Powerful Training-Free Membership Inference Against Autoregressive Language Models

Anonymous ACL submission

Abstract

Fine-tuned language models pose significant privacy risks, as they may memorize and expose sensitive information from their training data. Membership inference attacks (MIAs) provide a principled framework for auditing these risks, yet existing methods achieve limited detection rates, particularly at the low false-positive thresholds required for practical privacy auditing. We present EZ-MIA, a membership inference attack that exploits a key observation: memorization manifests most strongly at error positions, specifically tokens where the model predicts incorrectly yet still shows elevated probability for training examples. We introduce the Error Zone (EZ) score, which measures the directional imbalance of probability shifts at error positions relative to a pretrained reference model. This principled statistic requires only two forward passes per query and no model training of any kind. On WikiText with GPT-2, EZ-MIA achieves 3.8× higher detection than the previous state-of-the-art under identical conditions (66.3% versus 17.5% true positive rate at 1% false positive rate), with near-perfect discrimination (AUC 0.98). At the stringent 0.1% FPR threshold critical for real-world auditing, we achieve 8× higher detection than prior work (14.0% versus 1.8%), requiring no reference model training. These gains extend to larger architectures: on AG News with Llama-2-7B, we achieve 3× higher detection (46.7% versus 15.8% TPR at 1% FPR). These results establish that privacy risks of fine-tuned language models are substantially greater than previously understood, with implications for both privacy auditing and deployment decisions. Code is provided in the supplementary material and will be released publicly upon acceptance.

1 Introduction

The practice of fine-tuning large language models (LLMs) on private datasets has unlocked immense capabilities but also introduces significant privacy risks, as models may memorize and expose sensitive training data (Carlini et al., 2021). Membership inference attacks (MIAs), which aim to determine if a specific record was in a model's training set, are the standard tool for auditing these risks (Shokri et al., 2017). However, existing MIAs are fundamentally limited. Reference-free attacks, which threshold a model's loss or perplexity, suffer from high false-positive rates because they fail to distinguish true memorization from inherently "easy" samples (Yeom et al., 2018). Reference-based attacks like the Likelihood Ratio Attack (LiRA) mitigate this by calibrating scores against a reference model, but they require unrealistic access to data from the target's training distribution or are computationally prohibitive (Carlini et al., 2022). Critically, all prior methods reduce a sequence's rich, token-level predictions to a single scalar score, discarding valuable structural information.

Our central insight is that memorization manifests most strongly at *error positions*, specifically tokens where the model fails to predict correctly. At positions where the model succeeds, both the fine-tuned target and pretrained reference typically assign high probability to the correct token, revealing little about membership. But at error positions, a distinctive pattern emerges: for training members, fine-tuning elevates the correct token's probability even when it remains below competing predictions. This residual signal is the signature of memorization that aggregate statistics miss.

We operationalize this insight with EZ-MIA, a membership inference attack of high simplicity and effectiveness. Rather than extracting multiple features or training classifiers, EZ-MIA computes a single statistic: the ratio of upward to downward probability movement at error positions, relative to a pretrained reference model. This Error Zone (EZ) score measures the directional imbalance that memorization induces, a scale-invariant quantity with principled theoretical grounding. The attack

requires only two forward passes per query and no model training of any kind: no shadow models, no reference model fine-tuning.

Despite its simplicity, EZ-MIA substantially outperforms prior work across the large majority of dataset and model configurations we evaluate. On WikiText with GPT-2, we achieve 3.8× higher detection than SPV-MIA (Fu et al., 2024) under identical experimental conditions (66.3% versus 17.5% true positive rate at 1% false positive rate), using only the pretrained base model as reference. At the stringent 0.1% FPR threshold critical for real-world auditing, we achieve 8× higher detection than prior work (14.0% versus 1.8%), requiring no reference model training. These gains extend to larger architectures: on AG News with Llama-2-7B, we achieve 3× higher detection (46.7% versus 15.8% TPR at 1% FPR). We further demonstrate that fine-tuning methodology is a primary determinant of privacy risk: the same model (GPT-2) on the same data (XSum) yields 82.6% detection under full fine-tuning but only 1.5% under LoRA, a 55× reduction.

These results demonstrate that the privacy risks of fine-tuned language models are substantially greater than previously understood. For privacy auditing, they establish that evaluations using weaker attacks may dramatically underestimate true leakage. For practitioners, they reveal that fine-tuning methodology, not just model scale or training duration, fundamentally shapes privacy risk. EZ-MIA provides a new, more accurate baseline against which both privacy evaluations and defenses must be measured.

2 Background and Related Work

Membership inference attacks (MIAs) determine whether a specific data record was used to train a target model, serving both as a direct privacy threat and as a foundation for more sophisticated attacks such as training data extraction (Carlini et al., 2021). We review the evolution of these attacks and their application to language models.

2.1 Membership Inference Foundations

The study of membership inference began with the shadow model paradigm (Shokri et al., 2017). This approach trains multiple shadow models to mimic the target's behavior, using their outputs on known members and non-members to train a binary attack classifier. This established the core insight that models behave differently on training versus unseen data. However, shadow model attacks require substantial computational resources and assume the adversary has access to data from a distribution similar to the target's training set.

A simpler approach, the LOSS attack (Yeom et al., 2018), observes that training samples typically incur lower loss than non-members, enabling a threshold-based attack. While computationally efficient, this method suffers from high false-positive rates because sample difficulty varies independently of membership: some samples are inherently easy for any model, while others are hard.

Carlini et al. (2022) formalized membership inference as hypothesis testing and developed the Likelihood Ratio Attack (LiRA), which remains the gold standard for high-precision inference. LiRA calibrates a sample's score using shadow models to factor out its inherent difficulty. Crucially, this work established that attacks should be evaluated by true-positive rate at low false-positive rates (e.g., TPR@0.1%FPR), a metric critical for practical privacy auditing where high confidence is required. While powerful, LiRA's reliance on training hundreds of shadow models per sample limits its scalability.

2.2 Attacks and Reference Models

Modern MIAs can be categorized by their assumptions. *Reference-free attacks* operate solely on the target model's outputs, avoiding the need for auxiliary data. The Neighborhood Attack (Mattern et al., 2023) generates synthetic variants of a query sample and infers membership by comparing the original's likelihood to the average likelihood of its neighbors, at a high computational cost (~101 forward passes). More efficient methods like MIN-K% PROB (Shi et al., 2024) focus on the tokens assigned the lowest probability, intuiting that non-members are more likely to contain such outlier tokens.

