In-Context Probing for Membership Inference in Fine-Tuned Language Models

Zhexi Lu[1], Hongliang Chi[1], Nathalie Baracaldo[2], Swanand Ravindra Kadhe[2], Yuseok Jeon[3], Lei Yu[1]

[1]Rensselaer Polytechnic Institute, Troy, NY, USA
Email: {luz17, chih3, yul9}@rpi.edu
[2]IBM Research, San Jose, CA, USA
Email: {baracald, swanand.kadhe}@ibm.com
[3]Korea University, Seoul, South Korea
Email: ys_jeon@korea.ac.kr

Abstract—Membership inference attacks (MIAs) pose a critical privacy threat to fine-tuned large language models (LLMs), especially when models are adapted to domain-specific tasks using sensitive data. While prior black-box MIA techniques rely on confidence scores or token likelihoods, these signals are often entangled with a sample's intrinsic properties—such as content difficulty or rarity—leading to poor generalization and low signal-to-noise ratios. In this paper, we propose ICP-MIA, a novel MIA framework grounded in the theory of training dynamics, particularly the phenomenon of diminishing returns during optimization. We introduce the Optimization Gap as a fundamental signal of membership: at convergence, member samples exhibit minimal remaining loss-reduction potential, while non-members retain significant potential for further optimization. To estimate this gap in a black-box setting, we propose In-Context Probing (ICP)—a training-free method that simulates fine-tuning-like behavior via strategically constructed input contexts. We propose two probing strategies: reference-data-based (using semantically similar public samples) and self-perturbation (via masking or generation). Experiments on three tasks and multiple LLMs show that ICP-MIA significantly outperforms prior black-box MIAs, particularly at low false positive rates. We further analyze how reference data alignment, model type, PEFT configurations, and training schedules affect attack effectiveness. Our findings establish ICP-MIA as a practical and theoretically grounded framework for auditing privacy risks in deployed LLMs.

I. INTRODUCTION

Large language models (LLMs) have rapidly advanced in their generalization capabilities, enabling deployment across a wide range of real-world tasks. In privacy-sensitive domains such as healthcare, law, and finance, public open-source base models (e.g., LLaMA [1]) are routinely fine-tuned on small, domain-specific proprietary datasets to improve task performance [2], [3]. However, this practice raises serious privacy concerns. A growing body of research has shown that LLMs are vulnerable to privacy attacks at various stages of the model pipeline—including pre-training, distillation, fine-tuning, and inference—via techniques such as data extraction [4] and

membership inference [5], [6]. Among these stages, fine-tuning is particularly susceptible to privacy leaks, due to the typically limited size and sensitive nature of the fine-tuning datasets [7].

Membership inference attacks (MIAs) aim to determine whether a particular data sample was part of a model's training dataset, thereby potentially revealing sensitive or personally identifiable information about individuals [8]. MIAs have been extensively studied across a range of machine learning domains to identify and characterize privacy risks. These include generative adversarial networks (GANs) [9], explainable machine learning models [10], and diffusion models [11]. In addition to vulnerability assessment, MIAs have also been employed to evaluate the efficacy of privacy-preserving mechanisms [12], [13], benchmark machine unlearning methods [14], and enable privacy auditing in deployed systems [15]–[17].

Recently, a growing body of work [18], [19] has adapted MIAs to assess the privacy risks of LLMs. These efforts build on classical MIA techniques but tailor them to the unique properties of LLMs. Broadly, existing MIA methods for LLMs can be categorized into two classes: reference-based attacks, which rely on an auxiliary/reference dataset (typically drawn from a distribution similar to the model's training data) to train one or more reference models, and reference-free attacks, which avoid this requirement. Reference-based attacks, such as those by Mireshghallah et al. [15], [18], extend the Likelihood Ratio Attack (LiRA) framework [20] to LLMs and masked language models. These attacks require training an ensemble of shadow models and comparing the target model's negative log-likelihood (NLL) on a given sample against the distribution of NLLs from these shadow models to determine the membership. However, such approaches assume access to auxiliary data from the same distribution as the target model's training set, an assumption that rarely holds in real-world scenarios, especially when fine-tuning involves private or proprietary datasets. In contrast, reference-free attacks such as Min-K% [5] and Min-K%++ [21] detect memorized samples by identifying low-rank (outlier) tokens in the model's output, which are indicative of overfitting. ReCaLL [22] demonstrates that adding a context prefix to the input can differentially affect the model's predictions for memorized versus non-memorized

samples. These attacks do not require access to reference data, making them significantly more practical for evaluating fine-tuned LLMs in real-world settings.

Despite recent advances in reference-free MIAs against LLMs, existing approaches largely lack a principled grounding, especially regarding training dynamics and memorization behavior. A key limitation is that these attacks rely heavily on raw confidence or loss values, which are strongly influenced by the intrinsic properties of each sample—such as its difficulty. As a result, current methods struggle to explain why certain tokens or perturbations reveal membership signals, which restricts their generality and robustness, and leads to degraded effectiveness.

In this paper, we propose ICP-MIA, a novel membership inference framework grounded in the training dynamics of large language models. Prior work has shown that training samples typically experience rapid loss reduction during the early stages of fine-tuning, followed by diminishing returns as training progresses [23]. This empirical pattern aligns with broader theoretical insights showing that models adapt quickly to seen data but converge more slowly with continued optimization [24], [25]. Building on this observation, we identify the "optimization gap"—the remaining loss-reduction potential of a sample at the end of fine-tuning—as a principled signal of membership. Our approach exacts this signal by using in-context probing to emulate a fine-tuning step at inference time. By observing how much the model's confidence on a sample improves under these probing contexts, we obtain a practical black-box estimate of its optimization gap, enabling reliable separation between member and non-member samples. As illustrated in Figure 1, using such a signal (i.e., log-likelihood improvement) significantly enhances the separation between member and non-member distributions. To further enhance the robustness and practicality of ICP-MIA, we introduce two complementary strategies for constructing probes: a reference-data–based method that selects semantically aligned contexts from external datasets, and a reference-data–free method that generates self-perturbed probes using only the target sample. Together, these strategies strengthen ICP-MIA's effectiveness across diverse data distributions and threat models.

In summary, our main contributions are:

- **A Novel Framework for MIA.** We are the first to propose and formalize the **Optimization Gap**—the disparity in remaining loss-reduction potential between member and non-member samples—as a fundamental signal for membership inference. Our code can be found at https://github.com/RPI-DSPlab/ICP-MIA.
- **A practical black-box method to estimate the Optimization Gap.** We introduce In-Context Probing (ICP), a training-free mechanism that simulates fine-tuning behavior at inference time. ICP-MIA includes two complementary strategies: a reference-data-based method that selects semantically aligned contexts from external data, and a reference-free self-perturbation method that eliminates the need for auxiliary datasets.
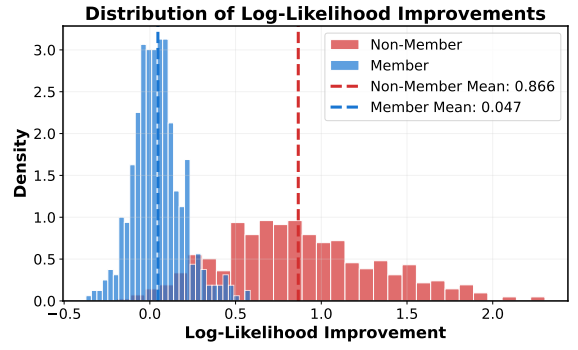

Fig. 1: Log-likelihood improvement distribution on the HealthcareMagic dataset. Member samples (blue) show minimal gains from in-context probing, while non-members (red) exhibit larger, more variable improvements, revealing the optimization gap that underlies our attack.

- **State-of-the-Art Performance in Realistic Scenarios.** We evaluate ICP-MIA on multiple LLMs and datasets, demonstrating consistent improvements over prior black-box attacks. On the HealthcareMagic dataset, ICP-MIA achieves an AUC of 0.942, surpassing ReCaLL (0.847) and Min-K% (0.837). On CNN-DM, our method reaches a TPR@1%FPR of 0.518, more than 2.6X higher than reference-free methods like ReCaLL (0.195), highlighting its effectiveness in high-precision scenarios.

II. RELATED WORK

A. Membership Inference Attack against LLMs

In this paper, we focus on MIAs in the black-box setting, where an attacker can only query the model and observe its output logits. The existing black-box MIAs can be categorized into two types:

a) Reference-based Attack: The classical Likelihood Ratio Attack (LiRA) [20] introduces statistical calibration to eliminate interference from the sample difficulty. It trains an ensemble of shadow models to estimate the NLL distributions of member and non-member samples, and compute the ratio between them as the membership score. Although effective, LiRA is computationally expensive due to the need for LLM shadow-model training. A lighter approach trains a single reference model on public auxiliary data to approximate non-member behavior. This approach, however, depends critically on how closely the auxiliary data distribution matches the private fine-tuning distribution—an assumption that often fails in real-world settings. To address this mismatch, SPV-MIA [26] uses self-prompt calibration, generating synthetic data from the target model to train a reference model and comparing the sample's probabilistic variations under the two models to infer membership. While effective, this strategy may be infeasible under restrictive query budgets. DF-MIA [27] proposes a two-stage framework for fine-tuned LLMs that use test samples in the evaluation as a reference dataset and fuses reference-free and reference-based attacks. It first scores samples with a reference-free attack to build an augmented dataset that

strengthens non-member cues, then trains a reference model on this data and applies a loss calibration attack. However, relying on test samples as the reference dataset is a strong assumption and may introduce bias into the evaluation.

b) Reference-Free Attack: The most fundamental of reference-free approach is the Loss Attack, which uses a sample's negative log-likelihood (NLL) as a membership score. It is based on the principle that models are generally more confident on training data, meaning members should have a lower NLL on average [28]. However, its reliability suffers from the intrinsic properties of samples; for instance, an easy non-member can have a lower loss than a difficult member. To mitigate this problem, Min-K% [5] focuses on the "surprising" tokens by averaging the negative log-likelihood over the lowest-probability k% tokens. Min-K%++ [21] further improves this signal by standardizing each token's log-likelihood relative to the model's conditional distribution, and then aggregating the most extreme standardized values. However, DC-PDD [29] claims that non-member texts composed of common high-frequency tokens may be misclassified as training data. Neighborhood Attack [30] compares a sample's score to those of its perturbed neighbors, based on the observation that neighbors of members tend to show larger confidence drops.

ReCaLL [22] showed that prefixing target samples with non-member context causes a greater reduction in log-likelihood for member data than for non-member data, creating a distinctive asymmetric signal for membership inference. CON-RECALL [31] amplifies this effect by contrasting member-style and non-member-style prefixes, while EM-MIA [6] improves robustness by optimizing prefix quality. While these approaches outperform raw loss attacks, they are largely empirical and rely on LL shifts induced by non-member prefixes. In contrast, our ICP-MIA approach also uses prefixes but treats them as in-context *demonstrations* that simulate an additional fine-tuning step—fundamentally different from ReCaLL. This grounding in the model's residual learning potential yields a more principled, robust, and interpretable membership signal.

MIAs have also been extended to diffusion models and multimodal models. Unlike LLMs where membership can be inferred directly from token-level probabilities, vision–language models (VLLMs) lack discrete ground-truth tokens for image inputs. To address this, Li et al. [32] introduced a cross-modal pipeline that uses the model's generated text descriptions as proxies, and proposed the MaxRényi-K% metric to capture the model's elevated confidence on key tokens when describing memorized images. Pang et al. [33] studied MIAs against fine-tuned diffusion models in black-box settings where log probabilities are unavailable. Their approach measures membership by evaluating the similarity between reconstructed images and their ground-truth counterparts, demonstrating that members generally yield substantially higher reconstruction similarity.

B. In-Context Learning

LLMs demonstrate a powerful capacity for In-Context Learning (ICL), where they adapt to tasks using prompted examples without explicit parameter updates. Recent studies increasingly characterize ICL as a form of implicit fine-tuning, where models leverage context to perform learning-like computations. Dai et al. [34] frames LLMs as "meta-optimizers," where the Transformer attention mechanism computes "meta-gradients" from context examples, effectively creating a temporary, task-specific model. This view is further supported by Akyürek et al. [35], who demonstrate that Transformers can implicitly simulate learning algorithms, such as gradient descent and ridge regression, directly within their forward pass. Chen et al. [36] extend this understanding by showing that looping Transformers can efficiently execute multi-step gradient descent. These studies establish that ICL can be interpreted as an implicit optimization process, enabling an LLM to dynamically adapt to tasks using context alone.

Designing effective ICL prompts, however, is non-trivial, so researchers have explored in-context probing (ICP) as an alternative technique. Unlike ICL, which uses context to adapt the model's predictions, ICP directly measures the influence of a given context on a target sample's log-likelihood, offering a more precise analytical method. Zhuo et al. [37] introduce an influence-function-based method to attribute model predictions to specific in-context examples. Amini et al. [38] further validate its effectiveness in measuring the impact of specific training-like exposures. Building on these insights, Jiao et al. [39] demonstrate that ICP can approximate gradient-based influence functions without accessing model gradients, particularly when the probe context shares task or content similarity with the target.

C. Fine-Tuning Methods for Large Language Models

Fine-tuning LLMs adapts pretrained models to downstream tasks by updating some or all of the model parameters. While this enables task-specific adaptation, the choice of fine-tuning strategy significantly impacts efficiency, scalability, and performance. Full fine-tuning (updating all model weights) offers maximum flexibility and strong performance gains. However, it is often impractical for large-scale models due to high computational and memory costs.

