Ensemble Privacy Defense for Knowledge-Intensive LLMs against Membership Inference Attacks

Haowei Fu[1]* **Bo Ni**[1]* **Han Xu**[2] **Kunpeng Liu**[3] **Dan Lin**[1] **Tyler Derr**[1]

[1]Vanderbilt University [2]University of Arizona [3]Clemson University

{haowei.fu, bo.ni, dan.lin, tyler.derr}@vanderbilt.edu

hanxu@arizona.edu, kungpeng@clemson.edu

Abstract

Retrieval-Augmented Generation (RAG) and Supervised Finetuning (SFT) have become the predominant paradigms for equipping Large Language Models (LLMs) with external knowledge for diverse, knowledge-intensive tasks. However, while such knowledge injection improves performance, it also exposes new attack surfaces. Membership Inference Attacks (MIAs), which aim to determine whether a given data sample was included in a model's training set, pose serious threats to privacy and trust in sensitive domains. To this end, we first systematically evaluate the vulnerability of RAG- and SFT-based LLMs to various MIAs. Then, to address the privacy risk, we further introduce a novel, model-agnostic defense framework, Ensemble Privacy Defense (EPD), which aggregates and evaluates the outputs of a knowledge-injected LLM, a base LLM, and a dedicated judge model to enhance resistance against MIAs. Comprehensive experiments show that, on average, EPD reduces MIA success by up to 27.8% for SFT and 526.3% for RAG compared to inference-time baseline, while maintaining answer quality. Our code will be made available public at https://github.com/RageFu2004/Ensemble-Privacy-Defense.

1 Introduction

Large language models (LLMs) have become the foundation of modern natural language processing, powering applications across diverse domains (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020). For knowledge-intensive tasks, two predominant paradigms inject external knowledge into LLMs: retrieval-augmented generation (RAG) and supervised fine-tuning (SFT) (Ovadia et al., 2024). RAG retrieves evidence at inference time and conditions generation on the retrieved passages (Lewis et al., 2020), whereas SFT adapts a pretrained model to a downstream task using task-specific data (Ouyang et al., 2022). Both approaches substantially improve factuality, relevance, and adaptability. As knowledge-intensive LLMs move into high-stakes settings, however, privacy risks have become paramount (Ni et al., 2025; Hu et al., 2022b). A key threat is membership inference attacks (MIAs), which aim to determine whether a particular example was used to train or condition the model (Yeom et al., 2018; Carlini et al., 2021). RAG and SFT create different attack surfaces (Anderson et al., 2025): SFT updates parameters, potentially tightening the coupling between specific fine-tuning data and model outputs; RAG introduces a pipeline to the external corpus where the retrieved contents can leak through generation. In practice, RAG often appears more MIA-resistant because knowledge resides primarily in a non-parametric index, retrieval adds stochasticity, and conditioning on heterogeneous evidence raises output entropy—reducing the separability between member and non-member scores that many MIAs exploit.

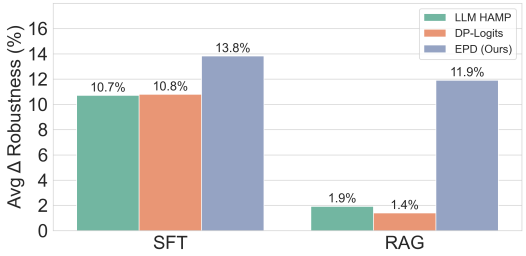


Figure 1: Comparison of improved robustness on MIAs across datasets. Our proposed method can significantly reduce the risk of privacy leakage under MIAs.

Prior work has studied MIAs and defenses for either RAG (Anderson et al., 2025) or SFT (Fu et al., 2023; Huang et al., 2025), but a systematic, head-to-head comparison is lacking, and many defenses are tightly coupled to specific architectures or training objectives (Anderson et al., 2025; Liu et al., 2025;

*Equal contribution.

Zhang et al., 2025), limiting real-world adoption. We address both gaps by (i) providing a controlled comparison of MIA vulnerability across RAG and SFT, and (ii) introducing a training-free, model-agnostic Ensemble Privacy Defense (EPD). Given a query, EPD obtains candidate answers from a knowledge-injected target model and a base model, and uses a judge LLM to select/synthesize the final output. Intuitively, aggregating heterogeneous candidates—and instructing the judge to penalize verbatim phrasing and down-weight abnormally low per-token loss—acts as an entropy-increasing regularizer: it narrows member–non-member likelihood gaps (e.g., LiRA (Carlini et al., 2022)) and weakens tail-token cues (e.g., Min-K (Shi et al., 2023)), reducing MIA effectiveness.

We study two questions: which paradigm—RAG or SFT—is more robust to MIAs, and can a training-free, model-agnostic ensemble mitigate MIAs while preserving answer quality? To answer them, we evaluate RAG and SFT models on multiple QA datasets against seven representative MIAs, and then comprehensively assess EPD defense capability against MIAs. Empirically, RAG generally shows stronger resistance than SFT (with higher latency due to retrieval), and EPD further improves defense abilities in both RAG and SFT. Figure 1 demonstrates that, on average, EPD can successfully improve the robustness of the knowledge-injected models on inference-time, with a 27% improvement for SFT and 526% improvement for RAG. These results reveal an efficiency–privacy trade-off and demonstrate that judgment-guided ensembling of EPD offers a practical path to privacy-preserving deployment. Our contribution can thus be summarized as follows:

- We systematically study MIAs on knowledge-intensive LLMs, analyzing the vulnerabilities of both RAG and SFT paradigms.
- Ensemble Privacy Defense (EPD): a training-free, model-agnostic framework that aggregates a target and base model with a judge LLM to attenuate membership signals while preserving utility.
- Experiments on QA benchmarks against seven MIAs, demonstrating substantial defense capabilities and providing insights for privacy-preserving RAG-/SFT-based LLM deployment.

2 Preliminaries

2.1 Knowledge-Intensive LLMs

LLMs have demonstrated remarkable capabilities across NLP tasks, from text generation and summarization to dialogue systems (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020). However, despite their impressive performance, traditional LLMs rely solely on their parametric memory, which is limited by their training data cutoff date and the inherent constraints of model capacity, leading to deteriorated performance in domain-specific, knowledge-intensive tasks (Ni et al., 2025; Lewis et al., 2020; Hu et al., 2022a). To address these limitations, two predominant paradigms have emerged for equipping LLMs with external knowledge: Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) and Supervised Finetuning (SFT) (Ovadia et al., 2023). These approaches enable LLMs to access external knowledge sources dynamically, thereby enhancing their factual accuracy, reducing hallucination, and improving their applicability to knowledge-intensive tasks.

2.2 Membership Inference Attacks

Here we formally introduce the definition and threat model of MIA on LLMs.

Definition 1 (Membership Inference Attacks)**.** *Let $\mathcal{D}_{train}$ denote the training dataset used to train a large language model (LLM) f_θ, where θ represents the model parameters. Given a query q, the LLM generates a response $a = f_\theta(q)$. Membership inference attacks (MIAs) aim to determine whether a specific data sample x was included in $\mathcal{D}_{train}$, i.e., to infer the membership status $m(x) \in \{0, 1\}$. Formally, an adversary is given black-box access to the LLM and, for a target sample x, attempts to construct an attack function $\mathcal{A}$ such that:*

$$\mathcal{A}(f_\theta, x) \approx m(x)$$

In the context of SFT, $\mathcal{D}_{\text{train}} = \mathcal{D}^{\text{SFT}}_{\text{train}}$, where the training data consists of labeled examples used to adapt a pre-trained model to a downstream task. In the context of RAG, $\mathcal{D}_{\text{train}}$ is replaced by the retrieval corpus $\mathcal{C}$. MIAs here attempt to determine whether a specific document or passage x is contained in $\mathcal{C}$. We assume that the attacker does not have any knowledge of the retrieval model.