Reference-based attacks achieve higher precision by calibrating the target's scores against a reference model. The key insight is that a sample with high likelihood under both target and reference models is simply "easy," whereas high likelihood only under the target suggests memorization. The primary challenge for these attacks is the strong assumption that an adversary can obtain data from the target's training distribution to train a suitable reference model.

To address this, Fu et al. (2024) proposed SPV-

MIA, which constructs a reference dataset via self-prompting, prompting the target model itself and fine-tuning a reference model on its generations. While this reduces data assumptions, its membership signal relies on aggregate probabilistic stability across multiple paraphrased inputs, requiring approximately 42 forward passes per sample. Our work departs from prior methods by identifying *where* the membership signal concentrates rather than aggregating across all positions. We show that error positions (where the model predicts incorrectly) carry substantially stronger signal than correct predictions, and exploit this with a principled statistic requiring only two forward passes and no reference model training.

2.3 Membership Inference Against LLMs

Applying MIAs to language models presents unique challenges. In foundational work, Carlini et al. (2021) demonstrated that LLMs such as GPT-2 memorize and can regurgitate verbatim training data, establishing memorization as a concrete privacy risk.

However, recent work has shown that MIAs against large pretrained models on web-scale data are often ineffective (Duan et al., 2024). The combination of massive training sets and limited training iterations prevents the strong per-sample memorization that MIAs typically detect. Apparent success in this setting often stems from distribution shifts between member and non-member evaluation sets rather than a genuine membership signal.

The picture is substantially different for fine-tuned models. Fine-tuning uses far smaller datasets, often for multiple epochs, creating a much stronger memorization signal. Studies have consistently shown that fine-tuned models are significantly more vulnerable to membership inference (Zhang et al., 2025b; Fu et al., 2024), though recent work suggests parameter-efficient methods like LoRA may reduce memorization (Wang and Li, 2025). Our work focuses on this high-risk fine-tuning setting and provides the first quantification of the privacy gap between full fine-tuning and LoRA under a high-precision membership inference attack. Concurrent work has also recognized the insufficiency of aggregate loss for sequence models (Rossi et al., 2025), but their approach adapts LiRA within the costly shadow model paradigm. In contrast, our work identifies that memorization signal concentrates at error positions and introduces a principled, scale-invariant statistic to measure it, achieving dramatic improvements without any shadow model training.

3 Methodology

We present EZ-MIA, a membership inference attack that exploits a key observation: memorization manifests most clearly at positions where the model fails to predict correctly. Our method achieves strong discrimination with only two forward passes and no model training.

3.1 Threat Model

We consider a practical threat model where the adversary has:

1. **Query access:** Access to the target model θ, enabling computation of token-level log-probabilities for any input sequence.

2. **Reference model access:** Access to the pre-trained base model from which the target was fine-tuned, or a public model of comparable architecture.

This threat model aligns with realistic attack scenarios against fine-tuned open-weight models and language model APIs that expose token probabilities. Notably, we require no access to samples from the target model's training distribution and no auxiliary model training.

3.2 Memorization at Error Positions

Prior membership inference methods aggregate statistics across all token positions, but this dilutes the membership signal. Our key observation is that *memorization manifests most strongly at positions where the model predicts incorrectly*.

The intuition is straightforward. At positions where the target model's top prediction matches the ground truth, both the fine-tuned target and the pretrained reference typically assign high probability to the correct token; success reveals little about membership. However, at error positions where the model fails to predict correctly, a different pattern emerges: for training members, fine-tuning still elevates the correct token's probability even when it remains below competing tokens. This residual signal (probability mass shifted upward despite prediction failure) is the signature of memorization.

We validate this empirically: restricting our statistic to error positions yields an AUC score 0.1 greater compared to using only correct-prediction positions (see Appendix E).

3.3 Notation

Let $\boldsymbol{x} = (x_1, \ldots, x_T)$ be a sequence of tokens. The log-probability assigned by model θ to token x_t given its prefix is $\ell_\theta^{(t)} = \log p_\theta(x_t \mid \boldsymbol{x}_{<t})$. Given target model θ and reference model $\hat{\theta}$, we define the token-level log-probability difference:

$$\delta^{(t)} = \ell_\theta^{(t)} - \ell_{\hat{\theta}}^{(t)} \tag{1}$$

We define the *error set* $\mathcal{E}$ as positions where the target model's top prediction is incorrect:

$$\mathcal{E} = \{t \mid \arg\max_v p_\theta(v \mid \boldsymbol{x}_{<t}) \neq x_t\} \tag{2}$$

3.4 The Error Zone Score

Fine-tuning induces probability changes at each error position. We decompose these changes by direction:

$$P = \sum_{t \in \mathcal{E}} [\delta^{(t)}]_+ \qquad N = \sum_{t \in \mathcal{E}} |\,[\delta^{(t)}]_-| \tag{3}$$

where $[x]_+ = \max(x, 0)$ and $[x]_- = \min(x, 0)$. Here P represents total probability mass moved *upward* by fine-tuning (relative to the reference), while N represents mass moved *downward*.

The Error Zone score measures the balance of this movement:

$$\mathrm{EZ}(\boldsymbol{x}) = \frac{P}{N} \tag{4}$$

This answers a simple question: *of all probability adjustments at error positions, how much more moved up than down?* Memorization creates upward pressure on token probabilities through gradient updates; EZ measures this directional imbalance.

A key property of EZ is **scale invariance**: multiplying all $\delta^{(t)}$ by a constant $c > 0$ leaves EZ unchanged, since both P and N scale equally. This allows meaningful comparison across sequences with different intrinsic variability; a sequence with volatile predictions and large $|\delta^{(t)}|$ values is compared on equal footing with a predictable sequence showing smaller movements. We provide a principled derivation of EZ, including formal properties and theoretical grounding, in Appendix D.

3.5 Reference Model

Our method requires a reference model $\hat{\theta}$ for computing the probability differences $\delta^{(t)}$. We use the pretrained base model checkpoint from before fine-tuning. This choice requires no additional training and provides a natural baseline: it captures general language modeling capabilities without any exposure to the target's training data.

