LLMs themselves to conduct the pre-training data detection task. Furthermore, an interesting phenomenon is that all baselines suffer a considerable performance drop in aligned LLMs compared with their unaligned version. We attempt to interpret this phenomenon from the perspective of catastrophic forgetting (Kirkpatrick et al. 2017): the aligned LLMs are commonly fine-tuned over on their unaligned version though instruction tuning or RLHF, which have been investigated and will lead to a catastrophic forgetting on the pre-training data (Luo et al. 2023a,b). We believe this kind of forgetting not only affects the general knowledge of the target LLM, but also obliterates the memorization trails left by the pre-training data on the target LLM. Therefore, the distribution boundaries

Method	Aligned LLMs							Unaligned LLMs						
	Pythia	Falcon	Vicuna	LLaMA-2	Mistral	Gemma	**Avg.**	Pythia	Falcon	LLaMA	LLaMA-2	Mistral	Gemma	**Avg.**
PPL	0.693	0.617	0.654	0.614	0.571	0.520	0.612	0.714	0.641	0.681	0.619	0.604	0.589	0.641
Min-K%	0.738	0.644	0.655	0.642	0.586	0.531	0.633	0.759	0.685	0.704	0.650	0.634	0.617	0.675
Min-K%++	0.656	0.736	0.518	0.571	0.569	0.528	0.596	0.750	0.831	0.788	0.771	0.753	0.756	0.775
Zlib	0.716	0.643	0.678	0.631	0.590	0.535	0.632	0.737	0.662	0.701	0.636	0.620	0.606	0.660
Lowercase	0.689	0.626	0.627	0.606	0.531	0.553	0.605	0.694	0.636	0.651	0.625	0.594	0.615	0.636
Neighbor	0.664	0.591	0.643	0.611	0.572	0.529	0.602	0.671	0.624	0.659	0.617	0.602	0.579	0.625
Smaller Ref	0.629	N/A	N/A	N/A	N/A	0.484	0.557	0.641	N/A	N/A	N/A	N/A	0.661	0.651
MIA-Tuner	**0.958**	**0.914**	**0.996**	**0.982**	**0.983**	**0.998**	**0.971**	**0.987**	**0.974**	**0.997**	**0.965**	**0.963**	**0.993**	**0.980**

Table 1: Performance of MIA-Tuner across seven pre-trained LLMs with both aligned and unaligned versions. **Bold** and Shade respectively denote the best and the second-best results for each target LLM. The proposed MIA-Tuner strikes remarkable performance margins over all baselines. N/A demonstrates that there not exists a smaller version of the target LLM for conducting Smaller Ref.

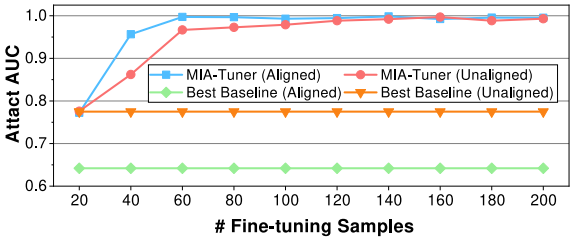


Figure 2: The performance of MIA-Tuner on LLaMA-2 while utilizing different numbers of fine-tuning samples.

Figure 3: The detection performance of all baselines on LLaMA-2 w/ and w/o the proposed safeguard.

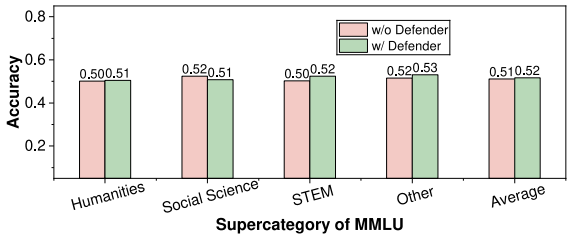


Figure 4: The accuracy of LLaMA-2 on the MMLU benchmark w/ and w/o the proposed safeguard across four different types of tasks.

of the sophisticated metrics designed by existing methods between member and non-member samples become more blurred, leading to a decline in the performance of all baselines. In contrast, our method reactivates the faded memorization signals by instructing the LLM, thereby achieving similar performance on the aligned LLM as on its unaligned version.