Threat Model. We consider a setting where the victim is the large language model f_θ trained either by SFT on a dataset $\mathcal{D}^{\text{SFT}}_{\text{train}}$ or deployed with

RAG using an external knowledge corpus $\mathcal{C}$, and the attacker is an external adversary aiming to determine whether a specific sample x was used in finetuning or whether a document/passage x exists in the retrieval corpus. We assume a black-box setting where the adversary can only query the target model and observe its outputs (e.g., probabilities, likelihoods, or generated text) without access to parameters or training data, but may possess limited auxiliary knowledge such as samples from a similar distribution, domain expertise, or the ability to construct probing queries.

2.2.1 MIA Methods

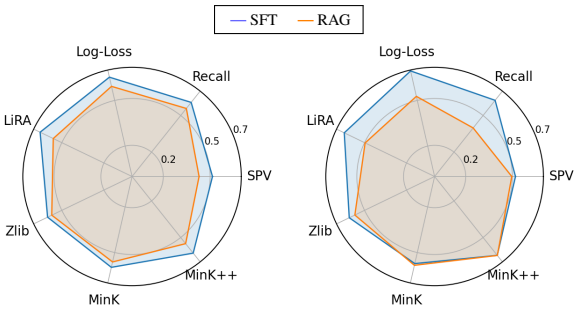
The MIAs can be categorized into two categories: Reference-free attacks and Reference-based attacks, where Reference-free attacks rely solely on the outputs of the target model, whereas reference-based attacks additionally leverage auxiliary datasets or reference models for calibration. We explore the following common MIA methods in the rest of the paper.

Reference-free attacks. Representative methods include *Recall*, which leverages the model's recall or confidence on the queried sample (Xie et al., 2024); *LL (Log-Loss)*, which uses the negative log-likelihood of the model's prediction as a membership signal (Yeom et al., 2018); *Zlib*, which measures the compressibility of the model's output since memorized samples tend to be more compressible (Carlini et al., 2021); and *Min-K/Min-K++*, which focus on the lowest K token losses or their z-score normalization to detect membership signals (Shi et al., 2023; Zhang et al., 2024).

Reference-based attacks. Representative methods include *SPV-MIA (Self-calibrated Probabilistic Variation)*, which constructs a reference dataset by prompting the target LLM itself and introduces a probabilistic variation metric for more reliable membership signals in practical scenarios (Fu et al., 2023); *LiRA* (Carlini et al., 2022) compares the output of the target model to that of a reference model trained on a different dataset.

3 Robustness of RAG vs. SFT under Membership Inference Attacks

To evaluate the robustness of RAG and SFT against membership inference attacks, we benchmark them on a set of common MIAs. We first detail the experimental setup and datasets, then present results and analysis. For fairness, all methods are evaluated under an identical protocol with matched hyperparameters, prompts, and inference budgets; the same preprocessing, attack implementations, and metrics are used unless noted. This controlled design enables a direct comparison of RAG and SFT robustness across a range of MIA variants.



(a) Attack Success Rate (b) AUC Score

Figure 2: Robustness of RAG and SFT, averaged over three datasets, against seven MIAs, shown in terms of (A) attack success rate and (B) AUC score.

3.1 Findings

3.2 Knowledge-Intensive LLM Settings

Supervised Finetuning. We employ Llama-2-7B (Touvron et al., 2023) as the base model for supervised finetuning. For model finetuning, we utilize a Parameter-Efficient Fine-Tuning (PEFT) technique, specifically LoRA (Hu et al., 2022a), to update the model parameters.

Retrieval Augmented Generation. For fair comparison, we use the Llama-2-7B as the LLM for inference. For dense retrieval, we employ the SentenceTransformer (Reimers and Gurevych, 2019) to embed both the input queries and all context passages from the datasets. At inference time, we compute the cosine similarity between the query embedding and all stored context embeddings, retrieving the top-K most similar contexts (with $K = 5$) to construct the final prompt for the LLM.

3.3 Datasets and Metrics

We utilize two widely used datasets for knowledge-intensive LLMs (Ovadia et al., 2023; Kim and Lee, 2024), more specifically, TriviaQA (Joshi et al., 2017) and SQuAD (Rajpurkar et al., 2016). Both question answering datasets have a database constructed from Wikipedia to help answer the queries. The dataset statistics are presented in Table 8, and more details are provided in Appendix C.

For evaluation, we leverage the commonly used metrics for privacy attacks: AUC and ASR. Area

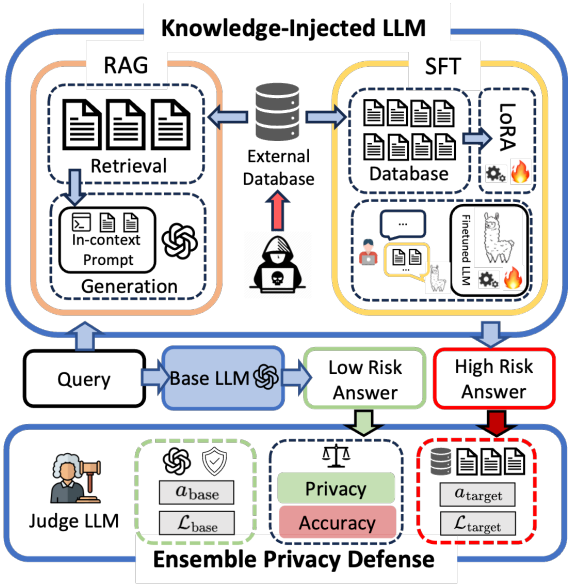


Figure 3: Overview of the proposed EPD framework.

Under Curve (AUC) measures the overall discriminative ability of the MIA attacks by calculating the area under the ROC curve, i.e., the probability that a randomly chosen member receives a higher attack score than a non-member, and Attack Success Rate (ASR) is the proportion of successful membership inference attempts.

We report the average ASR and AUC across the two datasets in Figure 2, with the full detailed results supplemented in Appendix Table 7. Our findings reveal that RAG demonstrates consistently stronger resistance to MIA attacks compared to SFT, particularly in terms of lowering ASR. On Trivia-QA, RAG achieves substantially lower ASR values across nearly all attack methods while maintaining lower overall AUC scores. On SQuAD, RAG also shows advantages under most attack settings. Averaged across both datasets, RAG can reduce ASR and AUC by nearly 10%, demonstrating its robustness as a more privacy-preserving paradigm. The unfinetuned LLM in RAG means that the context are supplemented explicitly, without changing its parameters as in SFT. This helps avoid member data memorization while the retrieved contexts can provide same, or even more relevant information for the downstream generation.

Nevertheless, knowledge-intensive LLMs remain exposed, with ASR consistently exceeds 50% across both RAG and SFT settings. Existing defense strategies are often tightly coupled with model architectures and require additional training or fine-tuning on the base model (Anderson et al., 2025; Liu et al., 2025; Zhang et al., 2025), which significantly limits their practicality and adaptability in real-world deployments. The superior performance of RAG suggests that unmodified LLMs, which rely on explicit retrieval rather than parameter updates, are inherently less prone to data memorization and thus can help mitigate MIA risks.