The pretrained reference is both principled and practical. Probability differences $\delta^{(t)}$ directly measure what fine-tuning changed, isolating the effect of training on the target dataset. We evaluate alternative reference constructions in Appendix E.

3.6 Attack Procedure

Given a query sequence $\boldsymbol{x}$:

1. Compute token-level log-probabilities under the target model θ and reference model $\hat{\theta}$.
2. Identify error positions $\mathcal{E}$ where the target's top prediction differs from the ground truth.
3. Compute the Error Zone score $\mathrm{EZ}(\boldsymbol{x}) = P/N$.
4. Classify as member if $\mathrm{EZ}(\boldsymbol{x})$ exceeds a threshold τ.

The threshold τ is chosen to achieve a desired false positive rate. The entire attack requires only two forward passes per query (one through the target model and one through the reference) with no shadow model training, no classifier fitting, and no reference model fine-tuning. This represents an order of magnitude reduction in inference-time computational cost compared to methods like SPV-MIA (~42 forward passes) and the Neighborhood Attack (~101 forward passes), while reducing training-time computational cost to zero.

4 Experimental Setup

We evaluate our method across diverse datasets and model architectures to demonstrate its generalizability. This section describes the datasets, target models, baseline methods, and evaluation metrics used in our experiments.

Datasets and Models. We evaluate on three primary datasets spanning different domains: *AG News* (Zhang et al., 2015) (short, topical news), *WikiText-103* (Merity et al., 2016) (diverse encyclopedic text), and *XSum* (Narayan et al., 2018) (formal journalistic prose). For each, we create disjoint 10k member/10k non-member sets for evaluation, with a separate 500-sequence validation set; we select the checkpoint with lowest validation loss to avoid overfitting artifacts. All sequences

are exactly 128 tokens, constructed by concatenating consecutive texts. We evaluate across three model scales: *GPT-2* (Radford et al., 2019) (124M), *GPT-J* (Wang and Komatsuzaki, 2021) (6B), and *Llama-2* (Touvron et al., 2023) (7B). GPT-2 is fully fine-tuned for 3 epochs, while larger models use LoRA (Hu et al., 2022) (rank 16, alpha 32) for computational efficiency. To evaluate domain generalization, we additionally test on *Swallow-Code* (Fujii and Yokota, 2024) (Python code) with *Stable-Code-3B* (Pinnaparaju et al., 2023). Full training and data processing details are in Appendix A.

Reference Model Construction. Our attack requires a reference model to compute the membership score. We use the pretrained model checkpoint before any fine-tuning, requiring no additional computation. This provides a natural baseline capturing general language modeling capabilities without exposure to the target's training data.

Baselines. We compare against representative methods spanning two attack paradigms:

Reference-free attacks: LOSS attack (Yeom et al., 2018), which thresholds on model loss; Zlib (Carlini et al., 2021), which normalizes perplexity by zlib compression length; and Min-K%++ (Zhang et al., 2025a), which aggregates z-score normalized log-probabilities at low-probability tokens.

Reference-based attacks: Reference Loss (RefL), which computes the likelihood ratio between the target model and a pretrained reference following the framework of Carlini et al. (2022); and SPV-MIA (SPV) (Fu et al., 2024), the previous state-of-the-art. In tables, we abbreviate Min-K%++ as MK++.

All baselines are our own implementations evaluated under identical experimental conditions: same models, same fine-tuning protocol, same data splits (10k members, 10k non-members), and same 3-epoch training. For SPV-MIA, we use the official released code executed without modification. This controlled comparison ensures that performance differences reflect the methods themselves rather than experimental setup.

Evaluation Metrics. We report three complementary metrics: *AUC* (Area Under the ROC Curve) for overall discrimination, *TPR@1%FPR* (True Positive Rate at 1% False Positive Rate) for practical auditing scenarios, and *TPR@0.1%FPR* for high-precision settings where false positives are costly. Following Carlini et al. (2022), we focus analysis on the low-FPR metrics, as these determine an attack's practical utility for privacy auditing.

Computational Cost. Our method requires no training and only two forward passes per query (one through target, one through reference), compared to approximately 42 for SPV-MIA. This efficiency enables practical large-scale auditing.

Computational Resources. All experiments were conducted on a single NVIDIA H200 GPU. Fine-tuning and evaluation for each model-dataset configuration completes within one hour.

5 Results

Figure 1 and Table 1 present our main experimental results. We compare EZ-MIA against prior baselines spanning reference-free attacks (LOSS, Zlib, Min-K%++) and reference-based attacks (Reference Loss, SPV-MIA). Our evaluation encompasses three text domains (AG News, WikiText-103, XSum) and three model scales (GPT-2 124M, GPT-J 6B, Llama-2 7B).

5.1 Comparison with Prior Methods

EZ-MIA substantially outperforms all prior attacks in the large majority of configurations. Against reference-free methods, we achieve average AUC improvements of +0.25 over LOSS, +0.25 over Zlib, and +0.28 over Min-K%++. The strongest prior attack, SPV-MIA (0.84 average AUC), trails our method by +0.06 AUC on average. Reference Loss, which like our method uses the pretrained model as reference, achieves 0.78 average AUC. These gaps confirm that focusing on error positions captures memorization signals invisible to aggregate statistics.

The performance advantage is most pronounced on fully fine-tuned models. On WikiText with GPT-2, EZ-MIA achieves 0.984 AUC compared to 0.899 for SPV-MIA. At the stringent TPR@0.1%FPR threshold critical for privacy auditing, we achieve 14.0% detection compared to 1.8% for SPV-MIA: an 8$\times$ improvement.

5.2 Main Results

Table 1 presents comprehensive results comparing EZ-MIA against baselines. GPT-2 is fully fine-tuned, while GPT-J and Llama-2 use LoRA (rank 16, alpha 32) for computational efficiency.