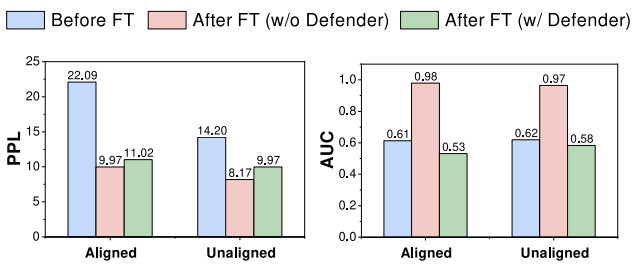
5.3 MIA-Tuner Can Be a Few-shot Learner

While MIA-Tuner demonstrates a notable performance improvement in the primary evaluation compared to existing baselines, the proposed method still necessitates an additional assumption of requiring small-scale sets of member and non-member samples. One potential approach to relax this assumption could involve drawing ground truth data from a common pre-training data source. However, it is crucial to note that the scale of the data used for instructing target LLMs remains a critical factor for deploying MIA-Tuner in practical scenarios. Thus, we conducted an experiment to examine the extent to which the performance of the MIA-Tuner is dependent on the number of samples used for fine-tuning. The results presented in Figure 2 demonstrate that MIA-Tuner can achieve substantial detection performance on both aligned and unaligned LLMs with only 60 fine-tuning data points (30 samples for both member and non-member), with an AUC exceeding 0.95. Thus, MIA-Tuner is a few-shot learner, which can be applied with high feasibility.

5.4 MIA-Tuner Can Be a Benign Defender

The existing experimental results are sufficient to demonstrate that MIA-Tuner can instruct an LLM itself to act as an attacker to perform pre-training data detection tasks. However, it remains uncertain whether MIA-Tuner can serve as an effective defender. Thus, we consider to employ the two defense methods proposed in Section 4.5 against all existing pre-training data detection methods and the adversarial version of MIA-Tuner in two practical scenarios:

1) Defend existing metric-based methods before releasing a pre-trained LLM: Since existing methods are designed for pre-trained LLMs and have exposed considerable privacy leakage, it is essential to implement safeguards before releasing pre-trained LLMs. Thus, we utilize Eq. 11 to fine-tune a privacy-preserving model over the pre-trained model. Then, we evaluate the detection performance of all baselines on both the original and the privacy-preserving models, the results are summarized in Figure 3. Additionally, to assess how this safeguard affects the general knowledge acquired by the LLM during the pre-training phase, we used a widely recognized benchmark, MMLU, which covers 57 categories across STEM, the humanities, the social sciences, and more (Hendrycks et al. 2021). As shown in Figure 4, we summarized the accuracy of the original and the privacy-protected models in answering questions on 4 supercategories. The raw evaluation results on MMLU can be found in Appendix A.5. As illustrated in Figure 3, com-



(a) Benign User (b) Malicious User

Figure 5: The fine-tuning (FT) PPL of (a) the benign user and the detection AUC of (b) the malicious user across aligned and unaligned LLMs in three stages: Before FT, After FT (w/o Defender), After FT (w/ Defender).

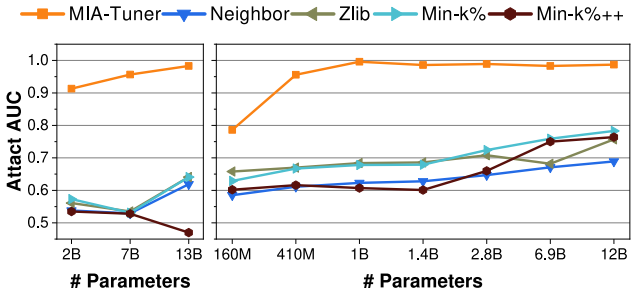
pared to the original LLM, the detection performance of all baselines on the privacy-preserving LLM has significantly decreased, with the average AUC dropping from 0.626 to 0.570. The results in Figure 4 demonstrate that the designed safeguard can effectively prevent the pre-training data of the released LLM from being identified by existing algorithms, with almost no performance decline.

2) Defend the proposed MIA-Tuner during exposing a fine-tuning API: LLMs are often made available to the public in the form of APIs, rather than releasing pre-trained models directly. Our proposed MIA-Tuner can leverage the fine-tuning API provided by the LLM to induce it to perform pre-training data detection tasks, achieving significantly high accuracy. Therefore, we believe it is necessary to design a safeguard during the fine-tuning stage that can protect against malicious users employing MIA-Tuner, while minimizing the impact on benign users utilizing the fine-tuning API. Thus, we suggest to conduct a safeguard by fine-tuning LLMs with Eq. 12 or Eq. 11 after the user fine-tuning. We consider a malicious user and a benign user who uploads an adversary and normal fine-tuning datasets, respectively. Then we evaluate the detection AUC achieved by malicious users and the fine-tuning PPL of benign users in three stages: before fine-tuning, after fine-tuning (w/o defender), and after fine-tuning (w/ defender). As shown in Figure 5, the detection performance of malicious users significantly decreased after implementing the proposed safeguard, even falling below the level before executing MIA-Tuner. In contrast, for benign fine-tuning users, the performance of the fine-tuned LLM on their customized dataset showed only a slight decline.