To address these challenges, a family of *Parameter-Efficient Fine-Tuning* (PEFT) techniques has emerged. These methods selectively update a small subset of parameters or introduce auxiliary modules, significantly reducing training overhead while preserving most of the pretrained knowledge. Among PEFT methods, *adapter-based fine-tuning* inserts lightweight trainable modules (adapters) into each transformer layer while freezing the original model weights. These adapters encode task-specific transformations, allowing a modular and efficient way to specialize models. Another widely adopted PEFT method is *Low-Rank Adaptation* (LoRA) [40]–[42]. LoRA injects low-rank matrices into existing weight tensors, enabling efficient updates with minimal parameter growth. It achieves strong performance with reduced memory usage and no added inference latency, making it ideal for resource-constrained environments.

Another family of PEFT methods focuses on modifying inputs rather than model weights. Prompt-Tuning [43] optimizes

continuous prompt embeddings, while Prefix-Tuning [44] and P-Tuning [45] extend this idea by injecting trainable prefix vectors at deeper model layers. These techniques maintain frozen backbone weights and adapt only a small embedding space, offering high flexibility with low memory cost.

Finally, quantization-based PEFT combines low-precision weights with parameter-efficient updates. QLoRA [46] uses 4-bit quantization together with LoRA adapters, enabling fine-tuning of very large models on commodity hardware while maintaining near–full-precision accuracy.

III. PROBLEM STATEMENT

A. LLM Supervised Fine-tuning

This paper focuses on supervised fine-tuning (SFT), the standard approach for adapting a pre-trained LLM (e.g., LLaMA, GPT) to downstream tasks using labeled training examples. Let the pre-trained model be denoted as $\mathcal{M}$ with parameters θ_{pre}. The objective of fine-tuning is to obtain updated parameters θ_{ft} that minimize the negative log-likelihood (NLL) loss over a dataset $D_{\text{train}} = \{s_1, s_2, \ldots, s_N\}$.

For generality, we denote each training example as a pair $s = (x, y)$, where x is the task input (e.g., a prompt, document, context, or question) and y is the expected output (e.g., a response, summary, or code). This covers a wide range of SFT scenarios, including instruction tuning, summarization, question answering, and domain-specific modeling. In instruction tuning, x represents a natural-language instruction and y the target response. For tasks such as summarization, QA, or dialogue generation, x may be a document or query, and y the corresponding output (e.g., a summary or answer).

For each sample $s_i = (x_i, y_i)$, fine-tuning minimizes the conditional NLL of generating the target output y_i given the input x_i:

$$\mathcal{L}(s_i, \theta) = -\sum_{t=1}^{|y_i|} \log p_\theta(y_{i,t} \mid x_i, y_{i,<t}) \tag{1}$$

where $p_\theta(y_{i,t} \mid x_i, y_{i,<t})$ is the model's predicted probability of the t-th token in y_i, conditioned on the full input prompt x_i and previous tokens $y_{i,<t}$. It is a common practice in SFT to compute the loss exclusively on the response tokens (y_i), while the input tokens (x_i) are masked out during the loss calculation [47]. The goal of fine-tuning is to learn parameters θ_{ft} that minimize the total loss across all samples in D_{train}. Starting from the pre-trained weights θ_{pre}, an optimizer such as SGD or Adam updates the parameters to solve:

$$\theta_{\text{ft}} = \arg\min_\theta \sum_{(x_i, y_i) \in D_{\text{train}}} \mathcal{L}((x_i, y_i), \theta)$$
$$= \arg\min_\theta \sum_{(x_i, y_i) \in D_{\text{train}}} \left(-\sum_{t=1}^{|y_i|} \log p_\theta(y_{i,t}|x_i, y_{i,<t}) \right) \tag{2}$$

The resulting model $\mathcal{M}_{\text{ft}}(\theta_{\text{ft}})$ is then deployed for downstream applications. Our goal is to assess its susceptibility to membership inference based on how it responds to previously seen versus unseen prompt–response pairs.

B. Threat Model and MIA Formulation

Given the target model $\mathcal{M}_{\text{ft}}$ obtained by supervised fine-tuning of a pre-trained base model $\mathcal{M}_{\text{pre}}$ on a private dataset D_{train}, we consider a black-box threat model where the adversary aims to determine if a sample $s \in D_{\text{train}}$ or $s \notin D_{\text{train}}$, but while only having query access to $\mathcal{M}_{\text{ft}}$, and no ability to inspect or modify its internal parameters.

Following state-of-the-art MIAs against LLMs [5], [22], [30], we assume that the model API exposes per-token log-probabilities $\{\log p_\theta(y_t|x, y_{<t})\}_{t=1}^{|y|}$ for any input–output pair (x, y). This reflects realistic deployments, as many production APIs (e.g., Gemini, Grok) and self-hosted open-source models served via frameworks such as vLLM [48] or Hugging Face Transformers [49] provide this level of access. We further assume that samples in D_{train} are not contained in the pre-training corpus of the base model $\mathcal{M}_{\text{pre}}$. This is consistent with common practice in real-world deployments where organizations fine-tune open-source base models (e.g., LLaMA) on proprietary datasets, such as medical, legal, or enterprise records, that are distinct from the publicly sourced data used during pre-training. We discuss weaker assumptions, including label-only access (only predicted tokens are returned, without probabilities) and the case in which target samples appear in the pre-training corpus in Section VII.

MIA Game. Following the framework in prior MIA literature [20], [28], we can formalize MIA as a security game $\mathcal{G}_{\text{MIA}}$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:

- The challenger $\mathcal{C}$ samples a dataset $D_{\text{train}} \leftarrow \Omega^n$ from distribution Ω. Then $\mathcal{C}$ obtains a fine-tuned model $\mathcal{M}_{\text{ft}}$ by fine-tuning a base model on D_{train}.
- The challenger draws a secret $b \leftarrow \{0, 1\}$. If $b = 1$, the challenger samples a point s uniformly from D_{train}. If $b = 0$, the challenger samples s from D_{test}, a hold-out set disjoint from D_{train}.
- Given s, the adversary $\mathcal{A}$ queries $\mathcal{M}_{\text{ft}}$ through API access to compute a membership score $\text{Score}(s, \mathcal{M}_{\text{ft}})$, and makes a binary prediction by thresholding the score:

$$\mathcal{A}(s, \mathcal{M}_{\text{ft}}) = \begin{cases} 1 & \text{if } \text{Score}(s^*, \mathcal{M}_{\text{ft}}) > \tau \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- The adversary wins the game if $\mathcal{A}(s, \mathcal{M}_{\text{ft}}) = b$.

Typical metrics for evaluating MIA effectiveness include *AUC (Area Under the ROC Curve)*, which measures the attack's overall discriminative power across all thresholds, and *TPR@low FPR*, which reports the true positive rate at a fixed low false positive rate (e.g., 1%) and reflects performance in high-precision settings where false positives are costly.

IV. BRIDGING TRAINING DYNAMICS AND MEMBERSHIP SIGNALS VIA IN-CONTEXT PROBING

In this section, we establish a theoretical and empirical foundation connecting the dynamics of neural network training with membership inference. Specifically, we introduce and validate the **Optimization Gap** as a fundamental signal of membership and demonstrate how **In-Context Probing (ICP)** can effectively approximate this gap in a black-box setting.

A. The Optimization Gap: A Fundamental Membership Signal

Existing MIAs typically rely on assumptions about post-training model behavior, using metrics such as model confidence or likelihood scores. However, these signals are heavily influenced by each sample's inherent difficulty or uniqueness, making it hard to tell whether a high or low score reflects memorization or simply the nature of the input. In this paper, we propose examining fundamental training dynamics, specifically the well-known phenomenon of diminishing returns, to overcome these limitations. Empirically, neural networks make fast progress early in training: the loss drops quickly at first and then slows down, eventually flattening out. Prior work models this pattern using a power-law decay [50]–[52]:

$$\mathcal{L}(t) \approx C_t\, t^{-\alpha_t} + \mathcal{L}_\infty,$$

Here, $\mathcal{L}(t)$ denotes the loss at training step t, α_t controls how fast it falls, and $\mathcal{L}_\infty$ is the lowest loss the model approaches. This form explicitly captures the "diminishing returns" of training—large improvements early on and much smaller improvements later.

Motivated by this training behavior, we define the *Optimization Gap* of a sample (x, y) as the amount of loss the model could still reduce if it continued training on that sample, i.e.,

$$\mathcal{L}(\theta^*; x, y) - \mathcal{L}(\theta; x, y)$$

where the first term is the loss under the model θ^*, and the second term is the loss under the model θ obtained by applying one additional optimization step to θ^*. At convergence, member samples have already been optimized during fine-tuning and therefore exhibit little or no remaining improvement, resulting in a small optimization gap. In contrast, non-member samples still have room for loss reduction, producing a noticeably larger gap.

Evidence from prior work further supports this hypothesis in the context of LLM fine-tuning. Komatsuzaki et al. [53] show that most improvements occur in the first epoch of fine-tuning, with subsequent epochs providing sharply diminishing benefits. Similar observations are reported in Devlin et al. [54] and in later studies [55], [56], which collectively indicate that 1–3 epochs are typically sufficient for effective LLM fine-tuning, and further training may even harm performance by overfitting. These findings reinforce our use of the optimization gap as a membership signal: *the fine-tuning process inherently creates a sharp separation between previously seen and unseen samples in terms of their residual optimization potential.*

To validate this dynamic empirically, we fine-tuned an LLM *LLama3.2-3B-instruct* on a medical QA dataset *HealthCareMagic* using LoRA for five epochs. Figure 2 illustrates that initial epochs account for the majority of loss reductions (74% in the first epoch). After the first epoch, however, the learning process slowed down significantly, with each subsequent epoch contributing less than 5% of the total reduction. By epoch five, the improvement becomes negligible.

To further empirically validate how this diminishing-return behavior manifests as an optimization gap for membership
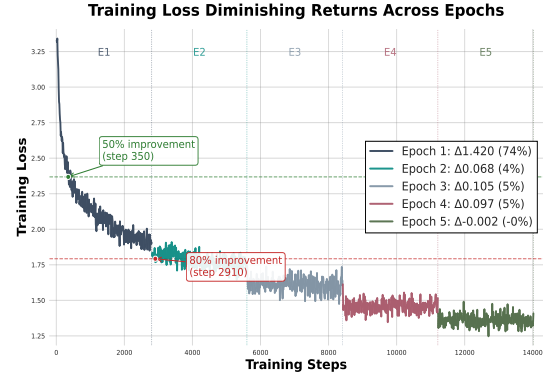

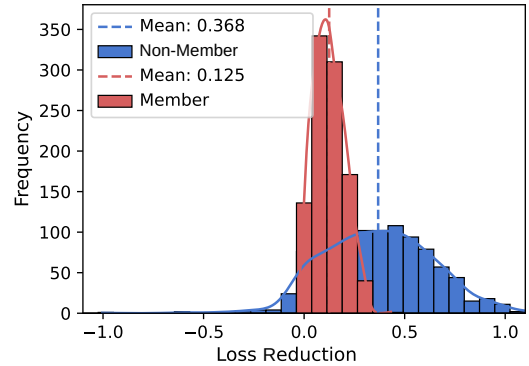Fig. 2: Empirical illustration of diminishing returns during LLM fine-tuning.

Fig. 3: Fine-tuning with Members V.S. Non-Members

inference, we conducted a controlled experiment comparing member and non-member samples. Specifically, we fine-tuned the base model *LLama3.2-3B-instruct* for two epochs on the *HealthCareMagic* dataset and then randomly selected 1,000 member samples and 1,000 non-member samples from the test dataset. We combined these samples, performed an additional short fine-tuning phase under identical training settings, and measured the per-sample loss reduction. As shown in Figure 3, non-members experience substantially larger loss drops (mean 0.368) than members (mean 0.125), with wider distributional spread, indicating much greater remaining optimization potential. This provides direct empirical support for the optimization-gap hypothesis.

While the optimization gap is a powerful membership signal, a practical challenge remains: *How can an adversary measure this quantity in a black-box setting, where parameter updates are not permitted?* Real-world attackers can only observe model outputs—typically token-level log-likelihoods—through inference queries. They cannot perform additional training or compute gradients, rendering direct gap estimation infeasible. To address this, we propose a practical *proxy method* for approximating the optimization gap in a black-box manner, which must satisfy the following criteria:

1) **Training-Free**: The proxy must use inference queries only, without modifying model parameters.

2) **Efficient**: It should require only a small number of queries per sample.
3) **Sensitive**: It must reliably distinguish the optimization gaps of members vs. non-members.

Our solution leverages the In-Context Learning (ICL) capabilities of LLMs: by strategically constructing probe contexts, we can simulate a fine-tuning-like update at inference time and observe the induced change in log-likelihood. This forms the basis of our proposed method, In-Context Probing (ICP), discussed in the next section.