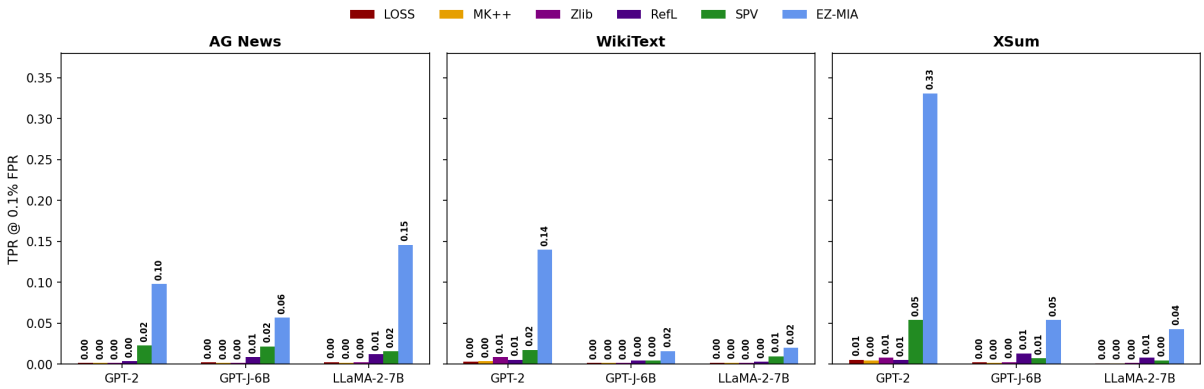

Figure 1: TPR@0.1%FPR comparison across datasets and models. All methods evaluated under identical experimental conditions. GPT-2 uses full fine-tuning; GPT-J and Llama-2 use LoRA. EZ-MIA achieves up to 9× higher detection rates than prior work at this stringent threshold critical for privacy auditing.

Under identical experimental conditions, EZ-MIA achieves the highest AUC in the large majority of configurations.

The results reveal two distinct performance regimes. On fully fine-tuned GPT-2, EZ-MIA achieves strong discrimination: 0.984 AUC on WikiText, 0.950 on AG News, and 0.993 on XSum, with TPR@1%FPR of 66.3%, 39.4%, and 82.6% respectively. At the stringent 0.1% FPR threshold, we detect 14.0%, 9.8%, and 33.1% of members on WikiText, AG News, and XSum respectively (Table 5; visualized in Figure 1).

Performance on LoRA fine-tuned models (GPT-J, Llama-2) remains strong but substantially lower, with average AUC of 0.86 compared to 0.98 for full fine-tuning. This gap reflects the reduced memorization induced by parameter-efficient fine-tuning, which we analyze in the following subsection.

5.3 Effect of Fine-tuning Method

The gap between full fine-tuning and LoRA results in Table 1 conflates fine-tuning method with model architecture. To isolate the effect of fine-tuning method, we evaluate EZ-MIA on GPT-2 under both full fine-tuning (100% of parameters) and LoRA (0.71% of parameters) on XSum.

The difference is striking: full fine-tuning yields 0.993 AUC and 82.6% TPR@1%FPR, while LoRA yields only 0.553 AUC and 1.5% TPR@1%FPR, a reduction of 55× in detection rate. This demonstrates that fine-tuning method is a significant determinant of privacy risk, independent of model scale.

This finding has important implications. First, it explains the performance gap between GPT-2 and larger models in our main results: the gap primarily reflects fine-tuning methodology rather than model capacity. Second, it suggests that parameter-efficient fine-tuning provides substantial (though not complete) protection against membership inference, consistent with recent findings that LoRA reduces memorization compared to full fine-tuning (Wang and Li, 2025). Third, it implies that privacy audits must account for fine-tuning methodology; evaluations on LoRA-tuned models may dramatically underestimate the risks of full fine-tuning.

5.4 Computational Efficiency

EZ-MIA requires only two forward passes per query (one through the target model, one through the reference) and no reference model training. This compares favorably to SPV-MIA, which requires approximately 42 forward passes plus reference model training. The efficiency gain enables practical large-scale privacy auditing with minimal computational overhead.

5.5 Generalization to Code

To evaluate domain generalization beyond natural language, we test EZ-MIA on Swallow-Code with Stable-Code-3B (LoRA fine-tuned). We achieve 0.893 AUC and 38.8% TPR@1%FPR. This strong performance, comparable to LoRA-tuned natural language models, confirms that EZ-MIA captures domain-agnostic memorization signals and generalizes beyond the text domains used for method development.

		AUC						TPR@1%FPR (%)					
Dataset	**Model**	**LOSS**	**Zlib**	**MK++**	**RefL**	**SPV**	**EZ-MIA**	**LOSS**	**Zlib**	**MK++**	**RefL**	**SPV**	**EZ-MIA**
WikiText	GPT-2	.725	.735	.688	.810	.899	**.984**	4.1	5.3	2.7	5.2	17.5	**66.3**
WikiText	GPT-J	.563	.566	.540	.615	.763	**.781**	1.7	1.7	1.4	2.7	4.9	**10.7**
WikiText	Llama-2	.553	.555	.541	.623	**.780**	.771	1.7	1.7	1.3	2.5	5.4	**9.4**
AG News	GPT-2	.742	.740	.709	.773	.879	**.950**	2.0	2.0	2.4	2.0	18.1	**39.4**
AG News	GPT-J	.704	.701	.665	.828	.915	**.955**	2.1	2.1	2.3	3.1	21.6	**42.6**
AG News	Llama-2	.691	.686	.639	.844	.898	**.961**	1.6	1.5	1.5	3.8	15.8	**46.7**
XSum	GPT-2	.742	.743	.682	.961	.927	**.993**	5.5	5.6	4.2	14.7	31.4	**82.6**
XSum	GPT-J	.578	.577	.551	.787	.761	**.860**	1.8	2.0	1.5	6.2	5.6	**19.7**
XSum	Llama-2	.579	.576	.553	.799	.756	**.840**	1.9	1.9	1.5	5.5	5.3	**17.6**

Table 1: Main results across datasets and models. All baselines are our own implementations under identical conditions. GPT-2 uses full fine-tuning; GPT-J and Llama-2 use LoRA. Bold indicates best per row.

Fine-tuning	**AUC**	**TPR@1%**	**TPR@0.1%**
Full (100%)	0.993	82.6%	33.1%
LoRA (0.71%)	0.553	1.5%	0.1%

Table 2: Effect of fine-tuning method on privacy leakage. Same model (GPT-2), same dataset (XSum), same attack (EZ-MIA). Full fine-tuning induces dramatically stronger memorization.