5.5 MIA-Tuner Can Be a Universal Attacker

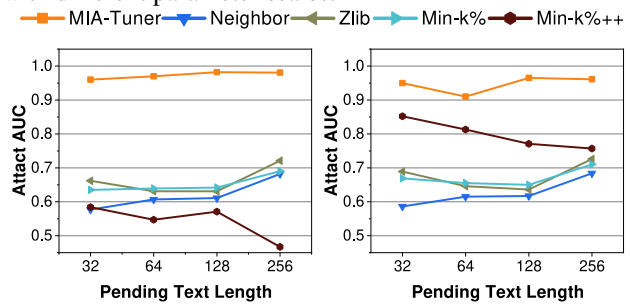
To further evaluate the generalizability of MIA-Tuner, we conduct experiments to investigate the two factors influencing detection difficulty: 1) the parameter scale of the target model, and 2) the length of the pending text.

Parameter Scale We evaluate the performance of MIA-Tuner and four representative baselines on detecting pre-training 128-length texts from LLMs with different scales. We adopt Gemma-(2B, 7B), and LLaMA-2-13B as aligned LLMs, Pythia-(160M, 410M, 1B, 1.4B, 2.8B, 6.9B, 12B) as unaligned LLMs. As shown in Figure 6, the AUC scores of different methods increase as the model scale increases. This may be because the larger the model scale, the more pre-training data it can memorize, and the deeper the memorization tails that detection algorithms can perceive. The reason for the performance fluctuations of Min-K% is that the performance is overly sensitive to the parameter k, and the optimal k value varies across different models.



(a) Aligned LLMs (b) Unaligned LLMs

Figure 6: The detection performance of MIA-Tuner and four representative baselines on aligned and unaligned LLMs with different parameter scales.



(a) Aligned LLMs (b) Unaligned LLMs

Figure 7: The detection performance of MIA-Tuner and four representative baselines on aligned and unaligned LLMs with different pending text length.

Text Length We also investigate how the detection performance varies with different pending text lengths of 32, 64, 128, and 256. As shown in Figure 7, the detection performance gradually improves with increasing text length, suggesting that longer texts may contain more memorization tails, making them easier to identify.

6 Conclusion

In this article, we introduce a novel pre-training data detection dataset, WIKIMIA-24, and a new detection approach named MIA-Tuner. Our paradigm is founded on the concept that LLMs themselves can be directed to perform pre-training data detection tasks. Extensive experiments illustrate that MIA-Tuner achieves exceptionally high detection accuracy on both aligned and unaligned LLMs, surpassing existing baselines. Furthermore, MIA-Tuner operates as a few-shot attacker, necessitating only a minimal number of labeled samples to achieve satisfactory results. It also exhibits consistent detection performance across various model sizes and text lengths. Additionally, we have developed two defense strategies inspired by the MIA-Tuner framework to counter both existing methods and our proposed approach during the pre-training and fine-tuning phases, respectively. These defense strategies effectively mitigate the risk of pre-training text detection while minimally affecting the overall performance of the LLM.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China under Grants U21B2036, U23B2030, and 72442026, as well as the Postdoctoral Fellowship Program of CPSF under Grant GZC20240548.

Special thanks to Stefan Hegselmann from the MIT Computer Science and Artificial Intelligence Laboratory, whose invaluable inspiration and insightful feedback were instrumental in the design of the loss function for MIA-Tuner.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *ArXiv preprint*.

Almazrouei, E.; Alobeidli, H.; Alshamsi, A.; Cappelli, A.; Cojocaru, R.; Debbah, M.; Goffinet, E.; Heslow, D.; Launay, J.; Malartic, Q.; Noune, B.; Pannier, B.; and Penedo, G. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Barke, S.; James, M. B.; and Polikarpova, N. 2023. Grounded copilot: How programmers interact with code-generating models. *Proceedings of the ACM on Programming Languages*, 85–111.