Motivated by these observations, we leverage the complementary strengths of knowledge-injected models (high task accuracy but higher leakage) and base models (stronger privacy but weaker specialization) to construct a hybrid ensemble defense. We instantiate this idea as Ensemble Privacy Defense (EPD), a lightweight and model-agnostic inference-time defense method that obtains candidate answers from both a knowledge-injected and base model, and leverages an LLM-as-a-judge to integrate their outputs, thereby offering strong post-processing protection against membership inference with no retraining overhead.

4 Ensemble Privacy Defense

In this section, we introduce Ensemble Privacy Defense (EPD), a model-agnostic inference-time framework designed to mitigate membership inference risks in knowledge-intensive LLMs. Unlike existing defenses that require retraining or architectural modifications, EPD can be seamlessly applied to both finetuned and retrieval-augmented models without altering their internal parameters.

As visualized in Figure 3, the intuition is to harness the complementary strengths of a task-specific target model and a more general base model: while the target model delivers high-accuracy answers but is prone to privacy leakage, the base model offers stronger protection at the expense of task specialization. Given a query, EPD generates candidate responses from both models and employs an LLM-as-a-judge to evaluate and integrate them into a privacy-aware final output.

On a high level, let q be a query and a be the generated response. Let $\mathcal{M}_{\text{target}}$ denote a target model (either finetuned or RAG-based) and $\mathcal{M}_{\text{base}}$ denote a base model without task-specific training. EPD is a function $\mathcal{E}$ that produces the answer $a_{\text{final}} = \mathcal{E}(\mathcal{M}_{\text{target}}, \mathcal{M}_{\text{base}}, q)$ where a_{final} exhibits lower membership leakage compared to direct outputs from $\mathcal{M}_{\text{target}}$.

Candidate Answer Generation For each input query q, we generate two candidate answers along

with their corresponding loss values:

$$a_{\text{target}} = \mathcal{M}_{\text{target}}(q),\ a_{\text{base}} = \mathcal{M}_{\text{base}}(q)$$

$$\mathcal{L}_{\text{target}}(q) = -\frac{1}{T_{\text{tar}}} \sum_{t=1}^{T_{\text{tar}}} \log p_{\theta_{\text{target}}}\left(g_t^{\text{tar}} \mid x(q),\ g_{<t}^{\text{tar}}\right)$$

$$\mathcal{L}_{\text{base}}(q) = -\frac{1}{T_{\text{base}}} \sum_{t=1}^{T_{\text{base}}} \log p_{\theta_{\text{base}}}\left(g_t^{\text{base}} \mid x(q),\ g_{<t}^{\text{base}}\right)$$

where a_{target} represents the high-accuracy but potentially privacy-leaking response, and a_{base} represents the privacy-preserving but potentially less accurate response. The loss values $\mathcal{L}_{\text{target}}$ and $\mathcal{L}_{\text{base}}$ quantify model confidence and and potential membership leakage. The loss we report is the standard causal language modeling cross-entropy (token-level negative log-likelihood). Let $g_{\text{target}} = (g_1^{\text{tar}}, \ldots, g_{T_{\text{tar}}}^{\text{tar}})$ be the tokens generated by the target model from q, and $g_{\text{base}} = (g_1^{\text{base}}, \ldots, g_{T_{\text{base}}}^{\text{base}})$ for the base model. We truncate both g_{target} and g_{base} to same length $T_{\text{tar}} = T_{\text{base}}$. We evaluate next-token likelihood on the generated region only.

LLM-as-a-Judge Selection To combine these candidates optimally, we introduce a dedicated judge model $\mathcal{M}_{\text{judge}}$ that evaluates both responses along with their loss values and generates a privacy-aware final answer. The judge model produces the final answer a_{final} with:

$$a_{\text{final}} = \mathcal{M}_{\text{judge}}(\phi(q, a_{\text{target}}, \mathcal{L}_{\text{target}}, a_{\text{base}}, \mathcal{L}_{\text{base}}))$$

where ϕ is a prompt formatting function. ϕ is designed to ensure that the final output preserves the accuracy from the target model while incorporating the privacy advantage of the base model. The specific instantiation of ϕ is in Appendix B.

5 Experiments

In this section, we conduct a comprehensive evaluation of the effectiveness of our proposed Ensemble Privacy Defense (EPD) framework.

5.1 Experimental Setup

Implementation We follow the same methodology outlined in Section 3.2 for both SFT and RAG. The judge model $\mathcal{M}_{\text{judge}}$ leverages the DeepSeek-R1-8B (Guo et al., 2025). Detailed implementation specifications for all models and experimental configurations are provided in Appendix D.

Datasets Here we evaluate on TriviaQA (Joshi et al., 2017) and AG News (Zhang et al., 2015) datasets. Specifically, for AG News, only SFT is evaluated because AG News does not provide retrieved/relevant contexts in its original dataset.

Evaluation Metrics Besides AUC and ASR, we also use TPR1%FPR (shortened to TPR@1 for brevity) for MIA defense evaluation. TPR1%FPR measures the true Positive Rate at 1% False Positive Rate, indicating the attack performance under strict privacy constraints.

5.2 Inference-time Baselines

We include two state-of-the-art baselines for MIA defense. Although recent research has proposed more advanced defense strategies for RAG (Anderson et al., 2025; Liu et al., 2025), we do not include them in our experiment because the MIAs are designed for both SFT and RAG. Moreover, to ensure a fair comparison with our proposed approach, we restrict to inference-time methods that do not require retraining or architectural modifications.

HAMP (Hiding Auxiliary Membership Privacy): An inference-time defense method that enforces less confident predictions by introducing high-entropy soft labels and an entropy-based regularizer (Chen and Pattabiraman, 2023).

DP-Logits: Applies differential privacy to the output logits during inference by adding calibrated noise to the predictions (Rahimian et al., 2020).

5.3 Main Results

In this section, we present our experiment results against the seven representative MIA attack methods introduced in Section 2.2.1. The results are presented in Tables 1, 2, and 4. At the same time, we present our experiment results measuring the final answer accuracy to show that our defense method preserves answer quality in Table 3.

5.3.1 MIA Success Rate Reduction

As shown in Table 1, on TriviaQA with finetuning, our LLM Judge framework substantially reduces both AUC and ASR across most attack methods. The strongest gains are achieved on reference-based attacks: LiRA (49.2% reduction), Log-Loss (21.2%), Recall (18.5%), and Zlib (10.2%). For ASR, the method achieves notable reductions of 18.0% on Log-Loss and 14.9% on Recall, while maintaining competitive results on other MIAs. Averaged across metrics, EPD outperforms the runner-up by 36.2% on robustness improvement.

Table 2 presents results on TriviaQA with RAG. Although RAG already exhibits lower susceptibility to MIAs compared to finetuned models, our framework still improves robustness further, especially on LiRA and Zlib attacks, with 100% and

Table 1: Defense results of SFT-based models against seven membership inference attacks on Trivia-QA. Each cell reports the metric value, with the relative change (%) from the base model (no defense) shown in parentheses.