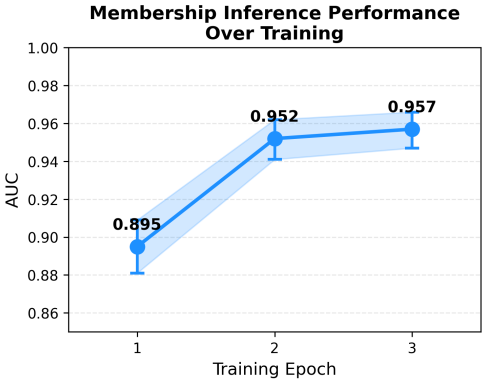

Figure 2: EZ-MIA performance over training epochs on XSum/GPT-2-XL. Privacy leakage emerges early (epoch 1: AUC = 0.895) and increases with continued training. Error bars show 95% bootstrap confidence intervals (1,000 resamples of evaluation sequences).

6 Analysis

Beyond comparing against baselines, we analyze when and why EZ-MIA succeeds or fails.[1] We examine how the membership signal evolves during training and characterize the conditions under which our method is most and least effective.

6.1 Relationship to Training Dynamics

Membership inference exploits memorization, and memorization emerges through training. We investigate how EZ-MIA performance evolves as fine-tuning progresses, evaluating on XSum with GPT-2-XL at checkpoints after each training epoch.

Figure 2 shows that AUC increases monotonically from 0.895 after epoch 1 to 0.957 after epoch 3. Notably, even a single epoch of fine-tuning induces sufficient memorization for strong membership inference (TPR@1%FPR = 54.1% at epoch 1, rising to 90.2% at epoch 3). This indicates that privacy risks emerge early in training.

More striking is the relationship between EZ-MIA performance and overfitting. We compute the train-test loss gap (a standard measure of overfitting) at each checkpoint and observe that AUC increases monotonically with this gap across all three epochs. This pattern is consistent with our theoretical framework: the EZ score measures precisely the probability boost that fine-tuning confers on training examples, which is the token-level manifestation of overfitting. While three checkpoints are insufficient for formal statistical inference, this connection suggests that EZ-MIA could serve not only as a post-hoc auditing tool but also as a real-time privacy monitor during training, a hypothesis that warrants further investigation with more fine-grained training dynamics.

6.2 Robustness and Failure Modes

We examine whether EZ-MIA performance depends on sample difficulty and characterize when the method fails.

Robustness to Sample Difficulty. One concern is that EZ-MIA might only succeed on "easy" samples while failing on difficult ones. We partition sequences into quartiles by reference model perplexity (a proxy for difficulty) and compute AUC

[1] Analysis experiments in this section use XSum with fully fine-tuned GPT-2-XL (1.5B parameters) and 1k members/1k non-members. We use GPT-2-XL here to demonstrate that qualitative findings hold across model scales; the larger model also exhibits stronger memorization, making failure mode analysis more informative.

within each quartile on XSum/GPT-2-XL:

	Q1 (easy)	Q2	Q3	Q4 (hard)
AUC	0.924	0.893	0.878	0.887

Performance is consistent across difficulty levels (range: 0.046), with no systematic degradation on hard samples. EZ-MIA's focus on error positions may explain this robustness: difficult samples have more error positions, providing more signal rather than less.

Failure Mode Characterization. At 10% FPR on XSum/GPT-2-XL, we examine false negatives (members incorrectly classified as non-members) and false positives (non-members incorrectly classified as members):

	Avg Tokens	**Avg Errors**	**Mean δ**
All Members	354	173	$+0.30$
False Negatives	430	230	-0.15
All Non-members	363	189	$+0.02$
False Positives	94	46	$+0.41$

Table 3: Characteristics of failure cases on XSum/GPT-2-XL evaluated on 1k members and 1k nonmembers, with unrestricted sequence length. False negatives are longer sequences with more errors and negative mean δ; false positives are shorter sequences where the reference model performs unusually poorly.

False negatives (missed members) are substantially longer than average (430 vs. 354 tokens) with more error positions (230 vs. 173) and negative mean δ; these are members on which the model actually performs *worse* than the reference, perhaps due to distribution shift within the training set. False positives are notably shorter (94 vs. 363 tokens) with fewer errors (46 vs. 189) and high mean δ; these are non-members where the reference model happens to perform poorly, creating a spurious signal.

These failure modes are predictable from sequence characteristics rather than random. Practitioners should exercise additional caution when auditing sequences of extreme relative length, where EZ-MIA's reliability degrades.

7 Conclusion

We have presented EZ-MIA, a membership inference attack against fine-tuned language models that achieves dramatic improvements over prior work through a simple insight: memorization manifests most strongly at error positions. Rather than aggregating statistics across all tokens, we focus on positions where the model predicts incorrectly, measuring the directional imbalance of probability shifts relative to a pretrained reference. This Error Zone score captures memorization signal that previous methods miss entirely.

The magnitude of our improvements warrants emphasis. At false positive rates relevant for practical privacy auditing, we detect members at rates up to 9$\times$ higher than prior methods. On fully fine-tuned models, we achieve 83% TPR at 1% FPR on XSum with GPT-2. Yet our method requires only two forward passes per query and no model training of any kind, an order of magnitude more efficient than prior reference-based attacks.

Our results carry significant implications. For privacy auditing, they demonstrate that current evaluation practices using weaker attacks may substantially underestimate true privacy risks. For practitioners, they reveal that fine-tuning methodology fundamentally shapes privacy exposure: the same model yields 83% detection under full fine-tuning but only 1.5% under LoRA, a 55$\times$ reduction. For defense design, EZ-MIA establishes a new baseline against which mitigations must be evaluated. Ultimately, this work demonstrates that the privacy risks of fine-tuned models are greater than previously understood, underscoring the urgent need for more rigorous privacy evaluation and methodology-aware deployment decisions.

Limitations

Our method has several important limitations.

First, while EZ-MIA outperforms prior work in the large majority of configurations, it is not universally superior. On WikiText with Llama-2-7B, SPV-MIA achieves marginally higher AUC (0.780 vs 0.771).

Second, our evaluation is scoped to the fine-tuning setting. Applying EZ-MIA to models pretrained on web-scale data remains an open challenge, as the memorization signal is far weaker and more diffuse in that regime.