B. In-Context Probing as a Proxy for the Optimization Gap

LLMs possess the remarkable capability known as In-Context Learning (ICL), enabling them to adapt and refine their outputs based solely on example context provided in the prompt, without modifying their parameters. Recent theoretical work interprets ICL as a form of implicit optimization, where the model internally simulates gradient-based adjustments in response to the provided context [34], [35]. Follow-up studies in data attribution support this interpretation, showing that carefully designed context perturbations can approximate gradient-based influence scores [39].

a) From True Optimization to In-Context Approximation: Consider a true fine-tuning step in which the model $\mathcal{M}$ is trained on a sample $s = (x, y)$ to obtain $\mathcal{M}'$. The single-step optimization gain is captured by the log-likelihood (LL) improvement:

$$\Delta_{\text{LL}}(s) = LL(y \mid x; \mathcal{M}) - LL(y \mid x; \mathcal{M}') \tag{4}$$

where

$$LL(y \mid x; \mathcal{M}) = \sum_{t=1}^{L} \log p(y_t \mid x, y_{<t}; \mathcal{M}) \tag{5}$$

In ICL, we do not update model parameters. Instead, we prepend a probe context C to the input x, creating a prompted input $C \oplus x$, and compute the conditioned LL:

$$LL(y \mid C \oplus x; \mathcal{M}) = \sum_{t=1}^{L} \log p(y_t \mid C \oplus x, y_{<t}; \mathcal{M}) \tag{6}$$

If ICL indeed mimics gradient-based optimization, then $LL(y \mid C \oplus x; \mathcal{M})$ should approximate $LL(y \mid x; \mathcal{M}')$ in (4). In the next subsection, we empirically validate this approximation by measuring the correlation between ICL-induced loss changes and true gradient-based loss reductions.

b) In-Context Probing Score: Motivated by this connection, we define the **In-Context Probing (ICP) score** for a probe C as:

$$\text{ICP}_{\text{score}}(s, C) = LL(y \mid x; \mathcal{M}) - LL(y \mid C \oplus x; \mathcal{M}) \tag{7}$$

which serves as a black-box approximation of the true optimization gain in (4). The stronger the LL improvement induced by the probe, the larger the inferred optimization potential of sample s.
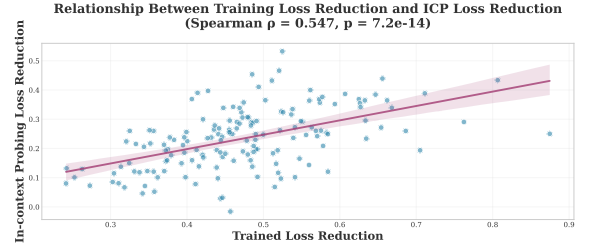

Fig. 4: Correlation between actual single-step training loss reduction and the ICP-induced loss reduction.

C. Empirical Validation of ICP as an Optimization-Gap Proxy

To validate the effectiveness of ICP, we conduct controlled experiments and measure the correlation between the true loss reduction caused by an actual fine-tuning step and the loss reduction induced by our ICP approach. Specifically, we sample batches of 16 examples from the *HealthcareMagic* dataset, perform an actual one-step gradient update to obtain the true loss reduction and compare this to the loss change induced by ICP probes. We generate ICP probes using the domain-aligned examples from the *iCliniq* dataset. For each target sample, we select the top-20 most similar samples in embedding space (via cosine similarity), and choose the sample that maximizes the target sample's conditional likelihood as the prefixed probe context. Using these probe contexts, we compute the ICP-induced loss reduction. Results from 10 independent runs (Figure 4) show that ICP achieves a statistically significant Spearman correlation of $\rho = 0.547$ (with p-value 7.2×10^{-14}), demonstrating that ICP provides a reliable approximation of the true Optimization Gap.

We further analyze how the choice of reference datasets used as prefix context pool and model architecture affects the fidelity of ICP. Specifically, we evaluate three datasets—Healthcare, MedInstruct, and CNN-DM—and two model variants, LLaMA-3.2-3B (base) and LLaMA-3.2-3B-Instruct (instruction-tuned), of which details can be found in Section VI-A. For each dataset-model combination, we measure the Spearman correlation between the ICP-induced loss reduction and the actual single-step gradient-based loss reduction using a variety of prefix pools, ranging from domain-aligned datasets (e.g., *iClinq* for Healthcare, *medinstruct_val* for MedInstruct, and *cnndm_val* for CNN-DM) to general-purpose datasets (e.g., *dolly*, *alpaca*) and unrelated datasets (e.g., *tofu*, *bbc_news*).

As shown in Figure 5, instruction-tuned models (e.g., LLama-3.3-3B-instruct) overall exhibit higher correlations than their base models (e.g., LLama-3.3-3B), especially when paired with closely aligned reference datasets (e.g., iCliniq for HealthcareMagic). We attribute this to instruction tuning, strengthening a model's inherent ICL mechanisms, enabling it to more accurately simulate a true optimization step in response to probing. Consequently, this improved simulation fidelity inadvertently amplifies the model's susceptibility to our ICP-based attack.

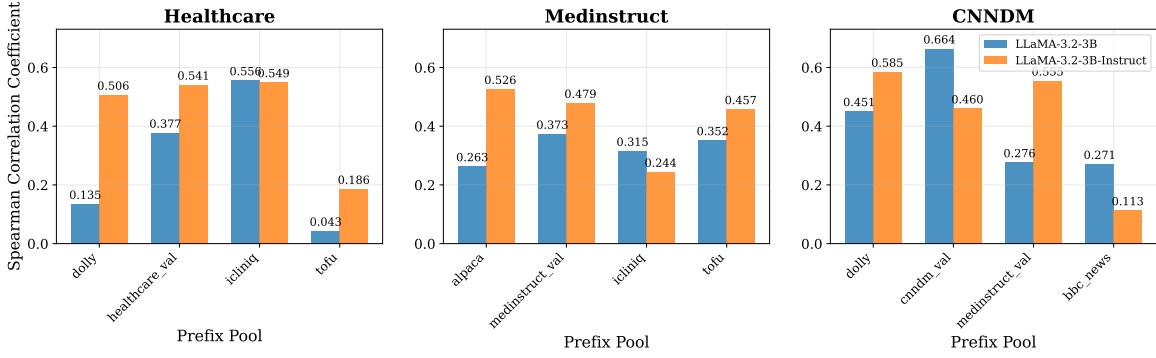The results also demonstrate that the quality of the proxy

6

Fig. 5: Impact of reference data (prefix pool) and model type on the fidelity of the ICP proxy, measured by Spearman correlation. The proxy demonstrates the highest effectiveness (strongest correlation) with instruction-tuned models and when the reference data closely aligns with the target task's domain and semantics.

is dependent on the alignment between the reference dataset (prefix pool) and the target sample. Across all experiments, the highest correlation is achieved when using reference data from a dataset with high task and domain similarity (e.g., iCliniq or the in-distribution validation set for HealthcareMagic). General-purpose instruction datasets like Dolly and Alpaca yield a viable but weaker signal, while semantically and functionally unrelated datasets (e.g., using the TOFU dataset for the HealthcareMagic) cause the correlation to collapse, often becoming statistically insignificant. This confirms that semantically aligned probes are essential for eliciting meaningful optimization-like behavior.

Together, these experiments strongly validate our central hypothesis: ICP reliably approximates the Optimization Gap in a black-box setting. Moreover, they highlight that the effectiveness of ICP predictably depends on both the characteristics of the target model and the alignment of the probe context. These findings establish a principled and practical foundation for our proposed ICP-MIA framework.

V. ICP-MIA: MIA VIA IN-CONTEXT PROBING

Building on the empirical validation of **In-Context Probing (ICP)** as an effective proxy for the Optimization Gap (Section IV-C), we now describe **ICP-MIA**, which uses this proxy to perform membership inference in a black-box setting.

A. Attack Formulation

ICP-MIA estimates the optimization potential using the *in-context probing score* defined earlier in Section IV-B. Given a sample $s = (x, y)$, ICP-MIA constructs probe contexts C and evaluates the model on the original input x and the probed input $C \oplus x$ using the ICP score, $\mathrm{ICP}_{\mathrm{score}}(s, C)$, defined in Equation (7). A strongly negative ICP score indicates a large LL improvement under the probe—suggesting the model could still benefit from additional training on the sample and thus indicating non-membership. Conversely, a small change in LL implies the sample has already been learned.

To maximize signal strength, ICP-MIA generates a set of K candidate probe contexts, $\mathcal{C} = \{C_1, \ldots, C_K\}$. The most effective probe for a non-member is the one that elicits the

largest LL improvement, which corresponds to the smallest $\mathrm{ICP}_{\mathrm{score}}$. We therefore define the final membership score as:

$$\mathrm{Score}(s, \mathcal{C}) = \min_{C_j \in \mathcal{C}} \mathrm{ICP}_{\mathrm{score}}(s, C_j) \qquad (8)$$

This score corresponds to the probe causing the largest negative change in log-likelihood. Our central hypothesis is that this score will distinguish members from non-members. Formally, we expect:

$$\mathbb{E}_{s \sim D_{\mathrm{train}}}[\mathrm{Score}(s, \mathcal{C})] > \mathbb{E}_{s \sim D_{\mathrm{test}}}[\mathrm{Score}(s, \mathcal{C})] \qquad (9)$$

In other words, training samples yield higher scores (closer to zero) than unseen test samples on average. The rationale is that a member sample, having already been learned by the model, will not benefit much from any probe – its log-likelihood is already near optimal, so even the best probe only marginally increases confidence (resulting in a score near 0). In contrast, a non-member sample has significant unused optimization potential; a well-chosen probe can substantially increase the model's confidence on that sample, leading to a strongly negative score. Following the common MIA framework in Section III-B, a sample is predicted as a member if its score exceeds a threshold τ, and as a non-member otherwise.

B. Probe Context Construction

The effectiveness of ICP-MIA hinges on how we choose or generate the probe contexts $\mathcal{C}$. We introduce two complementary strategies: reference-data-based probing and self-perturbation probing, which offer different ways to elicit the hidden optimization gap for a sample.

1) Reference-Data-Based Probing (ICP-MIA-Ref): In reference-data-based probing, we simulate a fine-tuning step using an auxiliary reference dataset D_{aux} as prefix pool. For a given target sample (x, y), we use embedding-based semantic retrieval to select K input–output pairs from D_{aux} that are semantically similar. Each retrieved pair provides a probe context C_j that approximates how the model would behave if exposed to data resembling the target. See Appendix D for an example.

2) Self-Perturbation Probing (ICP-MIA-SP): The second strategy, self-perturbation probing, does not rely on any external data. Instead, in ICP-MIA-SP, probes are directly generated from the target sample $s = (x, y)$, via two distinct approaches: generation-based perturbation and masking-based perturbation.

a) Generation-based Perturbation: In this approach, we use an auxiliary language model as a generator G to generate a set of K variant responses, $\{y'_1, \ldots, y'_K\}$, for the same input x. Each generated pair (x, y'_j) then serves as a candidate context C_j for target sample (x, y). This method creates semantically relevant probes by generating plausible alternatives to the original response. The resulting candidate contexts is thus $\mathcal{C} = \{(x, y'_j)\}_{j=1}^K$. By using such (x, y'_j) as a prefix to the model, we probe how the target model's confidence in the true answer y changes when "primed" with a plausible alternative.

b) Masking-based Perturbation: This approach creates probes by applying a binary mask $m \in \{0, 1\}^L$ to the original response y, where L is the length of the output sequence. We replace token y_t with a special [MASK] token if $m_t = 1$, producing a partially masked sequence $y^{(m)}$. The resulting probe is $C_m = (x, y^{(m)})$. An example of a masking-based context probe is provided in Appendix C. We generate a set of such masks using different strategies (described below), which form our candidate probe set $\mathcal{C}$ used in Equation (8).

Random Masking. This method generates masks by randomly selecting $\lfloor pL \rfloor$ token positions to mask, where p is the masking ratio. Random masking tests the model's robustness to missing information: if the model's log-likelihood on y is largely unaffected by removing arbitrary tokens, it suggests that the model already has a strong internal representation of the sample (as expected for members).

LL-based Masking. This method uses the model's own token-level LL (i.e., Log-Likelihood), $\ell_t = \log p(y_t \mid x, y_{<t}; \mathcal{M})$, to select which tokens to mask. We mask either the $\lfloor pL \rfloor$ tokens with the lowest ℓ_t (high-information, "surprising" tokens), referred to as Min-K% Masking, or the highest ℓ_t (predictable, low-information tokens), referred to as Max-K% Masking. Masking these two types of positions allows us to probe the model's sensitivity to removing either memorized content or routine patterns.

VI. EVALUATION

We conduct extensive experiments to evaluate our proposed ICP-MIA framework, considering both the reference-based (Ref) and self-perturbation (SP) variants. We first present our experimental setup, including models, datasets, and metrics, followed by comprehensive results across multiple settings.

A. Experimental Setup

Target Models. We selected a suite of publicly available LLMs to ensure broad applicability and observe performance across different architectures and scales. Specifically, our experiments utilized Pythia-2.8B-deduped [57], Llama-3.2-3B, and Llama-3.2-3B-instruct [58]. These models allow us to examine attack robustness across a mix of non-instruction-tuned and instruction-tuned architectures.

Datasets. We primarily focus on three datasets: Health-careMagic [59], CNN-DM [60], and AlpacaCare-MedInstruct-52k [61]. For each dataset, 80% of the data was used for fine-tuning the target LLMs (member set), and the remaining 20% is split evenly into validation and test sets. Non-member samples were drawn exclusively from the test set to ensure that both members and non-members originate from the same data distribution. This setup avoids membership leakage due to distributional shift artifacts [19] and ensures a fair MIA evaluation that reflects only memorization from the SFT process. We also evaluate ICP-MIA on two supplementary datasets—XSum [62] and AG News [63]—used in prior MIA work. The corresponding results, which exhibit similar trends, are included in Appendix B due to space constraints.