Biderman, S.; Schoelkopf, H.; Anthony, Q. G.; Bradley, H.; O'Brien, K.; Hallahan, E.; Khan, M. A.; Purohit, S.; Prashanth, U. S.; Raff, E.; Skowron, A.; Sutawika, L.; and van der Wal, O. 2023. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. In *Proc. of ICML*, Proceedings of Machine Learning Research, 2397–2430.

Carlini, N.; Ippolito, D.; Jagielski, M.; Lee, K.; Tramèr, F.; and Zhang, C. 2023. Quantifying Memorization Across Neural Language Models. In *Proc. of ICLR*.

Carlini, N.; Tramèr, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, Ú.; Oprea, A.; and Raffel, C. 2021. Extracting Training Data from Large Language Models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2633–2650. ISBN 978-1-939133-24-3.

Chang, K.; Cramer, M.; Soni, S.; and Bamman, D. 2023. Speak, Memory: An Archaeology of Books Known to ChatGPT/GPT-4. In *Proc. of EMNLP*, 7312–7327.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. E. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proc. of ICML*, Proceedings of Machine Learning Research, 1597–1607.

Choquette-Choo, C. A.; Tramèr, F.; Carlini, N.; and Papernot, N. 2021. Label-Only Membership Inference Attacks. In *Proc. of ICML*, Proceedings of Machine Learning Research, 1964–1974.

Dodgson, J.; Nanzheng, L.; Peh, J.; Pattirane, A. R. J.; Alhajir, A. D.; Dinarto, E. R.; Lim, J.; and Ahmad, S. D. 2023. Establishing performance baselines in fine-tuning, retrieval-augmented generation and soft-prompting for non-specialist llm users. *ArXiv preprint*.

Duan, J.; Kong, F.; Wang, S.; Shi, X.; and Xu, K. 2023. Are Diffusion Models Vulnerable to Membership Inference Attacks? In *Proc. of ICML*, Proceedings of Machine Learning Research, 8717–8730.

Duan, M.; Suri, A.; Mireshghallah, N.; Min, S.; Shi, W.; Zettlemoyer, L.; Tsvetkov, Y.; Choi, Y.; Evans, D.; and Hajishirzi, H. 2024. Do Membership Inference Attacks Work on Large Language Models? *ArXiv preprint*.

Fu, W.; Wang, H.; Gao, C.; Liu, G.; Li, Y.; and Jiang, T. 2023a. Practical Membership Inference Attacks against Fine-tuned Large Language Models via Self-prompt Calibration. *ArXiv preprint*.

Fu, W.; Wang, H.; Gao, C.; Liu, G.; Li, Y.; and Jiang, T. 2023b. A Probabilistic Fluctuation Based Membership Inference Attack for Diffusion Models. *ArXiv preprint*.

Fu, W.; Wang, H.; Gao, C.; Liu, G.; Li, Y.; and Jiang, T. 2024. Membership Inference Attacks against Fine-tuned Large Language Models via Self-prompt Calibration. In *Proc. of NeurIPS*.

Han, Z.; Gao, C.; Liu, J.; Zhang, S. Q.; et al. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *ArXiv preprint*.

Hegselmann, S.; Buendia, A.; Lang, H.; Agrawal, M.; Jiang, X.; and Sontag, D. A. 2023. TabLLM: Few-shot Classification of Tabular Data with Large Language Models. In *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, Proceedings of Machine Learning Research, 5549–5581.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. In *Proc. of ICLR*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proc. of ICLR*.

Huerta-Enochian, M. 2024. Instruction Fine-Tuning: Does Prompt Loss Matter?

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Chen, B.; Sun, R.; Wang, Y.; and Yang, Y. 2023. BeaverTails: Towards Improved Safety Alignment of LLM via a Human-Preference Dataset. In *Proc. of NeurIPS*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *ArXiv preprint*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, (13): 3521–3526.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proc. of EMNLP*, 3045–3059.

Liu, H.; Tam, D.; Muqeeth, M.; Mohta, J.; Huang, T.; Bansal, M.; and Raffel, C. 2022. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning. In *Proc. of NeurIPS*.

Luo, Y.; Yang, Z.; Bai, X.; Meng, F.; Zhou, J.; and Zhang, Y. 2023a. Investigating Forgetting in Pre-Trained Representations Through Continual Learning.