Third, while we demonstrate that fine-tuning method dramatically affects privacy risk, our LoRA experiments use a single configuration (rank 16, alpha 32). The relationship between LoRA hyperparameters and privacy leakage as measured by EZ-MIA remains unexplored; different rank or alpha values may yield different vulnerability profiles.

Fourth, the EZ score requires identifying error positions, which assumes access to ground truth tokens. This assumption holds for membership inference (where the attacker queries with candidate training sequences) but may limit applicability to other privacy attacks.

Finally, our results comparing full fine-tuning and LoRA cannot disentangle fine-tuning method from computational constraints: we use full fine-tuning for GPT-2 and LoRA for larger models due to resource limitations. While our controlled experiment on GPT-2 (full vs LoRA) isolates the fine-tuning method effect, we cannot rule out interactions between model scale and fine-tuning method without additional experiments.

Ethics Statement

The development of more powerful membership inference attacks presents a clear dual-use concern. The techniques presented in this paper could, in principle, be used by malicious actors to probe deployed language models and infer the presence of sensitive information in their training sets, potentially deanonymizing individuals or revealing confidential data. We acknowledge this risk and have carefully considered the ethical implications of our work.

Our primary motivation is defensive. The field of AI privacy operates on the principle that one cannot defend against a threat that is not well understood. Existing MIA benchmarks, by underestimating the true potential of these attacks, may provide a false sense of security. Our work serves as a more accurate "yardstick" for privacy risk, demonstrating that leakage from fine-tuned models is far more severe than previously established.

The simplicity of EZ-MIA amplifies both its risks and its benefits. Requiring only two forward passes and no model training, it is accessible to a wider range of actors, but this same accessibility enables broader adoption for legitimate privacy auditing. Organizations with limited computational resources can now conduct rigorous privacy evaluations that were previously impractical.

Our finding that fine-tuning methodology dramatically affects privacy risk carries immediate practical value. Practitioners can make informed decisions about the privacy-utility tradeoffs of full fine-tuning versus parameter-efficient methods. This actionable guidance (that LoRA reduces membership inference vulnerability by an order of magnitude) may prevent privacy harms that would otherwise occur.

By providing a stronger, more efficient auditing tool and releasing our code to the research community, we aim to empower developers, researchers, and regulators to:

- Conduct more realistic and rigorous privacy audits before deployment.
- Develop and calibrate stronger privacy-preserving defenses against a realistic threat model.
- Make informed choices about fine-tuning methodology based on privacy requirements.
- Foster greater transparency and accountability regarding the privacy properties of deployed AI systems.

We believe that the transparent, rigorous quantification of privacy risks is an essential prerequisite for developing technology that is both powerful and safe.

References

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

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? In *First Conference on Language Modeling*.

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. 2024. Practical membership inference attacks against fine-tuned large language models via self-prompt calibration. In *Advances in Neural Information Processing Systems*, volume 37.

Kazuki Fujii and Rio Yokota. 2024. Swallow-code-v0.1. https://huggingface.co/datasets/tokyotech-llm/swallow-code-v0.1.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28, pages 1693–1701.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343, Toronto, Canada. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*. Presented at ICLR 2017.

Rishabh Misra. 2022. News category dataset. *arXiv preprint arXiv:2209.11429*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. 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*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Nikhil Pinnaparaju, Reshinth Adithyan, Duy Phung, Jonathan Tow, James Baicoianu, and Nathan Cooper. 2023. Stable code 3b. `https://huggingface.co/stabilityai/stable-code-3b`.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Lorenzo Rossi, Michael Aerni, Jie Zhang, and Florian Tramèr. 2025. Membership inference attacks on sequence models. In *2025 IEEE Security and Privacy Workshops (SPW)*, pages 98–110. IEEE.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 billion parameter autoregressive language model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Fei Wang and Baochun Li. 2025. Leaner training, lower leakage: Revisiting memorization in LLM fine-tuning with LoRA. *arXiv preprint arXiv:2506.20856*.

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. 2025a. Min-k%++: Improved baseline for detecting pre-training data from large language models. In *The Thirteenth International Conference on Learning Representations*.

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

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, pages 649–657.

A Implementation Details

A.1 Data Processing

For each dataset, we sample sequences from the training split to create two disjoint evaluation sets: (1) 10,000 target members used to fine-tune the target model and labeled as members for evaluation, and (2) 10,000 target non-members labeled as non-members for evaluation. We maintain a held-out

validation set of 500 sequences for model selection during fine-tuning. All sequences are exactly 128 tokens, constructed by concatenating consecutive texts and discarding overflow.

A.2 Target Model Fine-tuning

Table 4 summarizes the fine-tuning configuration for each model. All models use the AdamW optimizer (Loshchilov and Hutter, 2019). We select the checkpoint with the lowest validation loss to mitigate overfitting artifacts that could confound membership signals.

Model	Method	Epochs	LR	Batch	LoRA Config
GPT-2 (124M)	Full	3	1×10^{-4}	16	—
GPT-J (6B)	LoRA	3	1×10^{-4}	16	r=16, α=32
Llama-2 (7B)	LoRA	3	1×10^{-4}	16	r=16, α=32
Stable-Code (3B)	LoRA	3	1×10^{-4}	16	r=16, α=32

Table 4: Fine-tuning configuration for target models. LoRA configurations use dropout 0.05. GPT-2 uses full fine-tuning; larger models use LoRA for computational efficiency.

B Baseline Implementation Details

All baseline attacks are evaluated under identical experimental conditions to EZ-MIA: same models, same fine-tuning protocol (3 epochs with validation-based checkpoint selection), same data splits (10k members, 10k non-members), and same evaluation metrics.

Reference-Free Attacks. We implement LOSS (Yeom et al., 2018), Zlib (Carlini et al., 2021), and Min-K%++ (Zhang et al., 2025a) following their original descriptions. Our implementations are included in the repository.

Reference Loss. We implement the likelihood ratio between target and pretrained reference models following the framework of Carlini et al. (2022). The score is computed as $L_{\text{ref}} - L_{\text{target}}$, where L denotes mean negative log-likelihood.