Defense Method	SPV			LiRA			Recall			Log-Loss		
	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓
No Defense	0.550	0.720	0.000	0.452	0.508	0.010	0.753	0.697	0.212	0.763	0.707	0.135
LLM HAMP	0.480(-13)	0.400(-44)	0.000(0)	0.239(-49)	0.503(-1)	0.005(-50)	0.718(-4)	0.692(-1)	0.057(-71)	0.674(-12)	0.637(-10)	0.083(-39)
DP-Logits	0.480(-13)	0.650(-10)	0.000(0)	0.239(-49)	0.500(-2)	0.030(+189)	0.700(-7)	0.663(-5)	0.119(-44)	0.674(-12)	0.637(-10)	0.083(-39)
EPD (Ours)	0.480(-13)	0.510(-29)	0.000(0)	0.230(-49)	0.500(-2)	0.000(-100)	0.613(-19)	0.593(-15)	0.010(-95)	0.601(-21)	0.580(-18)	0.021(-85)

Defense Method	Zlib			Min-K			Min-K++			Average (across the 7 MIAs)		
	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓
No Defense	0.719	0.674	0.093	0.692	0.655	0.047	0.780	0.731	0.083	0.673	0.670	0.083
LLM HAMP	0.673(-7)	0.635(-6)	0.078(-17)	0.565(-18)	0.554(-17)	0.031(-33)	0.495(-37)	0.521(-29)	0.021(-75)	0.549(-18)	0.563(-16)	0.039(-53)
DP-Logits	0.673(-7)	0.640(-5)	0.078(-17)	0.565(-18)	0.557(-11)	0.031(-33)	0.497(-36)	0.533(-27)	0.026(-69)	0.547(-19)	0.597(-11)	0.052(-37)
EPD (Ours)	0.646(-10)	0.614(-9)	0.026(-72)	0.589(-15)	0.573(-13)	0.026(-44)	0.519(-34)	0.550(-25)	0.017(-80)	**0.525**(-22)	**0.560**(-16)	**0.014**(-83)

Table 2: Defense results of RAG-based models against seven membership inference attacks on Trivia-QA.

Methods	SPV			LiRA			Recall			Log-Loss		
	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓
No Defense	0.491	0.580	0.005	0.377	0.513	0.010	0.386	0.603	0.031	0.599	0.662	0.104
LLM HAMP	0.480(-2)	0.400(-31)	0.000(-100)	0.270(-28)	0.500(-2)	0.010(0)	0.652(+69)	0.619(+3)	0.036(+17)	0.669(+12)	0.635(-4)	0.062(-40)
DP-Logits	0.480(-2)	0.650(+12)	0.000(-100)	0.270(-28)	0.500(-2)	0.030(+189)	0.680(+76)	0.637(+6)	0.042(+33)	0.669(+12)	0.635(-4)	0.062(-40)
EPD (Ours)	0.480(-2)	0.510(-12)	0.000(-100)	0.240(-36)	0.500(-2)	0.000(-100)	0.558(+45)	0.557(-8)	0.036(+17)	0.539(-10)	0.552(-17)	0.026(-75)

Methods	Zlib			Min-K			Min-K++			Average (across the 7 MIAs)		
	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓
No Defense	0.686	0.619	0.104	0.690	0.595	0.036	0.780	0.557	0.041	0.571	0.590	0.047
LLM HAMP	0.669(-3)	0.635(+3)	0.104(0)	0.598(-13)	0.588(-1)	0.031(-14)	0.580(-26)	0.575(+3)	0.005(-88)	0.560(-2)	0.565(-4)	0.035(-25)
DP-Logits	0.667(-3)	0.635(+3)	0.104(0)	0.598(-13)	0.586(-2)	0.021(-43)	0.577(-26)	0.567(+2)	0.005(-88)	0.563(-2)	0.601(+2)	0.038(-21)
EPD (Ours)	0.616(-10)	0.596(-4)	0.067(-35)	0.585(-15)	0.580(-3)	0.016(-57)	0.505(-35)	0.541(-3)	0.010(-75)	**0.503**(-12)	**0.548**(-7)	**0.022**(-53)

75% TPR@1 reduction, respectively. These results highlight that for RAG settings where vulnerability is already reduced, EPD can still effectively improve the privacy robustness. Averaged across metrics, EPD outperforms the runner-up by 232.2% on robustness improvement.

For AG News classification tasks, reported in Table 4, our framework consistently improves privacy protection across most MIAs. Notably, on the TPR@1%FPR metric, the LLM Judge achieves large reductions, including 67% for LiRA attacks and 51% for Min-K++ attacks. Averaged across metrics, EPD outperforms the runner-up by 138.7% on robustness improvement.

Overall, when compared to baseline defense methods across all three tables, our LLM Judge framework, EPD, nearly always outperforms or matches their performance across the MIA scenarios. Across the datasets, EPD enhances AUC by 205.2%, ASR by 23.7%, and TPR@1 by 166.8%. These results demonstrate that the LLM Judge framework provides a robust defense against membership inference attacks, making it a practical solution for privacy-preserving LLM deployment in sensitive applications.

Table 3: Accuracy Comparison of Different Models

Model	Exact Match	F1-Score
Ground Truth	0.3560	0.4466
DP-Logits	0.2690	0.3150
LLM-Hamp	0.2070	0.2874
EPD-Judge Model	**0.3320**	**0.4047**

5.3.2 Final Answer Accuracy

As shown in Table 3, we evaluated the model under SFT settings on TriviaQA using Exact Match (EM) and F1-Score. We run the experiments on 1000 extracted data from TriviaQA's test set compared to the provided ground truth. We report the result in the following table and observe that EPD preserves utility, with EM/F1 fluctuations within a narrow range compared to the original target model and two other defense baselines.

5.4 Ablation Studies

In this section, we conduct an extensive ablation study to investigate EPD's components. We will first explore the impact of Judge Model's capacity. Then we will explore the impact of k in RAG retrieval on the MIA defense. Additionally, we stud-

ied the Judge model's behavior based on answers from Target model and Base model to measure if it has bias towards any of them. Finally, we study adding adaptive noise injection (Wang et al., 2025) as an additional defense mechanism for EPD.

5.4.1 Impact of Judge Model Capacity.

To investigate the dependency on model capacity, we conduct a study comparing two different judge models with varying parameter scales: DeepSeek-R1-1.5B and DeepSeek-R1-8B. Both models are based on the DeepSeek-R1 architecture but differ in their parameter count and computational capacity.

The judge model's role in our framework is to evaluate candidate answers from both the target (finetuned/RAG) and base models, then synthesize a final response that balances accuracy with privacy protection. This requires sophisticated reasoning and we hypothesize that judge models with greater parameter capacity will demonstrate superior performance in this complex decision-making process. Our experimental setup maintains identical configurations for all other framework components, with the judge model capacity being the only variable.

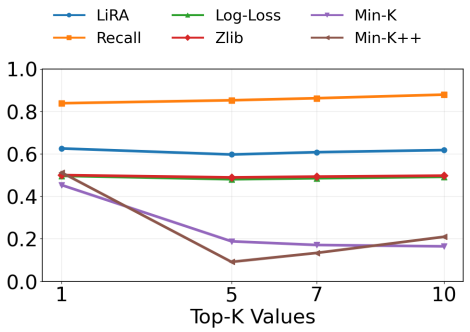
Table 5 presents the comprehensive results comparing the two judge model variants across multiple MIA attack methods and evaluation metrics. The results demonstrate a clear correlation between judge model capacity and defense performance across all attack scenarios.