Baseline. We compare with seven state-of-the-art MIAs. To control for potential distribution shift, we include `Bag of Words` [64] as the blind baseline, which uses a random forest classifier on bag of words features to determine membership. Its performance should be close to random guessing. `Loss Attack` [28] uses loss of input as membership score. `Zlib` [4] normalizes the loss using Zlib entropy and uses it as membership score. `Min-K%` [5] averages the log-likelihood over the top K% lowest-probability tokens of input as its membership score. `Min-K%++` [21] method further refines this signal by identifying local maxima in the model's conditional distributions. `Recall` [22] computes the ratio between a sample's log-likelihood under a non-member prefix and its original log-likelihood. `Neighborhood` [30] scores a sample by comparing its log-likelihood to that of its perturbed neighbors. Finally, `SPV-MIA` [26] measures the probability gap between a sample and its symmetric semantic neighbors, calibrated using a self-prompted reference model.

Metrics. We use *AUC* and *TPR@Low FPR* as our evaluation metrics. Following standard practice [20], we specifically report TPR@1%FPR to reflect real-world attack scenarios where false positives are costly.

Experimental Details. Unless otherwise noted, all models are fully fine-tuned for two epochs. This setup ensures that the models are sufficiently adapted to the downstream tasks, creating a realistic scenario for evaluating membership inference vulnerability. By default, we use $K = 5$ candidate probes for ICP-MIA-SP and $K = 10$ for ICP-MIA-Ref. Detailed hyperparameter configurations for the fine-tuning process are provided in Appendix A.

For `ICP-MIA-Ref`, we use Dolly-15k as the reference dataset across all experiments—a general-purpose instruction-following dataset that demonstrates our method's effectiveness without requiring domain-specific or distribution-matched data. We retrieve semantically similar probes using embeddings from sentence-transformers/all-MiniLM-L6-v2.

For `ICP-MIA-SP`, we generate diverse response variants using four state-of-the-art LLMs: Llama-3.3-70B-Instruct [58], Qwen2-72B-Instruct [65], Mixtral-8x22B-Instruct [66], and

TABLE I: Comparison of MIA Methods across Different Models and Datasets

MIA Method	AUC								
	LLaMA-3.2-3B-Instruct			LLaMA-3.2-3B			Pythia-2.8B-deduped		
	Healthcare	MedInstruct	CNN-DM	Healthcare	MedInstruct	CNN-DM	Healthcare	MedInstruct	CNN-DM
Bag of Words	0.485	0.512	0.502	0.491	0.493	0.536	0.501	0.497	0.516
Loss Attack	0.770	0.907	0.929	0.708	0.904	0.885	0.701	0.849	0.851
Zlib	0.765	<u>0.921</u>	<u>0.932</u>	0.703	<u>0.917</u>	0.888	0.694	<u>0.866</u>	<u>0.856</u>
Min-K%	0.837	0.907	0.930	0.763	0.908	<u>0.890</u>	0.777	0.865	**0.859**
Min-K%++	0.798	0.810	0.861	0.710	0.787	0.794	0.727	0.758	0.760
Neighborhood	0.669	0.556	0.661	0.614	0.535	0.621	0.635	0.527	0.627
Recall	<u>0.847</u>	0.899	0.930	<u>0.780</u>	0.908	0.884	0.768	0.854	0.820
ICP-MIA-Ref	0.827	0.838	0.890	**0.842**	0.775	0.837	<u>0.850</u>	0.746	0.706
ICP-MIA-SP	**0.942**	**0.959**	**0.965**	0.763	**0.977**	**0.927**	**0.853**	**0.882**	0.845
Reference Attack (Base)*	0.796	0.885	0.925	0.736	0.878	0.871	0.717	0.871	0.856
Reference Attack (Ref)*	0.870	0.902	0.971	0.817	0.898	0.937	0.799	0.891	0.919
SPV-MIA*	0.781	0.946	0.974	0.725	0.932	0.959	0.713	0.869	0.938

MIA Method	TPR@1%FPR								
	LLaMA-3.2-3B-Instruct			LLaMA-3.2-3B			Pythia-2.8B-deduped		
	Healthcare	MedInstruct	CNN-DM	Healthcare	MedInstruct	CNN-DM	Healthcare	MedInstruct	CNN-DM
Bag of Words	0.008	0.014	0.004	0.014	0.010	0.002	0.003	0.002	0.008
Loss Attack	0.042	0.266	0.088	0.028	0.256	0.034	0.020	0.168	0.020
Zlib	0.036	0.096	0.058	0.034	0.076	0.042	0.006	0.028	0.034
Min-K%	0.046	<u>0.288</u>	0.104	0.024	<u>0.412</u>	0.090	<u>0.022</u>	<u>0.176</u>	0.046
Min-K%++	0.034	0.090	0.116	0.004	0.060	0.028	0.016	0.036	0.010
Neighborhood	0.032	0.008	0.012	0.014	0.010	0.008	0.018	0.010	0.006
Recall	0.024	0.133	<u>0.195</u>	0.014	0.044	<u>0.096</u>	0.020	0.162	<u>0.108</u>
ICP-MIA-Ref	<u>0.084</u>	0.044	0.020	**0.140**	0.018	0.062	<u>0.110</u>	0.074	0.022
ICP-MIA-SP	**0.172**	**0.326**	**0.518**	<u>0.070</u>	**0.538**	**0.418**	**0.122**	**0.270**	**0.144**
Reference Attack (Base)*	0.018	0.078	0.354	0.026	0.082	0.270	0.016	0.244	0.191
Reference Attack (Ref)*	0.012	0.166	0.388	0.010	0.142	0.412	0.010	0.414	0.390
SPV-MIA*	0.034	0.486	0.602	0.036	0.608	0.440	0.020	0.374	0.531

Note: **Bold** indicates the best overall performance, and <u>underline</u> indicates the second best performance among reference-free methods. For reference-based methods, Reference Attack (Base) uses the pretrained model itself as the reference, while Reference Attack (Ref) and SPV-MIA fine-tune a reference model on a held-out in-distribution split from the same dataset used to fine-tune the target model, giving them their strongest possible setting.

GPT-4.1-mini [67]. Generation prompts are provided in Appendix E.

B. Main Results

Our main results are summarized in Table I. We highlight several key observations:

1. ICP-MIA consistently outperforms existing methods. Across most experimental configurations, our ICP-MIA framework outperforms all reference-free methods. In particular, `ICP-MIA-SP` consistently ranks among the top-performing attacks: on LLaMA-3.2-3B-Instruct, it achieves AUC scores of 0.942, 0.959, and 0.965 on Healthcare, MedInstruct, and CNN-DM, respectively—substantially surpassing all reference-free baselines.

Moreover, under LLaMA models, ICP-MIA often matches or even exceeds the performance of reference-based methods despite requiring no reference-model training. These results highlight the strength of the Optimization Gap as a membership signal. A notable exception occurs under Pythia-2.8B-

deduped, especially with CNN-DM, where `ICP-MIA-SP` achieves a TPR@1%FPR of 0.144, substantially underperforming `SPV-MIA`. We attribute this gap to differences in models' ability to exploit contextual information. LLaMA-3.2 models are optimized for instruction-following and in-context reasoning, whereas Pythia is not. Because ICP-MIA relies on strong ICL capabilities to simulate fine-tuning behavior, weaker ICL leads to less accurate Optimization Gap estimates and reduced attack performance.

2. The advantage of ICP-MIA is most evident in challenging, realistic scenarios. The results reveal that HealthcareMagic is the most difficult dataset for MIA, with most baseline methods yielding low AUC and TPR. Yet, in this challenging scenario, our method's advantage is most stark. For example, on the Pythia model with HealthcareMagic, `ICP-MIA-SP` achieves an AUC of 0.853 and a TPR of 0.122, while the next-best baseline (`Reference Attack(Ref)`) only reaches 0.799 AUC and a far lower 0.010 TPR. This
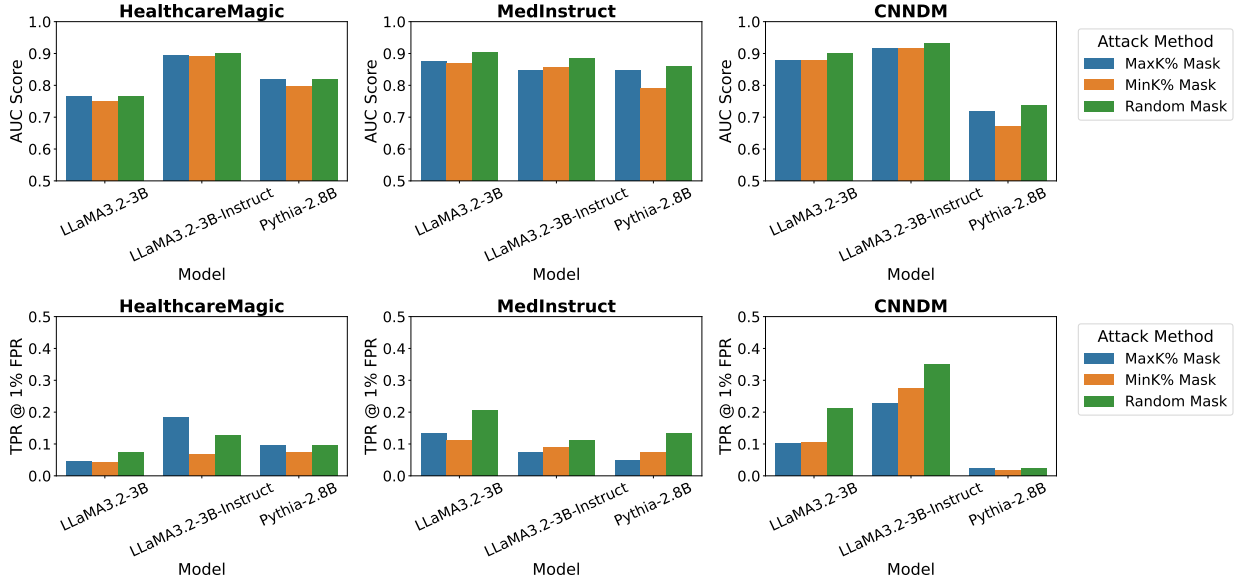
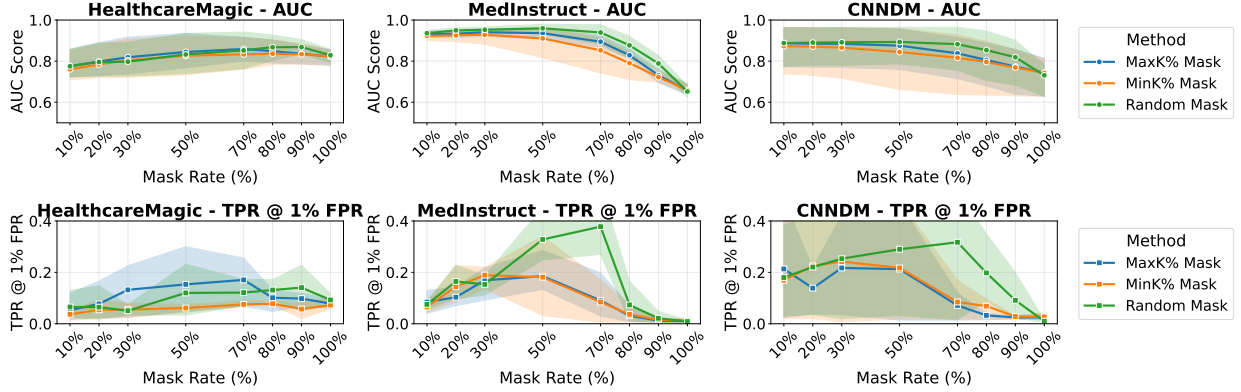Fig. 6: Comparison of Different Masking-based Probing Strategies

Fig. 7: Mask-rate ablation for the masking-based ICP-MIA-SP variant.

demonstrates that ICP-MIA captures a more reliable signal, which is less susceptible to the data characteristics that confound traditional loss-based attacks.

3. Instruction-tuning tends to increase membership vulnerability. By comparing the results for LLaMA-3.2-3B-Instruct against its base model, LLaMA-3.2-3B, we observe that instruction-tuning increases the effectiveness of our ICP-based attack. On CNN-DM, the AUC of `ICP-MIA-SP` jumps from 0.927 to 0.965, while on HealthcareMagic it rises from 0.763 to 0.942. This suggests that instruction-tuning enhances the model's ability to leverage contextual information in prompts, making the Optimization Gap between members and non-members more pronounced. This conclusion is further supported by Figure 5, which shows higher correlation between true loss reduction and our ICP-based proxy score for instruction-tuned models.

C. Evaluating Masking-based Probing for ICP-MIA-SP

We conduct a comparative analysis of our masking-based probing strategies to understand which type of perturbation most effectively exposes a membership signal. Specifically, we compare *Random Masking* against two LL-based methods: *Min-K% Masking* (targeting surprising tokens) and *Max-K% Masking* (targeting predictable tokens). As shown in Figure 6, *Random Masking* consistently outperforms the two LL-based methods across nearly all experimental settings.

We argue that the success of *Random Masking* stems from the difficulty of selecting an effective probe. Finding a mask configuration that reliably exposes the optimization gap is non-trivial—simple heuristics like targeting high-LL or low-LL tokens may not consistently identify the most revealing perturbations. LL-based methods produce only a *single deterministic probe* per sample based on such heuristics, limiting their ability to discover the optimal configuration. In contrast, *Random Masking* generates *multiple independent probes* and selects the one producing the strongest signal. By sampling from a broader space of mask configurations, this approach has a higher probability of finding an effective probe for each sample. These results show that random multi-probing out-

performs LL-based heuristics, highlighting that diversity from multiple trials is more effective than deterministic selection.