SPV-MIA. For SPV-MIA (Fu et al., 2024), we use the official released code executed without modification. Since the paper reports multiple algorithm variants, we use Algorithm 2 (the best-performing version with publicly available code). The code is included in our repository.

This controlled experimental setup ensures that performance differences between methods reflect the attacks themselves rather than experimental conditions.

Dataset	Model	EZ-MIA	SPV	Mult.
WikiText	GPT-2	14.0%	1.75%	8.0×
WikiText	GPT-J	1.6%	0.47%	3.4×
WikiText	Llama-2	2.0%	0.97%	2.1×
AG News	GPT-2	9.8%	2.29%	4.3×
AG News	GPT-J	5.7%	2.18%	2.6×
AG News	Llama-2	14.6%	1.57%	9.3×
XSum	GPT-2	33.1%	5.41%	6.1×
XSum	GPT-J	5.4%	0.73%	7.4×
XSum	Llama-2	4.3%	0.49%	8.8×
Average		**10.1%**	1.76%	**5.7**×

Table 5: TPR@0.1%FPR comparison. EZ-MIA achieves higher detection rates at this stringent threshold across all configurations, with improvements ranging from 2× to 9×.

C TPR at Stringent Thresholds

Table 5 presents TPR@0.1%FPR results across all configurations. This stringent threshold is critical for privacy auditing applications where false positives are costly.

D Principled Derivation of the Error Zone Score

This appendix provides a theoretical grounding for the Error Zone score, deriving its form from first principles and validating the underlying assumptions empirically.

D.1 The Detection Problem

Let $\delta_t = \log p_\theta(x_t \mid x_{<t}) - \log p_{\hat{\theta}}(x_t \mid x_{<t})$ denote the log-probability shift at position t after fine-tuning, where θ is the fine-tuned target and $\hat{\theta}$ is the pretrained reference. We restrict attention to error positions $\mathcal{E}$ where the fine-tuned model does not predict the correct token.

Goal: From the vector $\boldsymbol{\delta} = (\delta_t)_{t \in \mathcal{E}}$, determine whether sequence $\boldsymbol{x}$ was in the training set.

D.2 The Memorization Signal

We posit an additive model for the log-probability shift:

$$\delta_t = M_t + G_t \tag{5}$$

where:

- $M_t \geq 0$ is the memorization effect: the direct consequence of gradient updates on $\boldsymbol{x}$
- G_t is the generalization effect: the consequence of training on other sequences

The key assumption is that memorization is non-negative. When gradient descent optimizes

$-\log p_\theta(x_t \mid x_{<t})$, it increases $p_\theta(x_t \mid x_{<t})$. Training on a sequence cannot systematically decrease the probability of its own tokens.

For non-members, $M_t = 0$ by definition. Thus:

	Expected δ_t
Non-member	G_t
Member	$M_t + G_t \geq G_t$

The signal we seek is the *upward shift* induced by $M_t \geq 0$.

D.3 Why Raw Sums Fail

The natural first attempt is $\sum_t \delta_t$. If memorization adds positive mass, members should have larger sums. But this conflates two distinct quantities:

1. **Signal:** The memorization contribution $\sum_t M_t$
2. **Scale:** The intrinsic variability of $\boldsymbol{\delta}$ for sequence $\boldsymbol{x}$

Different sequences exhibit wildly different scales. A sequence with rare or unpredictable tokens will have large $|\delta_t|$ values regardless of membership; the model's probability estimates swing more dramatically on harder tokens.

Sequence	$\boldsymbol{\delta}$	$\sum \delta_t$	Pattern
A (predictable)	$(0.1, 0.2, -0.1)$	0.2	3/4 up
B (volatile)	$(1.0, 2.0, -1.0)$	2.0	3/4 up

Sequence B has a $10\times$ larger sum, but both have the same pattern: three-quarters of the movement is upward. Comparing raw sums across sequences conflates the membership signal with sequence-specific volatility.

D.4 Decomposing Probability Movement

We reframe the problem in terms of probability mass movement. Fine-tuning induces changes at each error position with a direction (up or down) and magnitude. Define:

$$P = \sum_{t \in \mathcal{E}} [\delta_t]_+ \qquad N = \sum_{t \in \mathcal{E}} |[\delta_t]_-| \tag{6}$$

where $[x]_+ = \max(x, 0)$ and $[x]_- = \min(x, 0)$.

- P = total probability mass moved upward
- N = total probability mass moved downward
- $P + N$ = total movement (the "budget" of change)

This decomposition is exhaustive: every probability adjustment is either up or down.

D.5 The Ratio Formulation

Rather than asking "how much did probabilities increase?" (scale-dependent), we ask:

> *Of all probability movement that occurred, what fraction was upward?*

This fraction is $f_{\text{up}} = P/(P + N)$, which is scale-invariant: multiplying all δ_t by a constant $c > 0$ leaves f_{up} unchanged.

The Error Zone score uses the equivalent odds formulation:

$$\text{EZ} = \frac{P}{N} \tag{7}$$

The relationship to the fraction form is monotonic: $f_{\text{up}} = \text{EZ}/(1 + \text{EZ})$.

Why odds? The odds formulation offers several advantages:

1. Interpretability: EZ $= 3$ means "three units of probability moved up for every one that moved down."
2. Multiplicative structure: Effects that scale the imbalance act multiplicatively on EZ.
3. Unbounded range: EZ $\in (0, \infty)$ with EZ $= 1$ as the neutral point, avoiding ceiling effects when the signal is strong.

D.6 Why EZ Captures Memorization

Under our additive model, what happens when memorization is present? For a member, $\delta_t = M_t + G_t$ with $M_t \geq 0$. Compared to a non-member (where $\delta_t = G_t$), adding $M_t > 0$ has the following effects:

Original G_t	Effect of $M_t > 0$	Impact
$G_t > 0$	δ_t increases	$P \uparrow$
$G_t < 0, \lvert G_t \rvert > M_t$	$\delta_t \to 0$	$N \downarrow$
$G_t < 0, \lvert G_t \rvert < M_t$	δ_t flips positive	$P \uparrow, N \downarrow$

In all cases, the ratio P/N increases (or stays the same if $M_t = 0$). The signature of memorization is an elevated EZ.