The results reveal that the larger 8B parameter model consistently outperforms the 1.5B parameter model across all attack methods and evaluation metrics except ASR in SPV-MIA method. The superior performance of the larger judge model can be attributed to several factors. Detailed analysis is provided in Appendix E.

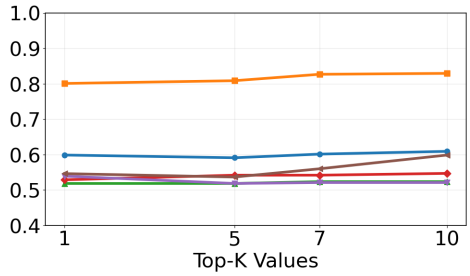
Our results suggest that continued scaling of judge model capacity may yield further improvements in MIA defense effectiveness, though the diminishing returns observed in other domains (Hoffmann et al., 2022; Kaplan et al., 2020) may also apply here. Future work should explore the optimal balance between model capacity and performance.

5.4.2 Impact of Top-K Retrieval on RAG-based MIA Defense

The number of retrieved contexts (k) directly determines how much external knowledge a RAG model conditions on, which in turn can influence its privacy behavior. To examine this relationship, we conduct a systematic ablation study analyzing



(a) AUC



(b) ASR

Figure 4: Effect of different choices of K on the attack performance. Colored lines are the attack methods.

how different top-k retrieval settings affect MIA defense effectiveness in RAG systems.

Specifically, we evaluate four retrieval configurations, $k \in 1, 5, 7, 10$, on the SQuAD dataset using our EPD framework under the RAG setting. Figure 4 summarizes the results across multiple MIA attack methods and evaluation metrics. Overall, EPD remains robust across different values of k. While we observe a slight increase in ASR as k grows, which aligns with intuition, since retrieving more passages can help the target model generate more grounded responses, the magnitude of this increase is small, highlighting the stability of our defense against most realistic retrieval depth.

5.4.3 Judge Model Bias

In EPD it is assumed that the judge model is not designed to reward stylistic patterns, verbosity, or structural preferences from either candidate model. Instead, its role is to aggregate outputs while suppressing signals indicative of membership data. To directly assess whether the judge systematically favors either the target or the base model, we conducted an explicit post-hoc bias evaluation.

As shown in detail within Algorithm 1 of Appendix H, each judge decision was categorized into Target, Base, and Mixed through a hierarchical matching process. To show that the Judge model does not bias towards certain model outputs, we

Table 4: Defense results of SFT-based models against seven membership inference attacks on Ag News.

Methods	SPV			LiRA			Recall			Log-Loss		
	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓
No Defense	0.490	0.580	0.005	0.529	0.513	0.027	0.513	0.603	0.012	0.514	0.662	0.013
LLM HAMP	0.480(-2)	0.400(-31)	0.000(-100)	0.525(-1)	0.530(+3)	0.026(-4)	0.509(-1)	0.508(-16)	0.011(-7)	0.514(0)	0.510(-23)	0.012(-6)
DP-Logits	0.480(-2)	0.650(+12)	0.000(-100)	0.525(-1)	0.530(+3)	0.026(-4)	0.514(0)	0.511(-15)	0.012(-1)	0.514(0)	0.510(-23)	0.012(-6)
EPD (Ours)	0.480(-2)	0.510(-12)	0.000(-100)	0.495(-7)	0.501(-2)	0.009(-68)	0.502(-2)	0.510(-15)	0.012(-3)	0.501(-3)	0.510(-23)	0.013(-2)

Methods	Zlib			Min-K			Min-K++			Average (across the 7 MIAs)		
	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓	AUC↓	ASR↓	TPR@1↓
No Defense	0.515	0.619	0.012	0.514	0.595	0.012	0.514	0.557	0.017	0.513	0.590	0.014
LLM HAMP	0.515(0)	0.517(-17)	0.013(+6)	0.513(0)	0.514(-14)	0.013(+7)	0.516(0)	0.515(-8)	0.015(-14)	0.510(-1)	**0.499**(-15)	0.013(-9)
DP-Logits	0.515(0)	0.516(-17)	0.013(+6)	0.513(0)	0.514(-14)	0.012(+6)	0.515(0)	0.515(-8)	0.015(-10)	0.511(0)	0.535(-9)	0.013(-7)
EPD (Ours)	0.507(-2)	0.511(-17)	0.010(-21)	0.495(-4)	0.505(-15)	0.010(-17)	0.500(-3)	0.504(-9)	0.008(-51)	**0.497**(-3)	0.507(-14)	**0.009**(-38)

Table 5: Results evaluating the impact of EPD's judge model capacity towards defending MIAs.

MIA Method	DeepSeek-R1-1.5B			DeepSeek-R1-8B		
	AUC	ASR	TPR@1	AUC	ASR	TPR@1
SPV-MIA	0.52	0.32	0.03	0.48	0.51	0.00
LiRA	0.27	0.59	0.01	0.23	0.50	0.00
Recall	0.63	0.61	0.05	0.61	0.59	0.01
Log-Loss	0.62	0.61	0.03	0.60	0.58	0.02
Zlib	0.65	0.63	0.04	0.65	0.61	0.03
Min-K	0.62	0.59	0.03	0.59	0.57	0.03
Min-K++	0.67	0.63	0.02	0.52	0.55	0.02
Average	0.569	0.569	0.030	**0.526**	**0.559**	**0.0157**

compute the percentage of answers that the judge model directly chooses the target model output (19.9%), base model output (10.4%), and the percentage of mixed output (69.7%). Additionally, we compute the bias score as the absolute difference between the percentage of instances in which the judge selected the target answer versus the base answer (9.5%). As evident by the results, the judge predominantly produces mixed or hybrid responses, which aligns with the intended design goal of blending information across heterogeneous distributions rather than copying either model's style. This empirical distribution further confirms that the judge behaves as a neutral aggregator.

5.4.4 Impact of Adaptive Noise Injection

Building upon recent advances in privacy-aware decoding for MIA defense (Wang et al., 2025), we evaluate adaptive noise injection as an additional defense mechanism in our LLM Judge framework. The key intuition is that strategic perturbations introduced during generation may further obfuscate membership signals without significantly degrading answer quality. We adapt the Privacy-Aware Decoding (PAD) methodology to our ensemble setting by modulating noise strength using similarity and loss differences between the target and base model responses, with full details provided in Appendix F. Table 6 reports the performance impact across six MIA attacks under two noise thresholds. The results show that noise injection has highly attack-dependent behavior: LiRA benefits consistently across both noise levels, Recall and Log-Loss exhibit minor improvements, Zlib shows minimal change, and Min-K/Min-K++ even degrade under certain thresholds. These heterogeneous outcomes highlight that noise-based defenses interact differently with score distributions across attacks and are therefore not universally effective.

Overall, while the more aggressive threshold ($\tau = 0.05$) sometimes enhances privacy metrics, its impact remains inconsistent and occasionally counterproductive. These findings indicate that adaptive noise injection offers only modest additional protection on top of the ensemble-based judge design, which already suppresses membership-specific signals effectively. In practice, this suggests that the LLM Judge's inherent aggregation behavior serves as the primary driver of privacy improvements, with noise playing only a secondary and context-dependent role. As such, noise perturbation may be best viewed as an optional enhancement rather than a core component of the defense pipeline.