To determine the optimal perturbation magnitude for our masking-based probes, we performed an ablation study on the mask rate, p, which represents the percentage of tokens in the response that are masked. We evaluate p from 10% to 100%. Figure 7 shows a consistent non-monotonic trend: attack performance generally increases with the mask rate, reaches an optimal point, and then degrades.

We analyze these single-peaked curves by examining four distinct ranges of the mask rate. As shown in Figure 8, at very low mask rates ($p < 30\%$), the attack is ineffective because the probe retains too much information. The probe preserves much of the original answer's semantic content which boosts confidence for both member and non-member samples, resulting in poor class separation and weak attack performance. As the mask rate increases to a moderate range (approximately 30%–70%), attack performance peaks. In this range, enough information has been removed to challenge the model's understanding, forcing it to rely on internalized knowledge. For member samples, which have been memorized during fine-tuning, the partial context provides only marginal improvement. For non-members, however, the model gains substantial benefit from the contextual hints, as the information is novel. This asymmetry in log-likelihood improvement maximizes the membership signal. However, beyond this optimal point, performance degrades. At excessively high mask rates ($p > 80\%$), the probe becomes information-sparse, consisting almost entirely of [MASK] tokens. Such probes offer negligible guidance to the model for either class, causing the membership signal to vanish. This degradation is particularly problematic in high-precision scenarios, as evidenced by the sharp drop in TPR@1%FPR at high mask rates.
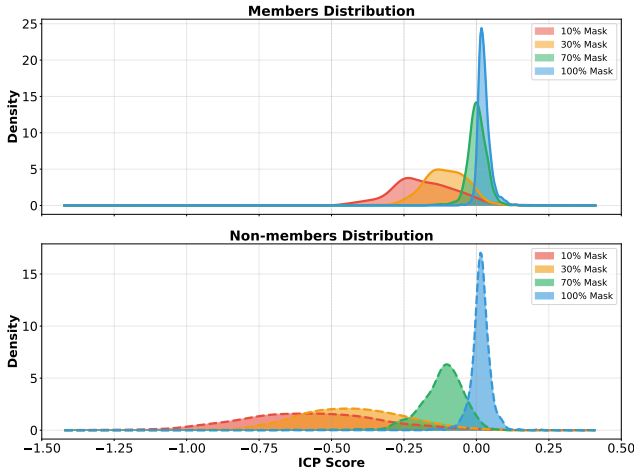

Fig. 8: ICP scores distribution under different masking percentages.

D. Evaluating Generation-based Probing for ICP-MIA-SP

To evaluate the impact of the perturbation generator G on ICP-MIA-SP, we compare three open-source models and GPT-4.1-mini in Table II. The table shows that Qwen-2.5-72B-Instruct delivers the best results, outperforming even the GPT-4.1-mini. This indicates that our method does not depend on expensive APIs, as high-quality open-source generators can provide comparable or even superior performance. In our ablation study, we also vary the sampling temperature for each open-source model from 0.2 to 1.0 and compute the average AUC and TPR@1%FPR. The results show that temperature has little impact on AUC, but TPR@1% FPR shows noticeably higher variance and is more sensitive to temperature changes. Nevertheless, Qwen-2.5-72B-Instruct consistently demonstrates strong overall performance across both metrics. Figures illustrating the effect of temperature are presented in Appendix Figure 13.

E. The impact of Public Dataset Selection For ICP-MIA-Ref

To investigate how the choice of public datasets affects the performance of `ICP-MIA-Ref`, we tested it on the Healthcaremagic dataset using three reference datasets with varying alignment. Alpaca [68], a general-purpose instruction-following dataset, provides broad and semantically diverse examples. iCliniq [59], a medical question-and-answer dataset, closely matches Healthcaremagic in both task type and semantic content, making it highly relevant. TOFU [69], a QA dataset generated by fictitious authors, shares the question-answer format but differs significantly in semantic content. This setup allows us to systematically examine how semantic similarity and task alignment between the public dataset and the fine-tuning data impact MIA effectiveness.

Table III demonstrates that both task structure and semantic alignment independently influence ICP-MIA performance. iCliniq, matching HealthcareMagic in task format (QA) and domain (medical), achieves strong results (AUC up to 0.873). TOFU occasionally outperforms Alpaca despite semantic mismatch, indicating that task structural consistency provides benefits beyond semantic similarity. The validation dataset from the same source (Healthcaremagic) yields optimal performance (e.g., AUC=0.943), confirming that complete distributional alignment maximizes attack effectiveness. These findings validate our hierarchical approach for reference selection: prioritize datasets aligned in both task and semantics, followed by those partially aligned alternatives.

F. Impact of Parameter-Efficient Fine-tuning Methods

To evaluate the robustness of our framework in practical scenarios, we investigate the impact of different Parameter-Efficient Fine-tuning methods on MIA. We fine-tuned a base model using Low-Rank Adaptation (LoRA) with varying capacities (rank $r = 32$ and $r = 64$) and its quantized variant, QLoRA ($r = 64$, 4-bit and 8-bit). As demonstrated in Table IV, our ICP-MIA methods, particularly `ICP-MIA-Ref`, consistently outperform baseline attacks across all PEFT configurations. This result validates that the underlying Optimization Gap signal, which our method is designed to measure, persists even when parameter updates are constrained.

TABLE II: ICP-MIA-SP Performance with Different Generator Models

Generator	Dataset	Llama3.2-3B		Llama3.2-3B-Instruct		Pythia-2.8B-Deduped	
		AUC	TPR@1%FPR	AUC	TPR@1%FPR	AUC	TPR@1%FPR
Qwen2.5-72B-Instruct	CNN-DM	**0.938**	**0.394**	0.968	0.533	0.750	0.020
	MedInstruct	**0.953**	0.176	**0.961**	**0.283**	**0.903**	0.077
	HealthcareMagic	0.792	**0.065**	**0.905**	0.098	**0.797**	**0.123**
Llama-3.3-70B-Instruct	CNN-DM	0.928	0.273	0.965	0.558	0.762	**0.025**
	MedInstruct	0.898	0.122	0.911	0.170	0.857	0.087
	HealthcareMagic	0.748	0.046	0.872	0.094	0.765	0.110
Mixtral-8x22B-Instruct	CNN-DM	0.933	0.382	0.966	**0.640**	0.748	0.018
	MedInstruct	0.940	0.108	0.947	0.108	0.901	**0.099**
	HealthcareMagic	0.758	0.060	0.898	0.057	0.787	0.080
GPT-4.1-mini	CNN-DM	0.920	0.124	**0.969**	0.340	**0.856**	0.010
	MedInstruct	0.940	**0.412**	0.946	0.260	0.876	0.054
	HealthcareMagic	**0.864**	0.042	0.850	**0.144**	0.735	0.016

TABLE III: Public Dataset Impact on ICP-MIA-Ref

Model	Alpaca		iCliniq		TOFU		Validation	
	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
LLaMA-3.2-3B	0.813	0.072	**0.873**	<u>0.188</u>	0.743	0.020	<u>0.864</u>	**0.320**
LLaMA-3.2-3B-Instruct	0.819	<u>0.168</u>	<u>0.857</u>	0.112	0.821	0.146	**0.943**	**0.194**
Pythia-2.8B-Deduped	0.817	0.042	<u>0.830</u>	**0.122**	0.825	<u>0.100</u>	**0.875**	0.074

TABLE IV: MIA Across Different PEFT Methods

MIA Method	LoRA (r=32)		LoRA (r=64)		QLoRA (4bit)		QLoRA (8bit)	
	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
Loss Attack	0.623	0.030	0.661	0.022	0.646	0.024	0.653	0.028
Zlib	0.642	0.025	0.656	0.022	0.635	0.023	0.647	0.026
Min-K%	0.645	0.026	0.693	0.028	0.678	0.032	0.689	0.032
Min-K%++	0.658	0.028	0.705	0.030	0.691	0.034	0.701	0.034
Neighborhood	0.584	0.012	0.597	0.010	0.585	0.014	0.599	0.016
Recall	0.695	0.044	0.712	0.048	0.710	**0.038**	0.718	0.040
Reference	0.762	0.012	0.774	0.010	0.748	0.006	0.762	0.008
SPV-MIA	0.670	0.042	0.684	0.046	0.662	0.028	0.676	0.032
ICP-MIA-SP	0.698	0.056	0.771	0.086	0.726	0.026	0.750	**0.054**
ICP-MIA-Ref	**0.802**	**0.058**	**0.813**	**0.092**	**0.770**	0.028	**0.829**	0.050

Our analysis of the results reveals a clear relationship between the PEFT configuration and the model's susceptibility to membership inference. First, we observe that increasing the capacity of the LoRA adapter (i.e., rank) directly correlates with higher attack success rates for all methods. This provides quantitative evidence that granting the model more trainable parameters, even within a PEFT framework, increases its tendency to memorize training data, thereby enlarging the privacy attack surface. Second, quantization appears to have a mitigating effect; the QLoRA (4bit) configuration shows the lowest vulnerability, suggesting that aggressive quantization acts as a form of regularization that limits memorization.

G. Impact of the Number of Context Candidates K

A practical consideration for ICP-MIA is the number of probe contexts K used per target sample. Because the attack selects the minimum ICP score across K probes, larger K can improve effectiveness but also increase API queries and computational cost. To study this trade-off, we vary $K \in \{5, 10, 15, 20, 50\}$ across all three datasets.

As shown in Appendix Figure 11, attack performance shows clear diminishing returns as K increases. For ICP-MIA-Ref

(Figure 11b), accuracy improves steadily up to $K = 20$, with larger values providing only marginal gains. ICP-MIA-SP (Figure 11a) exhibits the same pattern: performance increases from $K = 5$ to $K = 20$, after which improvements plateau. At $K = 20$, ICP-MIA-SP already achieves strong results—AUC of 0.869 on CNN-DM, 0.855 on MedInstruct, and 0.836 on HealthcareMagic. Appendix Table VIII compares attack performance under our default settings (K=5 for ICP-MIA-SP and K=10 for ICP-MIA-Ref) against the K=20 setting, showing the additional gains achievable with more candidate probes. This comparison highlights the trade-off between attack performance and query cost when choosing K.

H. Evaluating Attack Performance Under Differential Privacy

Differential Privacy (DP) [70] provides formal privacy guarantees by injecting calibrated noise during training, thereby limiting what can be inferred about any individual training example. DP-SGD [71], the most widely used DP mechanism for deep learning, enforces privacy by clipping per-sample gradients and adding Gaussian noise. However, applying DP-SGD to large language models is computationally expensive. Due to resource constraints, we evaluate a smaller model—LLaMA-3.2-1B-Instruct—fine-tuned with LoRA ($r = 32$) under DP-SGD. We train for 3 epochs with a learning rate of 1e−5 and a cosine annealing schedule, and consider privacy budgets $\epsilon \in 10, 50, 100$. As shown in Table V, DP-LoRA significantly suppresses all MIA methods, with AUC scores approaching random guessing (0.5) under strong privacy settings (e.g., $\epsilon = 10$). This behavior is expected: both LoRA and DP-SGD reduce memorization, and their combination further weakens attack signals. Nonetheless, ICP-MIA-SP consistently outperforms most baselines (including all reference-free methods) across all tested privacy budgets, demonstrating its advantages even under strong DP protection.

VII. DISCUSSION

This section discusses three additional factors affecting ICP-MIA's effectiveness in practical deployments: (1) its applicability under label-only API constraints, (2) the impact of residual memorization when test samples come from the pre-training corpus, and (3) how training dynamics create heterogeneous vulnerability across samples.

TABLE V: MIA AUC with different level DP-SGD

MIA Method	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
Loss Attack	0.5080	0.5083	0.5121
Zlib	0.5012	0.5013	0.5064
Min-K%	0.5184	0.5189	0.5228
Min-K%++	0.5223	0.5225	0.5285
Neighborhood	0.5075	0.5080	0.5118
Recall	0.5150	0.5190	0.5232
Reference Attack(Ref)	0.5160	0.5200	0.5248
SPV-MIA	**0.5244**	0.5302	0.5332
ICP-MIA-SP	0.5235	**0.5310**	**0.5367**
ICP-MIA-REF	0.5112	0.5208	0.5244

A. Label-only Attack

While our threat model focuses on models that expose token-level log-probabilities, many commercial LLM APIs, including OpenAI (GPT series) and Anthropic (Claude), only expose label-level or text-only outputs. Although label-only attacks are not the primary focus of this work, recent research has shown promising approaches under this setting. In particular, PETAL [72] approximates log-probabilities from token-level semantic similarity using a surrogate model and a learned regression, which enables effective membership inference without probability access. ICP-MIA naturally supports such adaptation: our method requires only a conditional scoring mechanism and does not rely on gradients or parameter updates. By replacing log-probability changes with semantic similarity–based scores, our in-context probing mechanism can operate in label-only scenarios.

We adapted ICP-MIA-SP using the PETAL framework. As shown in Table VI, our method consistently outperforms PETAL in AUC across all datasets on LLaMA-3.2-3B-Instruct, demonstrating that ICP-MIA remains effective even when restricted to label-only access.