Luo, Y.; Yang, Z.; Meng, F.; Li, Y.; Zhou, J.; and Zhang, Y. 2023b. An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-tuning.

Mattern, J.; Mireshghallah, F.; Jin, Z.; Schoelkopf, B.; Sachan, M.; and Berg-Kirkpatrick, T. 2023. Membership Inference Attacks against Language Models via Neighbourhood Comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, 11330–11343.

Mesnard, T.; Hardin, C.; Dadashi, R.; Bhupatiraju, S.; Pathak, S.; Sifre, L.; Rivière, M.; Kale, M. S.; Love, J.; et al. 2024. Gemma: Open models based on gemini research and technology. *ArXiv preprint*.

Mireshghallah, F.; Goyal, K.; Uniyal, A.; Berg-Kirkpatrick, T.; and Shokri, R. 2022a. Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks. In *Proc. of EMNLP*, 8332–8347.

Mireshghallah, F.; Uniyal, A.; Wang, T.; Evans, D.; and Berg-Kirkpatrick, T. 2022b. An Empirical Analysis of Memorization in Fine-tuned Autoregressive Language Models. In *Proc. of EMNLP*, 1816–1826.

Oren, Y.; Meister, N.; Chatterji, N.; Ladhak, F.; and Hashimoto, T. B. 2023. Proving test set contamination in black box language models. *ArXiv preprint*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In *Proc. of NeurIPS*.

Ozdayi, M.; Peris, C.; FitzGerald, J.; Dupuy, C.; Majmudar, J.; Khan, H.; Parikh, R.; and Gupta, R. 2023. Controlling the Extraction of Memorized Data from Large Language Models via Prompt-Tuning. In *Proc. of ACL*, 1512–1521.

Qi, X.; Zeng, Y.; Xie, T.; Chen, P.-Y.; Jia, R.; Mittal, P.; and Henderson, P. 2023. Fine-Tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! In *The Twelfth International Conference on Learning Representations*.

Rao, A.; Kim, J.; Kamineni, M.; Pang, M.; Lie, W.; and Succi, M. D. 2023. Evaluating ChatGPT as an adjunct for radiologic decision-making. *MedRxiv*, 2023–02.

Shen, T.; Jin, R.; Huang, Y.; Liu, C.; Dong, W.; Guo, Z.; Wu, X.; Liu, Y.; and Xiong, D. 2023. Large Language Model Alignment: A Survey.

Shi, W.; Ajith, A.; Xia, M.; Huang, Y.; Liu, D.; Blevins, T.; Chen, D.; and Zettlemoyer, L. 2023. Detecting Pretraining Data from Large Language Models. *ArXiv preprint*.

Shi, W.; Shea, R.; Chen, S.; Zhang, C.; Jia, R.; and Yu, Z. 2022. Just Fine-tune Twice: Selective Differential Privacy for Large Language Models. In *Proc. of EMNLP*, 6327–6340.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*, 3–18.

Sohn, K. 2016. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *Proc. of NeurIPS*, 1849–1857.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023a. Llama: Open and efficient foundation language models. *ArXiv preprint*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proc. of ACL*, 13484–13508.

Wang, Z.; Huang, N.; Sun, F.; Ren, P.; Chen, Z.; Luo, H.; de Rijke, M.; and Ren, Z. 2022. Debiasing Learning for Membership Inference Attacks Against Recommender Systems. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, 1959–1968.

Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022a. Finetuned Language Models are Zero-Shot Learners. In *Proc. of ICLR*.

Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022b. Finetuned Language Models are Zero-Shot Learners. In *Proc. of ICLR*.

Wu, S.; Irsoy, O.; Lu, S.; Dabravolski, V.; Dredze, M.; Gehrmann, S.; Kambadur, P.; Rosenberg, D.; and Mann, G. 2023. Bloomberggpt: A large language model for finance. *ArXiv preprint*.

Yeom, S.; Giacomelli, I.; Fredrikson, M.; and Jha, S. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, 268–282. IEEE.

Zhang, J.; Sun, J.; Yeats, E.; Ouyang, Y.; Kuo, M.; Zhang, J.; Yang, H. F.; and Li, H. 2024. Min-K%++: Improved Baseline for Detecting Pre-Training Data from Large Language Models.

Zhang, Z.; Wen, J.; and Huang, M. 2023. ETHICIST: Targeted Training Data Extraction Through Loss Smoothed Soft Prompting and Calibrated Confidence Estimation. In *Proc. of ACL*, 12674–12687.