6 Related Work

Defending against Membership Inference Attacks (MIAs) in Large Language Models (LLMs) has emerged as a critical research frontier. While traditional defenses such as Differential Privacy (DP) (Abadi et al., 2016) offer provable privacy guarantees, applying them directly to LLMs often incurs a severe "utility-privacy trade-off," result-

Table 6: Impact of Adaptive Noise Injection on LLM Judge MIA Defense Performance (Transposed, Two-Decimal Precision)

Methods	LLM Judge (Baseline)			+ Noise ($\tau = 0.1$)			+ Noise ($\tau = 0.05$)		
	AUC	ASR	TPR@1	AUC	ASR	TPR@1	AUC	ASR	TPR@1
LiRA	0.23	0.50	0.00	0.22	0.50	0.01	0.22	0.50	0.00
Recall	0.61	0.60	0.03	0.63	0.61	0.03	0.62	0.60	0.02
Log-Loss	0.61	0.61	0.03	0.63	0.61	0.03	0.62	0.61	0.02
Zlib	0.65	0.61	0.03	0.66	0.66	0.02	0.65	0.61	0.03
Min-K	0.60	0.59	0.03	0.65	0.65	0.01	0.63	0.61	0.05
Min-K++	0.52	0.55	0.01	0.54	0.54	0.06	0.55	0.59	0.02

ing in significant degradation of text generation quality and reasoning capabilities. Consequently, recent literature has bifurcated into training-time and inference-time strategies.

Training-time Approaches. Training-time approaches fundamentally alter the model's learning process. Techniques ranging from strict DP-SGD to data sanitization and deduplication aim to minimize memorization at the source. Recent specific methods, such as those by Tran et al. (Tran et al., 2025, 2024), modify the training objective or dataset composition to enhance resistance against extraction attacks. However, these methods suffer from substantial practical limitations: they require access to the full training corpus and necessitate computationally prohibitive retraining. For modern LLMs with billions of parameters, retraining for privacy is often infeasible, rendering these defenses impractical for pre-trained, proprietary, or API-served models.

Inference-time Approaches Inference-time defenses, in contrast, operate during the generation phase, making them model-agnostic and deployment-friendly. These methods generally aim to mask the confidence signals that attackers exploit. For instance, DP-Logits (Rahimian et al., 2020) introduces noise to the output logits, effectively applying DP at the decoding stage to obscure the true probability distribution. Similarly, HAMP (Chen and Pattabiraman, 2023) mitigates membership leakage by enforcing low-confidence predictions via high-entropy soft labels and entropy-based regularization. Despite their deployment advantages, inference-time defenses remain underexplored. Current methods often struggle to balance the masking of membership signals with the preservation of semantic coherence, motivating our investigation into lightweight, effective inference-time mechanisms. (See Appendix G for an extended survey).

7 Conclusion

In this work, we conducted a comprehensive investigation into the privacy vulnerabilities of RAG and finetuned LLMs against membership inference attacks. Our systematic evaluation reveals that RAG models demonstrate significantly stronger resistance to MIAs compared to their finetuned counterparts, though this enhanced privacy robustness comes at the cost of increased inference latency. To address the privacy vulnerabilities in both paradigms, we introduced the *Ensemble Privacy Defense (EPD)* framework, which integrates the outputs of a finetuned LLM, a non-finetuned LLM, and a dedicated judge model. Our comprehensive experiments demonstrate that this ensemble-based approach substantially reduces MIA success rates across diverse datasets and task types.

8 Limitations

While our proposed framework demonstrates significant improvements in privacy protection, several limitations should be acknowledged. First, our evaluation is primarily conducted on RTX4090 GPU machine, so we do not include the baseline retraining model/recreating the dataset, nor model components with greater capacity (e.g., DeepSeek-R1-33B). Second, the computational overhead of the ensemble approach, including the judge model inference, may limit its deployment in resource-constrained environments. Third, our noise injection experiments show mixed results, indicating that the effectiveness of additional privacy mechanisms may be context-dependent and require further optimization. Finally, the framework's performance is evaluated against existing MIA attack baselines, but future attacks may exploit different vulnerabilities that our current defense mechanisms do not address.

9 Ethical Considerations

While our research aims to enhance privacy protection, we acknowledge ethical considerations. Our evaluation is limited to specific datasets and model architectures, and the effectiveness of our methods may vary in different contexts. We encourage the research community to consider the broader implications of privacy attacks and defenses in real-world applications, and we commit to responsible disclosure of any significant vulnerabilities discovered during our research.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Maya Anderson, Guy Amit, and Abigail Goldsteen. 2025. Is my data in your retrieval database? membership inference attacks against retrieval augmented generation. In *Proceedings of the 11th International Conference on Information Systems Security and Privacy*, page 474–485. SCITEPRESS - Science and Technology Publications.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pages 1897–1914. IEEE.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, and 1 others. 2021. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650.

Zitao Chen and Karthik Pattabiraman. 2023. Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction. *arXiv preprint arXiv:2307.01610*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. 2023. Practical membership inference attacks against fine-tuned large language models via self-prompt calibration. *arXiv preprint arXiv:2311.06062*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022a. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

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

Zhiheng Huang, Yannan Liu, Daojing He, and Yu Li. 2025. Df-mia: A distribution-free membership inference attack on fine-tuned large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(1):32012.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Preprint*, arXiv:1705.03551.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Kiseung Kim and Jay-Yoon Lee. 2024. Re-rag: Improving open-domain qa performance and interpretability with relevance estimator in retrieval-augmented generation. *Preprint*, arXiv:2406.05794.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Mingrui Liu, Sixiao Zhang, and Cheng Long. 2025. Mask-based membership inference attacks for retrieval-augmented generation. *Preprint*, arXiv:2410.20142.

Bo Ni, Zheyuan Liu, Leyao Wang, Yongjia Lei, Yuying Zhao, Xueqi Cheng, Qingkai Zeng, Luna Dong, Yinglong Xia, Krishnaram Kenthapadi, Ryan Rossi, Franck Dernoncourt, Md Mehrab Tanjim, Nesreen Ahmed, Xiaorui Liu, Wenqi Fan, Erik Blasch, Yu Wang, Meng Jiang, and Tyler Derr. 2025. Towards trustworthy retrieval augmented generation for large language models: A survey. *Preprint*, arXiv:2502.06872.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2023. Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934*.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2024. Fine-tuning or retrieval? comparing knowledge injection in llms. *Preprint*, arXiv:2312.05934.

Shadi Rahimian, Tribhuvanesh Orekondy, and Mario Fritz. 2020. Sampling attacks: Amplification of membership inference attacks by repeated queries. *arXiv preprint arXiv:2009.00395*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *Preprint*, arXiv:1606.05250.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Bao Q Tran, Viet Nguyen, Anh Tran, and Toan Tran. 2024. Dual-model defense: Safeguarding diffusion models from membership inference attacks through disjoint data splitting. *arXiv preprint arXiv:2410.16657*.

Toan Tran, Ruixuan Liu, and Li Xiong. 2025. Tokens for learning, tokens for unlearning: Mitigating membership inference attacks in large language models via dual-purpose training. *arXiv preprint arXiv:2502.19726*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Haoran Wang, Xiongxiao Xu, Baixiang Huang, and Kai Shu. 2025. Privacy-aware decoding: Mitigating privacy leakage of large language models in retrieval-augmented generation. *arXiv preprint arXiv:2508.03098*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE.