TABLE VI: MIA AUC in Label-Only Setting

MIA Method	Healthcare	MedInstruct	CNN-DM
PETAL	0.7354	0.7756	0.9018
ICP-MIA-Ref	0.7861	0.7632	0.7661
ICP-MIA-SP	**0.9143**	**0.7812**	**0.9351**

B. The Impact of Test Samples Overlapping with the Pre-training Corpus

Our threat model assumes that the fine-tuning data is private and disjoint from the model's pre-training corpus, which is typical in many real-world deployments. However, corner cases may arise when the fine-tuning data domain partially overlaps with the pre-training data. If a test sample appears in both datasets, the model effectively "sees" it twice—once during pre-training and again during fine-tuning—which can artificially increase the true positive rate. The more concerning case is when a sample appears only in the pre-training corpus, not members of our targeted fine-tuning dataset. Such residual memorization can increase the false positive rate by making

pre-trained non-members look more like fine-tuning members. We focus on this latter scenario.

We test ICP-MIA-SP on Pythia-2.8B-deduped with Healthcaremagic dataset by progressively replacing portions of the 500 non-member test samples with data from WikiMIA [5] (constructed from training data of Pythia models). As shown in Figure 9, attack performance degrades only marginally as the proportion of pre-trained samples increases. This is consistent with our expectation, and also indicates that the effect of residual memorization is not strong enough to undermine ICP-MIA in practice.
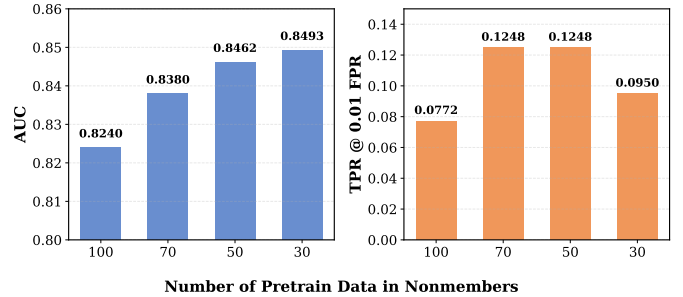

Fig. 9: The Impact of Residual Memorization of Non-Members

C. The Impact of Training Dynamics on MIA Vulnerability

ICP-MIA is motivated by the Optimization Gap induced by the fine-tuning process. A natural question is therefore: how do training dynamics shape sample-level vulnerability? We investigate two key factors—training order and learning rate schedule—to understand how different configurations produce heterogeneous susceptibility to membership inference.

Experimental Setting We partition the training set equally into 10 subsets, denoted $D_0, D_1, \ldots, D_9$. Within each epoch, these partitions are presented to training in a fixed sequence $(D_0 \rightarrow D_9)$. After each epoch, we save a model checkpoint. For each partition D_k, we compute an AUC score for each partition D_k on a balanced test set (2,000 members from D_k and 2,000 non-members). We use two different learning rate schedules: fixed and cosine annealing.

Training under Fixed Learning Rate. Under a fixed learning rate, we observe a strong recency effect: vulnerability increases with the partition index in the training sequence. As shown in Appendix Figure 12a, the final partitions (e.g., D_9) are substantially more vulnerable than early ones, since updates to later partitions are overwritten less and therefore leave a stronger imprint on the final model.

Training under Cosine Annealing Learning Rate. Using a cosine-annealing schedule yields more complex vulnerability patterns (Appendix Figure 12b). In a single epoch, vulnerability continues to rise even after the learning rate reaches its peak, as both recency effects and still-high learning rates jointly contribute to memorization. As the learning rate further decreases, vulnerability drops sharply because updates to later partitions become too small to induce memorization. Extending training to two epochs produces a similar non-monotonic,

single-peaked trend driven by the interplay between recency and the decaying learning rate. Additional ablations using shuffled partition orders (Appendix Figure 10) confirm that these patterns are driven by training dynamics rather than properties of specific data subsets.

VIII. CONCLUSION

In this paper, we propose ICP-MIA, a novel membership inference framework for fine-tuned LLMs that leverages the optimization gap between member and non-member samples. By introducing ICP as a training-free proxy for measuring residual learning potential, we designed a principled attack that operates effectively in black-box settings. Our framework supports both reference-based and reference-free probing strategies, enabling strong performance even without access to auxiliary data. Extensive experiments demonstrate that ICP-MIA achieves state-of-the-art results under strict false-positive constraints and offers actionable insights for privacy auditing in real-world LLM deployments.

ACKNOWLEDGMENT

IX. ETHICS CONSIDERATIONS

This study adheres to responsible disclosure and ethical AI research principles. Our membership inference attacks are conducted on publicly available datasets and models, strictly for the purpose of evaluating and improving model privacy. We do not attempt to deanonymize or identify any individuals behind the data samples. The goal is to highlight potential vulnerabilities and promote the development of privacy-preserving techniques. All experiments were performed in controlled environments, and no real user data or proprietary models were accessed without authorization.

REFERENCES

[1] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[2] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023.

[3] T. Savage, S. P. Ma, A. Boukil, E. Rangan, V. Patel, I. Lopez, and J. Chen, "Fine-tuning methods for large language models in clinical medicine by supervised fine-tuning and direct preference optimization: Comparative evaluation," *J Med Internet Res*, vol. 27, no. e76048, p. e76048, 2025.

[4] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson *et al.*, "Extracting training data from large language models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2633–2650.

[5] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer, "Detecting pretraining data from large language models," in *The Twelfth International Conference on Learning Representations*, 2023.

[6] G. Kim, Y. Li, E. Spiliopoulou, J. Ma, M. Ballesteros, and W. Y. Wang, "Detecting training data of large language models via expectation maximization," *arXiv preprint arXiv:2410.07582*, 2024.

[7] D. Yu, S. Naik, A. Backurs, S. Gopi, H. A. Inan, G. Kamath, J. Kulkarni, Y. T. Lee, A. Manoel, L. Wutschitz *et al.*, "Differentially private fine-tuning of language models," *arXiv preprint arXiv:2110.06500*, 2021.

[8] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[9] D. Chen, N. Yu, Y. Zhang, and M. Fritz, "Gan-leaks: A taxonomy of membership inference attacks against generative models," in *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, 2020, pp. 343–362.

[10] H. Liu, Y. Wu, Z. Yu, and N. Zhang, "Please tell me more: Privacy impact of explainability through the lens of membership inference attack," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2024, pp. 120–120.

[11] T. Matsumoto, T. Miura, and N. Yanai, "Membership inference attacks against diffusion models," in *2023 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2023, pp. 77–83.

[12] T. Wang, Q. Yang, K. Zhu, J. Wang, C. Su, and K. Sato, "Lds-fl: Loss differential strategy based federated learning for privacy preserving," *IEEE Transactions on Information Forensics and Security*, 2023.

[13] S. Rezaei, Z. Shafiq, and X. Liu, "Accuracy-privacy trade-off in deep ensemble: A membership inference perspective," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 364–381.

[14] M. Kurmanji, P. Triantafillou, J. Hayes, and E. Triantafillou, "Towards unbounded machine unlearning," *Advances in neural information processing systems*, vol. 36, pp. 1957–1987, 2023.

[15] F. Mireshghallah, K. Goyal, A. Uniyal, T. Berg-Kirkpatrick, and R. Shokri, "Quantifying privacy risks of masked language models using membership inference attacks," *arXiv preprint arXiv:2203.03929*, 2022.

[16] M. Kazmi, H. Lautraite, A. Akbari, M. Soroco, Q. Tang, T. Wang, S. Gambs, and M. Lécuyer, "Panoramia: Privacy auditing of machine learning models without retraining," *arXiv preprint arXiv:2402.09477*, 2024.

[17] Z. Wang, C. Zhang, Y. Chen, N. Baracaldo, S. R. Kadhe, and L. Yu, "Membership inference attacks as privacy tools: Reliability, disparity and ensemble," in *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '25. Association for Computing Machinery, 2025, p. 1724–1738.

[18] F. Mireshghallah, A. Uniyal, T. Wang, D. K. Evans, and T. Berg-Kirkpatrick, "An empirical analysis of memorization in fine-tuned autoregressive language models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 1816–1826.

[19] M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, and H. Hajishirzi, "Do membership inference attacks work on large language models?" in *First Conference on Language Modeling*, 2024.

[20] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1897–1914.

[21] J. Zhang, J. Sun, E. Yeats, Y. Ouyang, M. Kuo, J. Zhang, H. F. Yang, and H. Li, "Min-k%++: Improved baseline for pre-training data detection from large language models," in *The Thirteenth International Conference on Learning Representations*, 2025.

[22] R. Xie, J. Wang, R. Huang, M. Zhang, R. Ge, J. Pei, N. Gong, and B. Dhingra, "Recall: Membership inference via relative conditional log-likelihoods," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 8671–8689.

[23] M. Mosbach, M. Andriushchenko, and D. Klakow, "On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines," in *International Conference on Learning Representations*, 2020.

[24] T. Domhan, J. T. Springenberg, and F. Hutter, "Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves." in *IJCAI*, vol. 15, 2015, pp. 3460–8.

[25] T. Viering and M. Loog, "The shape of learning curves: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7799–7819, 2022.

[26] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, and T. Jiang, "Membership inference attacks against fine-tuned large language models via self-prompt calibration," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[27] Z. Huang, Y. Liu, D. He, and Y. Li, "Df-mia: A distribution-free membership inference attack on fine-tuned large language models," in

Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 1, 2025, pp. 343–351.

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

[29] W. Zhang, R. Zhang, J. Guo, M. de Rijke, Y. Fan, and X. Cheng, "Pretraining data detection for large language models: A divergence-based calibration method," in *EMNLP*, 2024.

[30] J. Mattern, F. Mireshghallah, Z. Jin, B. Schölkopf, M. Sachan, and T. Berg-Kirkpatrick, "Membership inference attacks against language models via neighbourhood comparison," *arXiv preprint arXiv:2305.18462*, 2023.

[31] C. Wang, Y. Wang, B. Hooi, Y. Cai, N. Peng, and K.-W. Chang, "Con-recall: Detecting pre-training data in llms via contrastive decoding," in *Proceedings of the 31st International Conference on Computational Linguistics*, 2025, pp. 1013–1026.

[32] Z. Li, Y. Wu, Y. Chen, F. Tonin, E. A. Rocamora, and V. Cevher, "Membership inference attacks against large vision-language models," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 2024, pp. 98 645–98 674.

[33] Y. Pang and T. Wang, "Black-box membership inference attacks against fine-tuned diffusion models," in *32nd Annual Network and Distributed System Security Symposium (NDSS) 2025, San Diego, California, USA, February 24–28, 2025*. The Internet Society, 2025.

[34] D. Dai, Y. Sun, L. Dong, Y. Hao, S. Ma, Z. Sui, and F. Wei, "Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers," in *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019.

[35] E. Akyürek, D. Schuurmans, J. Andreas, T. Ma, and D. Zhou, "What learning algorithm is in-context learning? investigations with linear models," in *The Eleventh International Conference on Learning Representations*, 2022.

[36] B. Chen, X. Li, Y. Liang, Z. Shi, and Z. Song, "Bypassing the exponential dependency: Looped transformers efficiently learn in-context by multi-step gradient descent," in *The 28th International Conference on Artificial Intelligence and Statistics*, 2024.

[37] Z. Zhou, X. Lin, X. Xu, A. Prakash, D. Rus, and B. K. H. Low, "Detail: Task demonstration attribution for interpretable in-context learning," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[38] A. Amini and M. Ciaramita, "In-context probing: Toward building robust classifiers via probing large language models," *arXiv preprint arXiv:2305.14171*, 2023.

[39] C. Jiao, W. Gao, A. Raghunathan, and C. Xiong, "On the feasibility of in-context probing for data attribution," in *Findings of the Association for Computational Linguistics: NAACL 2025*. Association for Computational Linguistics, Apr. 2025, pp. 5140–5155.

[40] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[41] S.-Y. Liu, C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, and M.-H. Chen, "Dora: Weight-decomposed low-rank adaptation," in *Forty-first International Conference on Machine Learning*, 2024.

[42] S. Hayou, N. Ghosh, and B. Yu, "Lora+: Efficient low rank adaptation of large models," *arXiv preprint arXiv:2402.12354*, 2024.

[43] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021, pp. 3045–3059.

[44] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.

[45] X. Liu, K. Ji, Y. Fu, W. L. Tam, Z. Du, Z. Yang, and J. Tang, "P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks," 2022.

[46] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in neural information processing systems*, vol. 36, pp. 10 088–10 115, 2023.

[47] M. Huerta-Enochian and S. Y. Ko, "Instruction fine-tuning: Does prompt loss matter?" in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2024, pp. 22 771–22 795.

[48] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with pagedattention," in *Proceedings of the 29th symposium on operating systems principles*, 2023, pp. 611–626.

[49] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.

[50] B. Bordelon, A. B. Atanasov, and C. Pehlevan, "A dynamical model of neural scaling laws," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, 2024.

[51] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark *et al.*, "Training compute-optimal large language models," *arXiv preprint arXiv:2203.15556*, 2022.

[52] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[53] A. Komatsuzaki, "One epoch is all you need," *arXiv preprint arXiv:1906.06669*, 2019.

[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019, pp. 4171–4186.

[55] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith, "Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping," *arXiv preprint arXiv:2002.06305*, 2020.

[56] F. Xue, Y. Fu, W. Zhou, Z. Zheng, and Y. You, "To repeat or not to repeat: Insights from scaling llm under token-crisis," *Advances in Neural Information Processing Systems*, vol. 36, pp. 59 304–59 322, 2023.