D.7 Empirical Validation of Assumptions

Non-negative memorization ($\mathbb{E}[M_t] \geq 0$). We test our core assumption that memorization induces non-negative probability shifts by partitioning sequences into difficulty bins based on reference model perplexity and computing the mean δ at error positions for members versus non-members. If this assumption holds, the member mean should exceed the non-member mean across all difficulty levels.

Bin	$\mathbb{E}[\delta]$ Mem.	$\mathbb{E}[\delta]$ Non-mem.	p
Q1 (easy)	0.312	0.041	< 0.0001
Q2	0.298	0.029	< 0.0001
Q3	0.291	0.018	< 0.0001
Q4	0.287	0.012	< 0.0001
Q5 (hard)	0.279	0.005	< 0.0001

Finding: $\mathbb{E}[\delta_{\text{member}}] > \mathbb{E}[\delta_{\text{non-member}}]$ in all five difficulty bins with $p < 0.0001$. The non-negative memorization assumption is strongly supported.

Distributional structure. A stronger assumption would be that member and non-member δ distributions differ only by a location shift. We test this by removing the mean difference and comparing distributions via the Kolmogorov-Smirnov test.

After mean-shift removal, we observe KS statistic $= 0.120$ ($p < 0.0001$), indicating the distributions are not identical. However, examining quantiles reveals that the distributions align closely across the central 80% of probability mass:

Quantile	Shifted Mem.	Non-mem.	Diff.
10th	-0.599	-0.518	-0.081
25th	-0.358	-0.247	-0.111
50th	-0.115	-0.021	-0.094
75th	$+0.223$	$+0.205$	$+0.018$
90th	$+0.699$	$+0.493$	$+0.206$

The divergence concentrates in the right tail (90th percentile and above), where members show excess positive mass. While the violation is statistically significant, we judge the effect size as modest and not practically significant.

E Ablations and Design Choices

This appendix provides empirical justification for key design decisions in EZ-MIA. All experiments in this section use a preliminary evaluation with 1,000 members and 1,000 non-members on XSum with GPT-2-XL. While absolute performance differs from the full 10k/10k evaluation in the main paper, relative comparisons between design choices remain valid.

E.1 Why Error Positions?

Error vs. Success Positions. We compare EZ computed at error positions (where target's top prediction $\neq$ ground truth) versus success positions (where top prediction $=$ ground truth).

Position Type	AUC
Error positions	0.895
Success positions	0.797

Restricting to error positions provides a +0.1 AUC improvement. This confirms our key insight: the membership signal concentrates where the model fails.

Error Count Distribution. Members exhibit fewer errors than non-members on average:

Group	Mean Errors	Std
Members	172.97	76.65
Non-members	189.31	81.91

This difference is significant ($p < 0.0001$) but the correlation with membership is weak ($r = -0.102$), confirming that EZ captures signal beyond simple error counting.

E.2 Reference Model Variants

Reference Model Quality. We compare different reference model choices:

Reference Model	AUC
Pretrained GPT-2-XL	0.895
Random initialization	0.591
Self-reference (target as reference)	0.500

The pretrained reference is essential. Random weights provide minimal signal, and self-reference yields chance performance (confirming pipeline correctness as a sanity check).

Cross-Architecture Reference. Can a smaller model serve as reference?

Reference Architecture	AUC
Same (GPT-2-XL $\rightarrow$ GPT-2-XL)	0.895
Cross (DistilGPT-2* $\rightarrow$ GPT-2-XL)	0.789

*Sanh et al. (2019)

Same-architecture reference performs substantially better (-0.106 AUC for cross-architecture).

Unrelated Domain Reference. Does the reference model need to match the target's domain?

Reference Model	AUC
Pretrained GPT-2-XL	0.895
WikiText fine-tuned GPT-2-XL	0.864

A reference fine-tuned on unrelated domain data (WikiText) still achieves strong performance, only 0.031 AUC below the pretrained baseline. This suggests EZ is robust to reference model choice.

Distillation Reference. Following Fu et al. (2024), we evaluate a distillation-based reference constructed by prompting the target model and fine-tuning on its generations.

Construction: We prompt the fine-tuned target with up to 10,000 seed texts from related public datasets (News Category Dataset (Misra, 2022) for AGNews, CNN/DailyMail (Hermann et al., 2015) for XSum, and Wikipedia for Wikitext, truncated to 16 tokens), generate 112-token completions using nucleus sampling (Holtzman et al., 2020) ($p = 0.9$, temperature $= 0.9$), and fine-tune a fresh copy of the base model on these synthetic texts for 4 epochs.

Reference	AUC	Additional Training
Pretrained (Base)	0.895	None
Distillation	0.891	4 epochs

Distillation provides no improvement over the simpler pretrained reference on this configuration while requiring additional computation. We recommend the pretrained reference as the default choice.

E.3 Aggregation Function

We compare alternative ways to aggregate the δ values at error positions:

Aggregation	AUC
Positive fraction	0.904
Median δ	0.899
$P - N$ (difference)	0.898
P/N (EZ)	0.895
$\log(P/N)$	0.895
Mean δ	0.877

Several aggregations achieve similar performance. We select P/N (EZ) for its theoretical grounding (scale invariance, interpretability as odds ratio) despite slightly lower AUC than positive fraction. All methods show high correlation ($r > 0.75$).

E.4 Error Definition

We vary the definition of "error" from top-1 (model's best prediction is wrong) to top-K (correct token not in top K predictions):

Definition	AUC	Error Fraction
Top-1	0.895	50.6%
Top-5	0.876	25.9%
Top-10	0.855	18.6%

Top-1 is optimal. Stricter definitions (top-5, top-10) reduce the number of positions considered and discard useful signal.

E.5 Zero-Error Sample Handling

Sequences where the model predicts every token correctly (zero errors) yield undefined EZ. In our evaluation, 0/1000 members and 0/1000 non-members had zero errors, so all strategies yield identical results. For robustness, we recommend treating such samples as members.

E.6 N=0 Edge Case

When all probability movement at error positions is upward ($N = 0$, $P > 0$), the EZ score P/N is undefined. However, this case represents the strongest possible membership signal—fine-tuning increased probability at every error position. In practice, we assign EZ $= \infty$ (or a large constant) to such sequences, classifying them as members. This edge case is rare; in our evaluation, 0/1000 members and 0/1000 non-members exhibited $N = 0$.