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

Kaiyuan Zhang, Siyuan Cheng, Hanxi Guo, Yuetian Chen, Zian Su, Shengwei An, Yuntao Du, Charles Fleming, Ashish Kundu, Xiangyu Zhang, and Ninghui Li. 2025. Soft: Selective data obfuscation for protecting llm fine-tuning against membership inference attacks. *Preprint*, arXiv:2506.10424.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*. Includes the AG News dataset.

A RAG vs Finetune

Detailed results found in Table 7.

B Prompt for LLM Judge

Judge prompt

```
You are a judge that needs to generate
a final answer based on two candidate
answers.
Consider both PRIVACY (low-loss answers
that might leak training data) and
ACCURACY (being factually correct).

Question: q
```

Table 7: Comparison of Robustness Evaluation Results on Trivia-QA and SQuAD (RAG vs. SFT)

Dataset	Paradigm	SPV-MIA		LiRA		Recall		Log-Loss		Zlib		Min-K		Min-K++	
		AUC	ASR	AUC	ASR	AUC	ASR	AUC	ASR	AUC	ASR	AUC	ASR	AUC	ASR
Trivia-QA	SFT	0.5454	0.7210	0.4521	**0.5078**	0.7526	0.6969	0.7628	0.7073	0.7187	0.6736	0.6918	0.6554	0.7804	0.7306
	RAG	**0.4906**	**0.5090**	**0.3768**	0.5126	**0.3856**	**0.6026**	**0.5991**	**0.6618**	**0.6863**	**0.6192**	**0.6900**	**0.5951**	**0.7800**	**0.5570**
SQuAD	SFT	**0.4951**	**0.3100**	0.8380	0.8005	0.4964	0.5181	0.6252	0.5984	0.4999	0.5285	**0.4533**	0.5389	**0.5123**	**0.5259**
	RAG	0.5025	0.3500	**0.6149**	**0.6088**	**0.4102**	**0.5130**	**0.4546**	**0.5233**	**0.4523**	**0.5233**	0.4779	**0.5285**	0.5145	0.5440
Average	SFT	0.5203	0.5155	0.6450	0.6542	0.6245	0.6075	0.6940	0.6529	0.6093	0.6011	**0.5726**	0.5972	**0.6464**	0.6283
	RAG	**0.4966**	**0.4295**	**0.4959**	**0.5607**	**0.3979**	**0.5578**	**0.5269**	**0.5926**	**0.5693**	**0.5713**	0.5840	**0.5618**	0.6473	**0.5505**

Answer A (target): a_{target}
Answer B (base): a_{base}

Model confidence (lower loss indicates higher confidence):
- Target answer loss: $\mathcal{L}_{\text{target}}$
- Base answer loss: $\mathcal{L}_{\text{base}}$

Your task:
1. Weigh privacy and factual accuracy.
2. Generate a new final answer that:
- Maintains accuracy while protecting privacy
- Uses general language instead of specific details
- Avoids exact quotes or precise measurements
- Combines the best aspects of both answers

Generate your final answer:

C Dataset Details

Table 8: Dataset Statistics and Characteristics

Dataset	Train Size	Test Size	Task Type	Domain
TriviaQA	87,622	11,313	QA (Generative)	Factual Knowledge
SQuAD	87,599	10,570	QA (Extractive)	Wikipedia Articles
AG News	120,000	7,600	Classification	News Articles

We evaluate our proposed framework on three diverse datasets to ensure comprehensive assessment across different domains and task types:

- **TriviaQA (Unfiltered)**: A large-scale reading comprehension dataset containing 95K question-answer pairs derived from Wikipedia articles. The dataset features complex factual questions that require reasoning over multiple documents, making it suitable for evaluating both knowledge retrieval and generation capabilities.

- **SQuAD**: The Stanford Question Answering Dataset, comprising 100K+ question-answer pairs based on Wikipedia articles. SQuAD focuses on extractive question answering, where answers are spans of text from the given context, providing a different challenge compared to generative tasks.

- **AG News**: A news classification dataset containing 120K news articles from four categories (World, Sports, Business, Sci/Tech). This dataset represents a different task type (classification) and domain (news), allowing us to assess the generalizability of our privacy defense mechanisms.

D Models and Implementation Details

D.1 Supervised Finetuning Setup

We employ Llama-2-7B as the base model for supervised finetuning. To efficiently adapt the large model to our target datasets, we utilize Parameter-Efficient Fine-Tuning (PEFT) techniques, specifically LoRA (Low-Rank Adaptation), which significantly reduces the number of trainable parameters while maintaining performance. The training process incorporates early stopping to prevent overfitting and ensure optimal generalization. Key hyperparameters include a learning rate of 1×10^{-4}, LoRA rank of 8, LoRA alpha of 16, batch size of 4, and a block size of 128. Early stopping is applied based on validation loss to avoid overfitting.

D.2 Retrieval-Augmented Generation Setup

For the retrieval-augmented generation (RAG) setting, we use the original (non-finetuned) Llama-2-7B as the core LLM. For dense retrieval, we employ the SentenceTransformer (Reimers and Gurevych, 2019) "all-MiniLM-L6-v2" model as the sentence encoder to embed both the input queries and all context passages from the TriviaQA and SQuAD datasets. All context embeddings are precomputed and stored. At inference time, we compute the cosine similarity between the query embedding and all stored context embeddings, retrieving the top-k most similar contexts (with $k = 5$) to construct the final prompt for the LLM. This setup enables efficient and effective retrieval-augmented generation

for open-domain question answering.

D.3 LLM Judge Configuration

For our ensemble defense framework, we configure the components as follows:

- **Target Model**: The target model is the finetuned Llama-2-7B model (as described above) or original Llama-2-7B model embedded RAG model.
- **Base Model**: The base model is an original Llama-2-7B model without any finetuning.
- **Judge Model**: The judge model is DeepSeek-R1-Distill-Qwen-8B (i.e., a distilled model providing efficient and reliable judgment capabilities).

E Model Capacity Analysis

The superior performance of the larger judge model can be attributed to several factors. First, increased parameter capacity enables more sophisticated understanding of complex prompts that require reasoning about privacy implications (Wei et al., 2022; Kojima et al., 2022). The 8B model demonstrates enhanced ability to identify potentially privacy-leaking information in candidate answers and generate appropriate alternatives that maintain utility while preserving privacy.

Second, larger models exhibit improved instruction-following capabilities, which is crucial for the judge model's task of synthesizing answers from multiple sources while adhering to privacy constraints (Ouyang et al., 2022; Wei et al., 2021). The enhanced capacity allows for more nuanced evaluation of the trade-offs between accuracy and privacy, leading to better-informed decisions about answer selection and modification.

Third, the increased model capacity facilitates better handling of edge cases and complex scenarios where simple heuristics may fail. This is particularly important for MIA defense, as attackers often exploit subtle patterns in model outputs that require sophisticated reasoning to detect and mitigate.

These findings have important implications for the practical deployment of our LLM Judge framework. While the 1.5B model provides a computationally efficient baseline, the 8B model offers significantly enhanced privacy protection at the cost of increased computational requirements. The choice between these models should be guided by the specific privacy requirements and computational constraints of the target application.