[57] S. Biderman, H. Schoelkopf, Q. G. Anthony *et al.*, "Pythia: A suite for analyzing large language models across training and scaling," in *International Conference on Machine Learning*. PMLR, 2023, pp. 2397–2430.

[58] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[59] Y. Li, Z. Li, K. Zhang, R. Dan, S. Jiang, and Y. Zhang, "Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge," *Cureus*, vol. 15, no. 6, 2023.

[60] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," *Advances in neural information processing systems*, vol. 28, 2015.

[61] X. Zhang, C. Tian, X. Yang, L. Chen, Z. Li, and L. R. Petzold, "Alpacare: Instruction-tuned large language models for medical application," *arXiv preprint arXiv:2310.14558*, 2023.

[62] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Oct.-Nov. 2018, pp. 1797–1807.

[63] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 649–657.

[64] M. Meeus, I. Shilov, S. Jain, M. Faysse, M. Rei, and Y.-A. de Montjoye, "Sok: Membership inference attacks on llms are rushing nowhere (and how to fix it)," in *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 2025, pp. 385–401.

[65] Qwen, A. Yang, B. Yang, B. Zhang *et al.*, "Qwen2.5 technical report," 2025.

[66] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford *et al.*, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024.

[67] OpenAI, "Introducing gpt-4.1 in the api (incl. gpt-4.1 mini)," OpenAI blog / documentation, 2025.

[68] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford alpaca: An instruction-following llama model," 2023.

[69] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, and J. Z. Kolter, "TOFU: A task of fictitious unlearning for LLMs," in *First Conference on Language Modeling*, 2024.

[70] C. Dwork and A. Roth, *The Algorithmic Foundations of Differential Privacy*, ser. Foundations and Trends in Theoretical Computer Science. Now Publishers Inc., 2014, vol. 9, no. 3–4.

15

[71] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[72] Y. He, B. Li, L. Liu, Z. Ba, W. Dong, Y. Li, Z. Qin, K. Ren, and C. Chen, "Towards label-only membership inference attack against pre-trained large language models," in *USENIX Security*, 2025.

APPENDIX

A. Implement Details

a) Fine-tuning Details.: We performed our fine-tuning experiments using the LLaMA Factory framework. We employed a full fine-tuning paradigm with the following hyperparameters: a maximum sequence length of 2048 tokens, a learning rate of 2e-4 for Healthcaremagic; 1e-5 for MedInstruct and CNN-DM with a cosine learning rate scheduler, and a warmup ratio of 0.1. Models were trained for 2 epochs with a global batch size of 16. All training was conducted on a single NVIDIA H100 GPU. Following standard practice for SFT, the loss was calculated exclusively on the response tokens; the input prompt tokens were masked out and did not contribute to the loss.

b) Dataset Details.: For each of the three datasets, we designated 80% of the data for training (the member set), 10% as the test set (the non-member set), and the remaining 10% as a validation set. This validation set was used either as a reference pool for attacks like `ICP-MIA-Ref` or to train reference models for baseline comparisons. For each attack evaluation, we constructed a balanced test cohort by randomly sampling 1,000 data points, consisting of 500 members from the training set and 500 non-members from the test set.

c) Attack Configurations.: Our fine-tuning follows an instruction-following format. Consequently, all attacks are evaluated based on the conditional log-likelihood of the response given the prompt. For the baseline methods, we used the following configurations:

- `Min-K%` and `Min-K%++`: We adopted the default setting of $k = 20$ from their original papers.
- `ReCaLL`: To ensure a fair comparison, the `ReCaLL` baseline was configured to use the same prefix pool as our `ICP-MIA-Ref` method—the Dolly-15k instruction dataset. Following their official implementation, we used 7-shot prefixes.
- `Neighborhood Attack`: We set the hyperparameters as follows: roberta-base as masked language model, neighbors=20, and top_k=5.
- `Reference Attack`: We consider two configurations of the Reference Attack. Reference Attack (Base) uses the original pre-trained model as the reference, requiring no additional data. Reference Attack (Ref) constructs the reference model by fine-tuning on a held-out dataset sampled from the same distribution as the target's training data. This held-out set constitutes 10% of the original dataset and maintains strict separation from both training and test partitions. The membership score is computed as $LL_{target}(y|x) - LL_{ref}(y|x)$. We assume that the attacker has no knowledge about the architecture of the target

model. For all experiments requiring training a reference model, we use Qwen3-0.6B as the base model.
- `SPV-MIA`: We use the same setup as the original work: T5-large for masking (span length=2, mask ratio=30%), generating 10 perturbations per sample. The reference model is trained on the held-out set with same hyperparameters as the target model.
- `PETAL`: We use the pretrained model of the target model as the surrogate model. We employ greedy decoding with a maximum generation length of 64 tokens and compute semantic similarity between predicted and ground-truth tokens using sentence-transformers/all-MiniLM-L6-v2.

B. Experiments on Additional Datasets

We conducted additional experiments on XSum [62] and AG News [63], two public benchmarks commonly used in prior MIA evaluations [26], [30]. We formulated XSum as a summarization task and AG News as a news completion task, performing full fine-tuning on LLaMA-3.2-3B, LLaMA-3.2-3B-Instruct, and Pythia-2.8B-deduped for 2 epochs with a learning rate of 1e-5 and cosine annealing schedule. For both ICP-MIA variants, we use $K = 20$ candidate probe contexts.

As shown in Table VII, ICP-MIA-SP outperforms all existing reference-free methods and also the Reference Attack across all configurations in AUC, while achieving performance very close to SPV-MIA despite requiring no reference-model training. On XSum with LLaMA-3.2-3B-Instruct, ICP-MIA-SP attains an AUC of 0.934, close to SPV-MIA (0.945) and well above baselines such as Min-K% (0.744) and ReCaLL (0.849). On AG News, ICP-MIA-SP reaches 0.905, slightly surpassing SPV-MIA (0.903) and substantially outperforming all reference-free baselines.

TABLE VII: MIA AUC on XSum and AG News

MIA Method	LLaMA-3.2-3B-Instruct		LLaMA-3.2-3B		Pythia-2.8B-deduped	
	XSum	AGNews	XSum	AGNews	XSum	AGNews
Loss Attack	0.765	0.594	0.800	0.612	0.780	0.658
Zlib	0.755	0.586	0.788	0.603	0.767	0.661
Min-K%	0.744	0.603	0.799	0.617	0.767	0.659
Min-K%++	0.694	0.570	0.730	0.574	0.703	0.604
Neighborhood	0.612	0.580	0.604	0.581	0.617	0.601
Recall	0.849	0.743	0.884	0.733	0.859	0.713
Reference	0.883	0.832	0.865	0.844	0.831	0.786
SPV-MIA	0.945	0.903	0.920	0.901	0.892	0.870
ICP-MIA-Ref	0.839	0.797	0.802	0.767	0.774	0.753
ICP-MIA-SP	0.934	0.905	0.941	0.909	0.885	0.879

C. Example of a masking-based context probe

x_{prompt} : Determine if the described symptoms relate to cystic fibrosis based on provided genetic information.
x_{input} : The patient exhibits regular bouts of persistent coughing, recurrent lung infections, and difficulty...
y : The described symptoms of regular bouts of persistent coughing, recurrent lung infections, and difficulty in...
C : The described symptoms of regular [MASK] of [MASK] [MASK] recurrent lung [MASK] and [MASK] in...

TABLE VIII: Performance of ICP-MIA with default setting vs. $K = 20$

MIA Method	AUC								
	LLaMA-3.2-3B-Instruct			LLaMA-3.2-3B			Pythia-2.8B-deduped		
	Healthcare	MedInstruct	CNN-DM	Healthcare	MedInstruct	CNN-DM	Healthcare	MedInstruct	CNN-DM
ICP-MIA-Ref(default)	0.827	0.838	0.890	0.842	0.775	0.837	0.850	0.746	0.706
ICP-MIA-SP(default)	0.942	0.959	0.965	0.763	0.977	0.927	0.853	0.882	0.845
ICP-MIA-Ref($K = 20$)	0.821	0.855	0.869	0.845	0.789	0.834	0.836	0.751	0.741
ICP-MIA-SP($K = 20$)	0.948	0.962	0.968	0.796	0.978	0.936	0.871	0.880	0.852

MIA Method	TPR@1%FPR								
	LLaMA-3.2-3B-Instruct			LLaMA-3.2-3B			Pythia-2.8B-deduped		
	Healthcare	MedInstruct	CNN-DM	Healthcare	MedInstruct	CNN-DM	Healthcare	MedInstruct	CNN-DM
ICP-MIA-Ref(default)	0.084	0.044	0.020	0.140	0.018	0.062	0.110	0.074	0.022
ICP-MIA-SP(default)	0.172	0.326	0.518	0.070	0.538	0.418	0.122	0.270	0.144
ICP-MIA-Ref($K = 20$)	0.114	0.116	0.018	0.146	0.022	0.008	0.124	0.152	0.044
ICP-MIA-SP($K = 20$)	0.215	0.410	0.696	0.069	0.604	0.374	0.200	0.324	0.140

D. Context Probe Example for ICP-MIA-REF

Instruction: If you are a doctor, please answer the medical questions based on the patient's description.
Question: I woke up this morning feeling the whole room is spinning when i was sitting down. I went to the bathroom walking unsteadily, as i tried to focus i feel nauseous. I try to vomit but it wont come out.. After taking panadol and sleep for few hours, i still feel the same....
Answer: Hi, Thank you for posting your query. The most likely cause for your symptoms is benign paroxysmal positional vertigo (BPPV), a type of peripheral vertigo. In this condition, the most common symptom is dizziness or giddiness, which is made worse with movements. ...

In-Context Probe:
Instruction: "If you are a doctor, please answer the medical questions based on the patient's description."
Question: "Hello doctor, After unsafe exposure, I got 49 days ELISA antibody test done, 71 days HIV proviral DNA PCR test, 87 days ELISA antibody test. All were negative. The antibody test I took is not a fourth generation test. Is it conclusive or should I take another test?...."
Answer: "Hi. Your tests are conclusive and you are not infected. No need for further tests. Ear ache and tongue papilla are not due to HIV and may be a simple bacterial infection...."

E. Prompt for perturbation generation

System: "You are a precise editor. Given the original text, generate a new text in which exactly 20 words are changed (added, removed, or replaced), but the overall meaning remains identical. Do not change more than 20 tokens. Output only the new text." **User:** "Original text:"
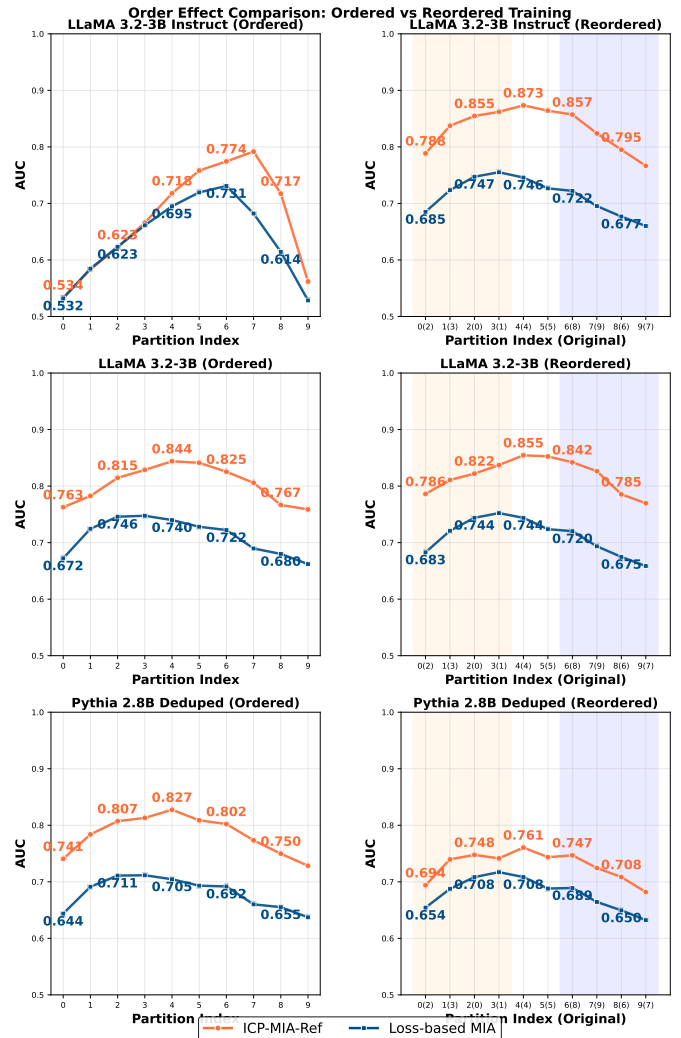