F Adaptive Noise Injection Strategy

Answer Similarity We measure the semantic similarity between target and base model responses using cosine similarity of their embeddings:

$$\text{sim}(a_{\text{target}}, a_{\text{base}}) = \frac{\mathbf{e}_{a_{\text{target}}} \cdot \mathbf{e}_{a_{\text{base}}}}{||\mathbf{e}_{a_{\text{target}}}|| \cdot ||\mathbf{e}_{a_{\text{base}}}||}$$

where $\mathbf{e}_{a_{\text{target}}}$ and $\mathbf{e}_{a_{\text{base}}}$ are the embeddings of the target and base model answers, respectively.

Loss Difference We compute the normalized difference between the target and base model losses:

$$\Delta_{\text{loss}} = \frac{|\mathcal{L}_{\text{target}} - \mathcal{L}_{\text{base}}|}{\max(\mathcal{L}_{\text{target}}, \mathcal{L}_{\text{base}})}$$

where $\mathcal{L}_{\text{target}}$ and $\mathcal{L}_{\text{base}}$ are the cross-entropy losses of the target and base models, respectively.

The noise injection strength is determined by a weighted combination of these metrics:

$$\beta = w_{\text{sim}} \cdot (1 - \text{sim}(a_{\text{target}}, a_{\text{base}})) + w_{\text{loss}} \cdot \Delta_{\text{loss}}$$

where w_{sim} and w_{loss} are weighting parameters (set to 0.6 and 0.4, respectively, in our experiments).

Threshold-based Noise Activation: Noise injection is activated when the computed strength exceeds predefined thresholds. We evaluate two threshold configurations: $\tau_1 = 0.1$ and $\tau_2 = 0.05$, representing conservative and aggressive noise injection strategies, respectively. When $\beta > \tau$, the system applies noise injection to the final answer generation process.

Token-level Noise Implementation: Following the PAD methodology (Wang et al., 2025), we implement token-level noise injection for tokens that appear in both the target answer and the final judge-generated answer. For each token t at position i, we inject calibrated Gaussian noise into the logits:

$$\tilde{\mathbf{z}}_i = \mathbf{z}_i + \mathcal{N}(0, \sigma_i^2 \mathbf{I})$$

where $\mathbf{z}_i$ represents the original logits for token i, and σ_i is the adaptive noise scale determined by:

$$\sigma_i = \sigma_{\text{base}} \cdot \lambda_{\text{amp}} \cdot \beta \cdot \exp(-\alpha \cdot \text{confidence}_i)$$

where σ_{base} is the base noise scale, λ_{amp} is the amplification factor, α is the confidence decay parameter, and confidence_i is the model's confidence for token i measured by the logit margin between the top two predictions.

To provide formal privacy guarantees, we employ a Rényi Differential Privacy (RDP) accountant that tracks the cumulative privacy loss across all noise-injected tokens. For each noise injection step, the RDP cost is computed as:

$$\text{RDP}_\alpha(\mathcal{M}) = \frac{\alpha\sigma_i^2}{2}$$

where $\alpha > 1$ is the RDP order parameter. The total privacy cost is accumulated across all protected tokens, providing an explicit (ε, δ)-differential privacy guarantee for the ensemble response.

G Extended Related Work

MIA Defense Mechanism Defending against MIAs in LLMs is an active area of research. Traditional defenses such as differential privacy (DP) (Abadi et al., 2016) provide provable privacy guarantees but often at the cost of significant utility degradation. Practical defenses aim to empirically reduce membership leakage while maintaining model performance.

Recent work on LLM MIA defense can be broadly categorized into training-time and inference-time approaches. Training-time methods, such as those proposed by Tran et al. (Tran et al., 2025, 2024), modify the training process and dataset to enhance privacy protection. However, these approaches require retraining pretrained models, which becomes computationally prohibitive and impractical for large-scale LLMs given their massive parameter counts and training costs. In contrast, inference-time defenses operate without modifying model parameters or requiring retraining, making them more suitable for real-world deployment. HAMP (Chen and Pattabiraman, 2023) enforces less confident predictions by introducing high-entropy soft labels and an entropy-based regularizer, while DP-Logits applies differential privacy to the output logits during inference. Rahimian et al. (Rahimian et al., 2020) propose sampling-based defenses that modify outputs at inference time to reduce membership leakage. These inference-time approaches provide a more practical solution for privacy protection in large-scale LLM deployments.

Despite the advantages of inference-time defenses, they remain relatively underexplored compared to training-time approaches. This gap motivates our investigation into more effective inference-time defense mechanisms that can provide robust privacy protection without the computational overhead of model retraining.

H Algorithm Pseudo Code

Algorithm 1 Judge Decision Categorization and Bias Score Calculation

1.5em 0.8em

Require: Judge answer j, target answer t, base answer b
Ensure: Category $c \in \{\text{target}, \text{base}, \text{mixed}, \text{unknown}\}$, bias score s

Normalize strings: $j' \leftarrow \text{normalize}(j)$, $t' \leftarrow \text{normalize}(t)$, $b' \leftarrow \text{normalize}(b)$
Initialize counters: $n_{\text{target}} \leftarrow 0$, $n_{\text{base}} \leftarrow 0$, $n_{\text{total}} \leftarrow 0$

for each judge decision **do**
    $n_{\text{total}} \leftarrow n_{\text{total}} + 1$

    if $j' = t'$ and $j' = b'$ **then**
        $c \leftarrow$ mixed (both)
        continue
    else if $j' = t'$ **then**
        $c \leftarrow$ target (exact match)
        $n_{\text{target}} \leftarrow n_{\text{target}} + 1$
    else if $j' = b'$ **then**
        $c \leftarrow$ base (exact match)
        $n_{\text{base}} \leftarrow n_{\text{base}} + 1$
    else
        Compute F1 scores: $f1_{\text{target}} \leftarrow \text{F1}(j', t')$, $f1_{\text{base}} \leftarrow \text{F1}(j', b')$
        if $f1_{\text{target}} < 0.3$ and $f1_{\text{base}} < 0.3$ **then**
            Compute prefix similarity: $p_{\text{target}} \leftarrow \text{prefix\_sim}(j'[:50], t'[:50])$
            $p_{\text{base}} \leftarrow \text{prefix\_sim}(j'[:50], b'[:50])$
            if $p_{\text{target}} > p_{\text{base}}$ **then**
                $c \leftarrow$ target (prefix similarity)
                $n_{\text{target}} \leftarrow n_{\text{target}} + 1$
            else if $p_{\text{base}} > p_{\text{target}}$ **then**
                $c \leftarrow$ base (prefix similarity)
                $n_{\text{base}} \leftarrow n_{\text{base}} + 1$
            else
                $c \leftarrow$ mixed(Similarity tie)
            end if
        else
            if $f1_{\text{target}} > 1.15 \times f1_{\text{base}}$ **then**
                $c \leftarrow$ target (F1 similarity)
                $n_{\text{target}} \leftarrow n_{\text{target}} + 1$
            else if $f1_{\text{base}} > 1.15 \times f1_{\text{target}}$ **then**
                $c \leftarrow$ base (F1 similarity)
                $n_{\text{base}} \leftarrow n_{\text{base}} + 1$
            else
                $c \leftarrow$ mixed (F1 tie)
            end if
        end if
    end if
end for
Compute percentages: $p_{\text{target}} \leftarrow n_{\text{target}}/n_{\text{total}}$, $p_{\text{base}} \leftarrow n_{\text{base}}/n_{\text{total}}$
Calculate bias score: $s \leftarrow |p_{\text{target}} - p_{\text{base}}|$
return s