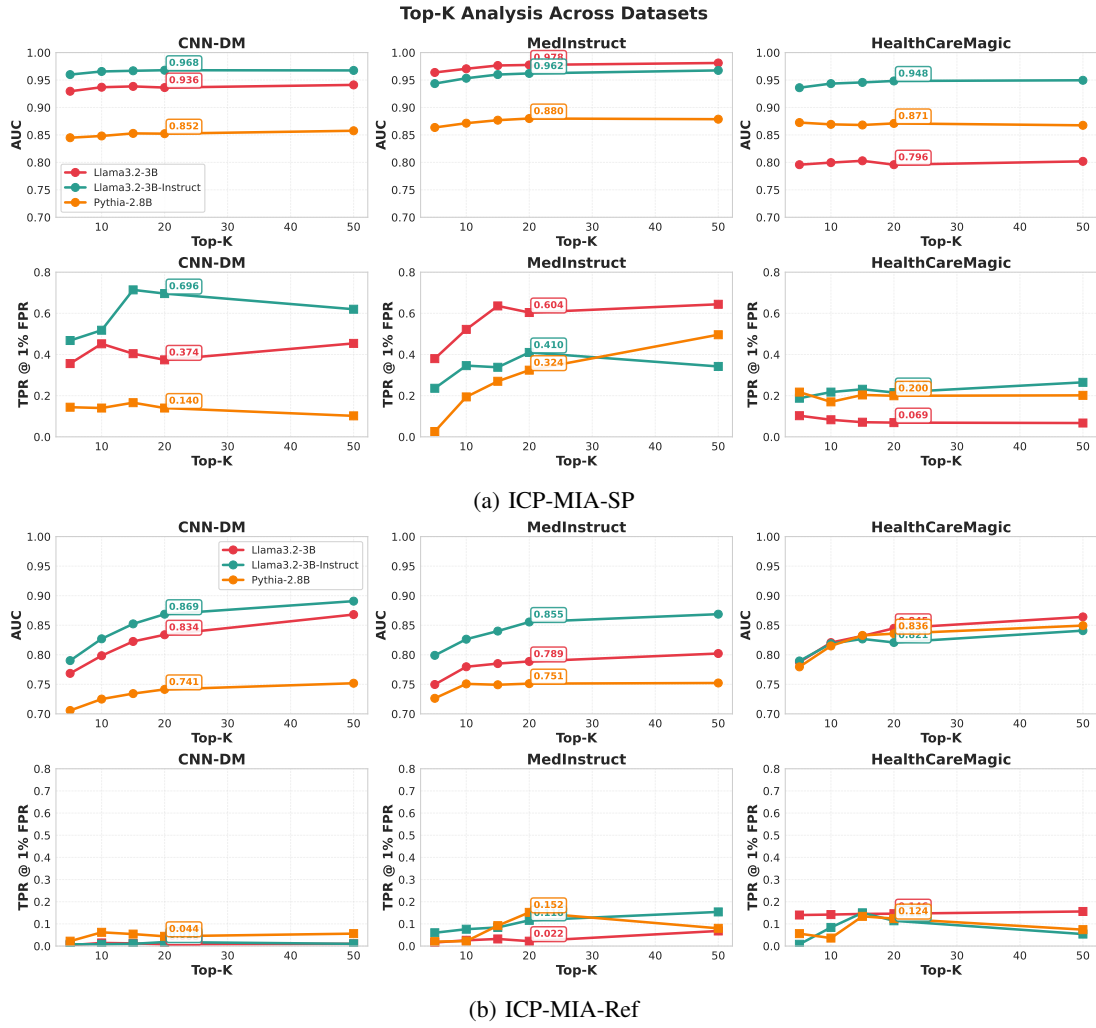Fig. 10: The Impact of Training Sequence

(a) ICP-MIA-SP

(b) ICP-MIA-Ref

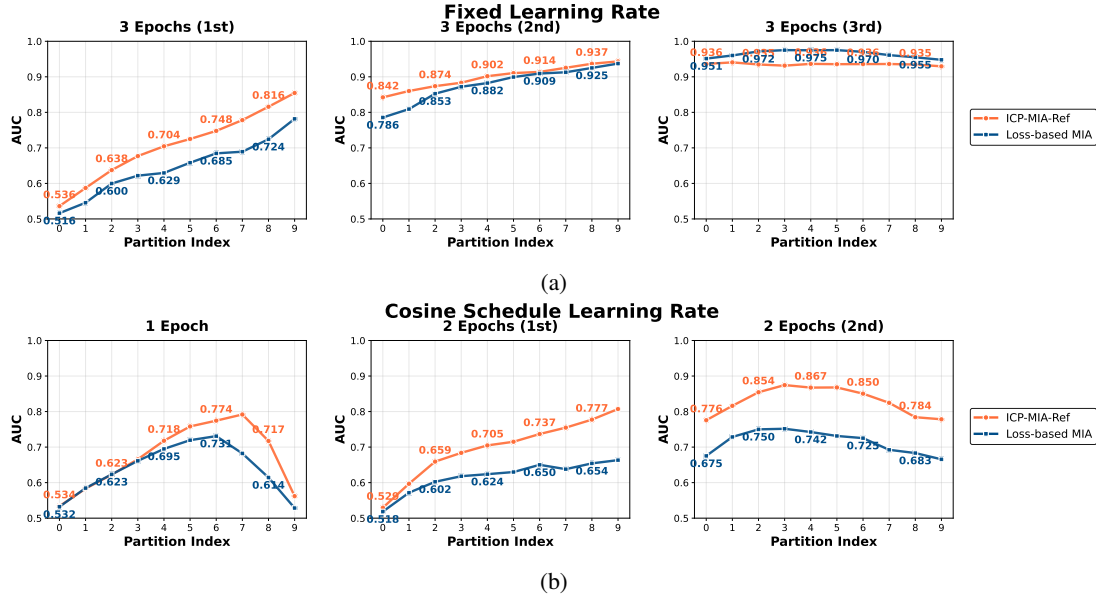Fig. 11: Efficiency analysis on Top K

(a)

(b)

Fig. 12: Training Sequence and Learning Rate Schedule Lead to Different MIA Vulnerability

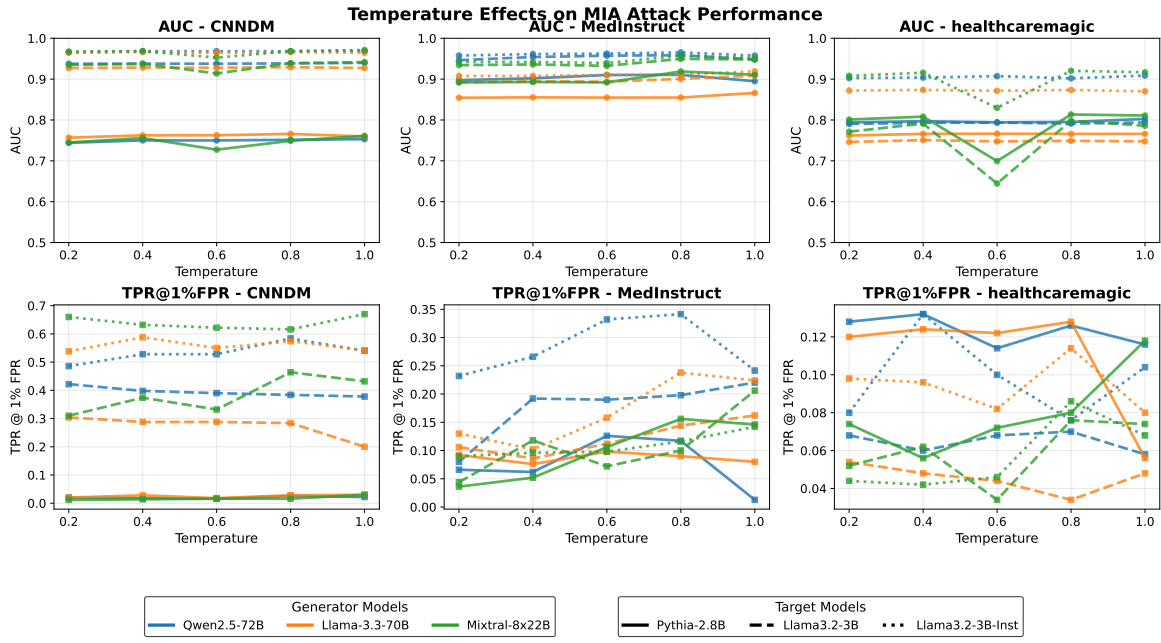Fig. 13: Ablation study on temperature in Prefix Generation

This appendix provides instructions for reproducing the main experimental results of our paper "In-Context Probing for Membership Inference in Fine-Tuned Language Models." Complete documentation is available in the `README.md` file in the repository.

A. Description and Requirements

1) How to Access: https://doi.org/10.5281/zenodo.17906756

2) Hardware Dependencies:

- **Minimal:** NVIDIA A100 (80 GB), 64 GB RAM, 140 GB storage.
- **Recommended:** NVIDIA H100 (80 GB), 128 GB RAM, 200 GB storage (used in paper). Batch size is adjustable for different GPU memory.

3) Software Dependencies:

- **Core:** Python 3.10+, CUDA 12.1+
- **Key packages:** PyTorch 2.5.1, Transformers 4.57.0, Datasets 4.1.1, FAISS-CPU 1.9.0, Sentence-Transformers 5.1.1.

Complete list in `requirements.txt`. LLaMA-Factory is used for fine-tuning.

4) Benchmarks:

- **Primary dataset:** HealthCareMagic-100k.
- **Reference dataset:** Dolly-15k.
- **Model:** LLaMA-3.2-3B-Instruct (requires HuggingFace account and license).
- **Optional:** CNN-DM, iCliniq, MedInstruct-52k, TOFU.

B. Installation and Configuration

Detailed installation instructions in `README.md`. Summary:

1) Create conda environment with Python 3.10.
2) Install dependencies: `pip install -r requirements.txt`
3) Install LLaMA-Factory (see README).
4) Login to HuggingFace: `huggingface-cli login`

C. Experiment Workflow

The workflow consists of four stages:

1) Data preparation
2) Model fine-tuning (3 epochs; use 2nd epoch checkpoint)
3) Perturbation generation
4) Attack execution

Each stage has a corresponding script with configuration files. Fine-tuning is performed in a `LLamaFactory` environment, while attacks are executed in a separate `ICPMIA` environment.

D. Major Claims

C1: ICP-MIA-SP significantly outperforms baselines (AUC: 0.942 vs ReCaLL: 0.847, Min-K%: 0.837). Validated by E1, Table 1.
C2: High-precision performance (TPR@1%FPR: 0.172 and 0.084). Validated by E1, Table 1.
C3: ICP-MIA-SP is practical and reference-free. Validated by E1, Table 1.
C4: ICP-MIA-Ref achieves competitive performance with general-purpose reference data (AUC: 0.827). Validated by E1, Table 1.

E. Evaluation

1) Experiment E1: Main Results on HealthCareMagic:
Time: 15 human-minutes + 8–9 compute-hours

This experiment reproduces Table 1 results (HealthCareMagic column, LLaMA-3.2-3B-Instruct) and validates claims C1–C4.

a) Preparation: **Step 1** [5 min]: Download and split data.

```
python prepare_data.py \
  --dataset lavita/ChatDoctor-HealthCareMagic-100k
    ↪ \
  --output_dir ./data/healthcaremagic
```

Creates train/val/test splits in `./data/healthcaremagic/`.

Step 2 [2 min]: Copy data files to LLaMA-Factory and add dataset entries to `dataset_info.json`. See `README.md` for details. For HealthCareMagic:

```
"healthcaremagic_train": {"file_name": "
    ↪ healthcaremagic_train.json"},
"healthcaremagic_val":   {"file_name": "
    ↪ healthcaremagic_val.json"},
"healthcaremagic_test":  {"file_name": "
    ↪ healthcaremagic_test.json"}
```

Step 3 [∼8 hours]: Fine-tune model for 3 epochs using LLaMA-Factory.

```
cd LLaMA-Factory
llamafactory-cli train ../config/
    ↪ config_training_Healthcare.yaml
```

The model checkpoints are saved in the `saves/` directory. **Use the 2nd epoch checkpoint** (e.g., `checkpoint-XXX`) as the target model for the attack.

Step 4 [5 min]: Generate perturbations for the attack dataset.

```
python generate_perturbations.py convert \
  --input ./data/healthcaremagic/
    ↪ healthcaremagic_attack.json \
  --output ./data/healthcaremagic/
    ↪ healthcaremagic_attack_perturbed.json \
  --mask_rate 0.7 --num_perturbations 20
```

b) Execution: **Step 5** [∼20 min]: Update `target_model_path` in config files to point to the 2nd epoch checkpoint, then run attacks.

```
# Self-perturbation ICP-MIA
python icp_mia_attack.py \
  --config config/config_icp_sp_healthcare.yaml

# Similarity-based ICP-MIA
python icp_mia_attack.py \
  --config config/config_icp_ref_Healthcare.yaml
```

For MedInstruct, use `config_icp_sp_MedInstruct.yaml` and `config_icp_ref_MedInstruct.yaml` instead.

c) Results: Results are saved in `./results/`:

- `{exp_name}_results.csv`: summary metrics (AUC, TPR@FPR).
- `{exp_name}_{attack}_detailed_scores.json`: raw scores.

Compare AUC and TPR@1%FPR with Table 1 to validate claims C1–C4.

2) Optional: Baseline Comparison: **Time:** 30 min

Run baseline attacks to reproduce baseline columns in Table 1:

```
python baseline/main.py \
  -c baseline/config/config_healthcaremagic.yaml \
  --attacks loss minkprob minkplusplus zlib
    ↪ reference \
  --output baseline_results.pkl
```

3) Optional: Extended Evaluation: The artifact supports additional experiments (other datasets, ablation studies). Modify configuration files as described in `README.md`. These are optional and not required for validating core claims.

F. Notes

- **Time:** ∼8–9 hours on H100 (mostly fine-tuning).
- **Disk space:** ∼140 GB.
- **Reproducibility:** All experiments use `random_seed: 42`. Minor variations may occur due to hardware differences.
- **Troubleshooting:** See `README.md` for common issues.